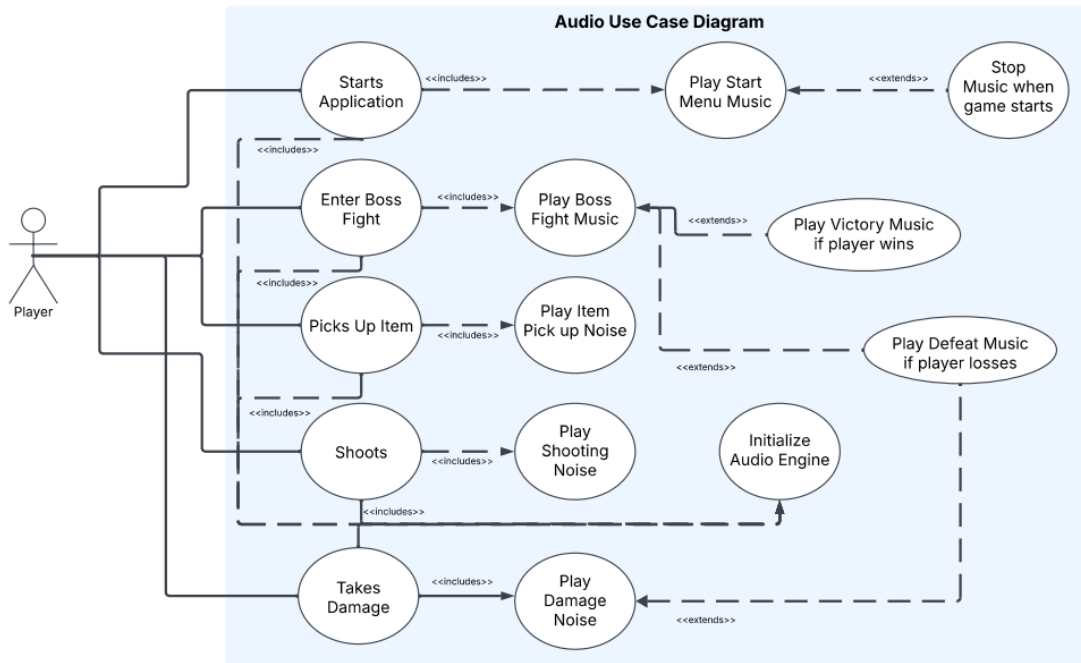


## 1. Brief introduction

I will implement the audio system for our game. The main menu and the final boss fight will both feature music. Unique sounds will play when the player shoots, picks up an item, and takes damage.

## 2. Use case diagram with scenario \_14



### Scenarios

**Name:** Start Application

**Summary:** The player loads the game.

**Actors:** Player

**Preconditions:** The game is on the computer

**Basic sequence:**

- 1) Player selects the application on their computer
- 2) Game is loaded
- 3) The main menu is show

**Exceptions:**

- 1) None, this will always occur when loading the game.

**Post conditions:** Player is in main menu.

**Priority:** 1\*

**ID:** C01

**Name:** Enter Boss Fight

**Summary:** The player has completed the main portion of the game and is ready to fight the boss

**Actors:** Player

**Preconditions:** The player has collected all the necessary items to fight the boss.

**Basic sequence:**

- 1) Player collects items
- 2) Player travels to the boss area
- 3) Player commits to fight the boss

**Exceptions:**

- 1) The player does not have the necessary items to fight the boss

**Post conditions:** Player is fighting the boss

**Priority:** 1\*

**ID:** C02

**Name:** Picks Up Item

**Summary:** The player picks up an item on their list in the main portion of the game

**Actors:** Player

**Preconditions:** The player has entered the main menu and selected to play the game

**Basic sequence:**

- 1) Player moves to area with item
- 2) Player moves over location with item
- 3) Player picks up item

**Exceptions:**

- 1) The player already has the item

**Post conditions:** Player has item in inventory and is one step closer to boss.

**Priority:** 1\*

**ID:** C03

**Name:** Shoots

**Summary:** The player shoots their weapon at an enemy

**Actors:** Player

**Preconditions:** The player has collected their weapon and selected in in their inventory

**Basic sequence:**

- 1) Player hovers mouse over location to shoot
- 2) Player clicks to shoot
- 3) Bullet travels to location and hits enemy or misses

**Exceptions:**

- 1) The player has dropped their weapon
- 2) The player is out of ammo

**Post conditions:** Player deals damage to enemy or bullet disappears without hitting enemy

**Priority:** 1\*

**ID:** C04

**Name:** Takes Damage

**Summary:** Player is hit by the weapon of an enemy

**Actors:** Player

**Preconditions:** Player is in area of enemy in the main game

**Basic sequence:**

- 1) Player enters vision of enemy
- 2) Enemy shoot projectile at player
- 3) Player is hit by projectile and takes damage

**Exceptions:**

- 1) The projectile misses

**Post conditions:** The players health is lower

**Priority:** 1\*

**ID:** C05

**Name:** Play Boss Fight Music

**Summary:** Player is fighting the boss and intense music plays

**Actors:** Player

**Preconditions:** Player has made it to the boss fight, audio engine is initialized

**Basic sequence:**

- 1) Player enters the boss fight
- 2) Music Plays

**Exceptions:**

- 1) None, if the player is in the boss fight, music will play

**Post conditions:** Music is playing

**Priority:** 2\*

**ID:** C06

**Name:** Play Main Menu Music

**Summary:** Player is in the main menu and music plays

**Actors:** Player

**Preconditions:** Player has loaded the game

**Basic sequence:**

- 1) Player loads the game
- 2) Player enters the main menu
- 3) Music Plays

**Exceptions:**

- 1) If the player starts the game, the music will stop

**Post conditions:** Music is playing

**Priority:** 2\*

**ID:** C07

**Name:** Stop music when game starts

**Summary:** Player starts the game, and the main menu music starts

**Actors:** Player

**Preconditions:** Player has selected to play the game, audio engine is initialized

**Basic sequence:**

- 1) Player selects to play the game in the main menu
- 2) Scene is changed to main game

**Exceptions:**

- 1) None, if the player is playing the game the main menu music will stop

**Post conditions:** Music is stopped

**Priority:** 2\*

**ID:** C08

**Name:** Play pick up noise

**Summary:** Player interacts with an item and a noise plays

**Actors:** Player

**Preconditions:** Player is playing the main game, audio engine is initialized

**Basic sequence:**

- 1) Player moves over the area of a item
- 2) Item is picked up
- 3) Noise plays to indicate items have been picked up

**Exceptions:**

- 1) None, if the player picks up an item the noise will play

**Post conditions:** Noise

**Priority:** 2\*

**ID:** C9

**Name:** Play shooting noise

**Summary:** Player shoots and a noise plays

**Actors:** Player

**Preconditions:** Player is playing the main game, audio engine is initialized

**Basic sequence:**

- 1) Player shoots a weapon
- 2) Projectile is fired
- 3) Shooting noise is played

**Exceptions:**

- 1) None, if the player shoots the noise will play

**Post conditions:** Noise

**Priority:** 2\*

**ID:** C10

**Name:** Play damage noise

**Summary:** Player takes damage from an enemy and a noise plays

**Actors:** Player

**Preconditions:** Player is playing the main game, audio engine is initialized

**Basic sequence:**

- 1) Player enters the area of an enemy
- 2) Enemy fires at the player and hits them
- 3) Damage is dealt and the noise plays

**Exceptions:**

- 1) If the player takes damage and runs out of health, then defeat music will play instead

**Post conditions:** Noise

**Priority:** 2\*

**ID:** C11

**Name:** Initialize Audio Engine

**Summary:** Player takes damage from an enemy and a noise plays

**Actors:** Game System

**Preconditions:** The game is launching or resuming from a state where no audio needs initialized

**Basic sequence:**

- 1) Game starts or resumes
- 2) Game system initializes the audio engine and all necessary resource
- 3) Audio system checks for available sound devices and loads assets
- 4) Background music starts playing if necessary

**Exceptions:**

- 1) If initialization fails, an error is logged, and a fallback mechanism is used.

**Post conditions:**

- 1) The audio engine is fully functional, ready to handle background music and sound effects.
- 2) Game sounds (such as shooting, item pickups, and damage sounds) can now play without issues.

**Priority:** 1\*

**ID:** C12

**Name:** Play victory music if player wins

**Summary:** Player defeats the boss and music plays

**Actors:** Player

**Preconditions:** Player is playing the main game, audio engine is initialized

**Basic sequence:**

- 1) Player enters the area of the boss
- 2) Enemy defeats the boss
- 3) A victory screen is displayed and music plays

**Exceptions:**

- 1) None, if the player defeats the boss music plays

**Post conditions:** Victory Music

**Priority:** 2\*

**ID:** C13

**Name:** Play defeat music if player losses

**Summary:** Boss defeats the player and music plays

**Actors:** Player

**Preconditions:** Player is playing the main game, audio engine is initialized

**Basic sequence:**

- 1) Player enters the area of the boss
- 2) Boss defeats the player
- 3) A defeat screen is displayed and music plays

**Exceptions:**

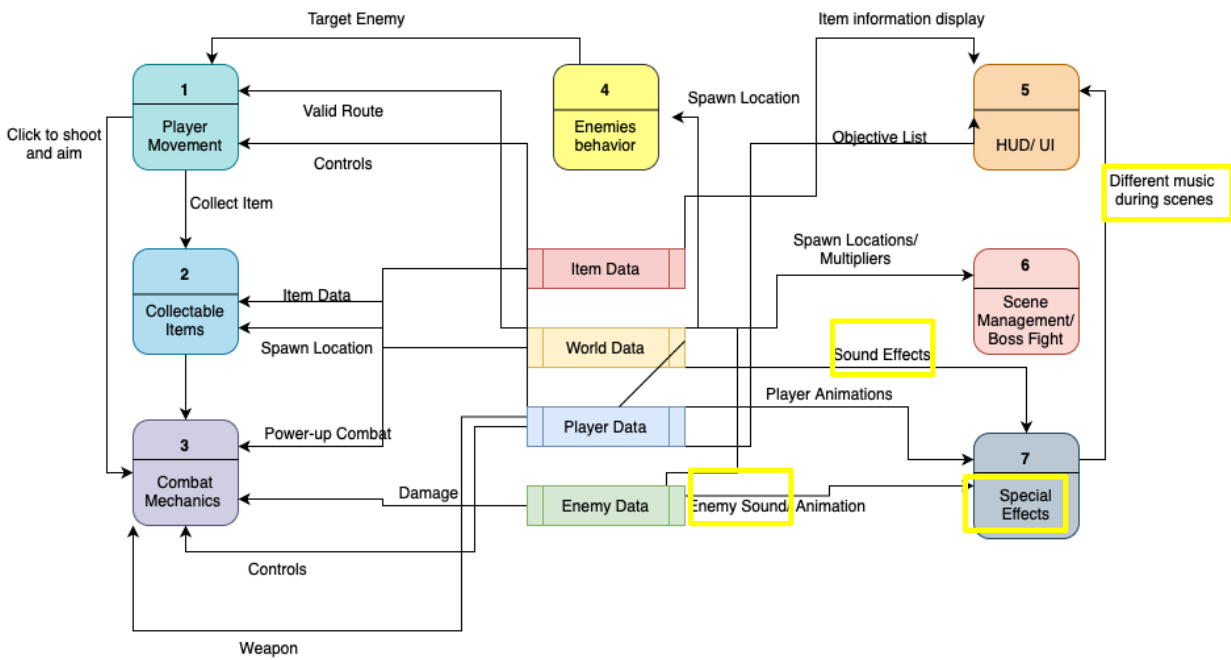
- 4) None, if the boss defeats the player music plays

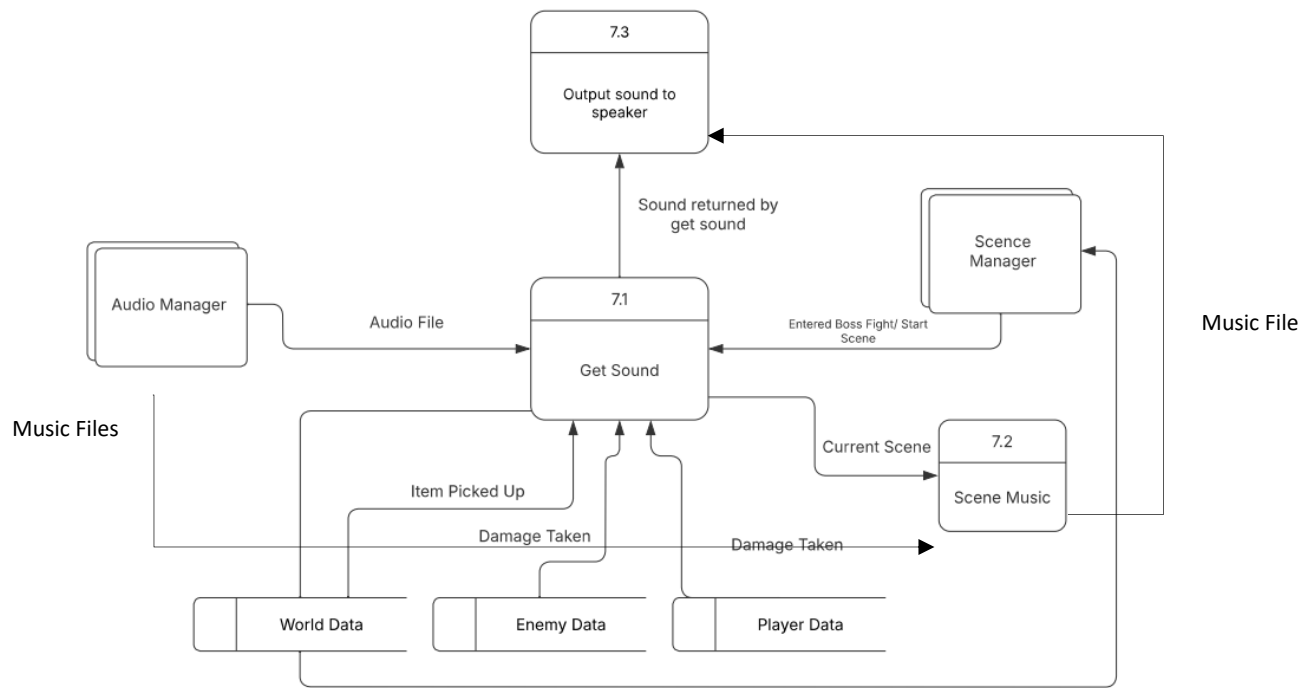
**Post conditions:** Defeat Music

**Priority:** 2\*

**ID:** C14

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





## Process Descriptions

### Get Sound:

IF Audio Manager is initialized and data received from world, enemy, or player data:

Detect type of data received from store

Determine what audio needs to be requested from the audio manager

Request Audio from Audio Manager and output to speaker

END

### Scene Music:

IF current scene data received from Get Sound and No music is playing (game is starting):

Play scene corresponding scene music

Loop music until different data is received

IF current scene data received from Get Sound and different music is playing:

Stop current scene music

Start Scene music for current scene



Loop music until different data is received  
END

Output to Speaker:

IF music Files received:

Play music continuously

If sound file received:

Play sound once

END

#### 4. Acceptance Tests \_\_\_\_\_9

Ensure that the correct audio queues play at the correct times. Ensure that music starts and stops on the correct scenes.

Example for music transitions from scene to scene:

Test feature on the transitions from the start menu scene to the main game scene.

Inputs:

- Start Menu Music
- Start Menu Scene
- Main Game Scene
- Main Game Music

Outputs:

- Player Selects 'Play Game' on Start Scene
  - Main Menu Music Stops
  - Game Music Plays
- Player Stays on the Starts Scene for an extended amount of time
  - Main Menu Music loops indefinitely

Example of Item Collection Sound Queues

Test Case	Description	Input	Output
1	Player Moves to location of item	Player Collision with item, Player Location, Item Type	Sound for corresponding item will be played
2	Player misses'	Player Collision with	No sound plays unless the player directly contacts the item.

	location of item	item, Player Location, Item Type	
3	Player moves to location of item that has already been picked up	Player Collision with item, Player Location, Item Type,	No sound plays for an item that has already been collected. It should be removed from the scene.

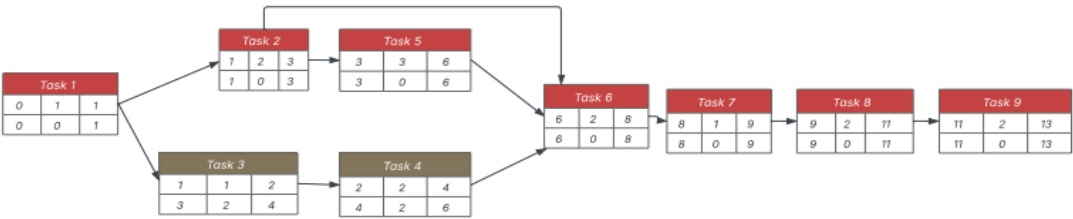
## 5. Timeline \_\_\_\_/10

### Work items

Task	Duration (PWks)	Predecessor Task(s)
1. Requirements Collection	1	-
2. Scene Design	2	1
3. Find Audio Files to include	1	1
4. Program Audio Manager	2	3
5. Creation of the World, Player, and Enemy Data Sources	3	2
6. Program Get Sound feature	2	5, 2
7. Integrate Get Sound into Scene Manger and output to speaker	1	6
8. Testing	2	7

9. Deployment	2	8
---------------	---	---

### Pert diagram



### Gantt timeline

