

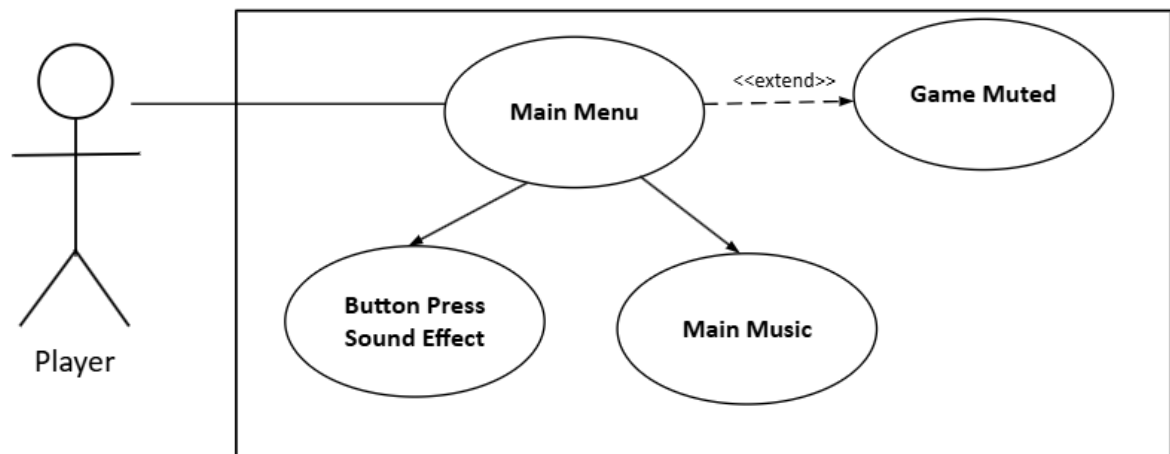
1. Brief introduction __/3

My feature is pretty simple, sound. I will be responsible for all the background music playing throughout different scenes. I am also responsible for all the different sound effects that accompany different animations.

This includes a few different background songs that will loop and a group of sound effects to accompany different events. The largest group of these sound effects will be different sound effects for each group of monsters, which will need to be retrieved every turn of battle.

2. Use case diagram with scenario __14

Use Case Diagrams



Scenarios

Name: Main Menu

Summary: This is the main menu/ title screen that introduces the player to the game.

Actors: Player.

Preconditions: Game is loaded.

Basic sequence:

Step 1: Load game, bringing up the main menu.

Step 2: Play the Main music.

Step 3: Play the Button Press Sound Effect each time a button is pressed.

Exceptions:

Step 1: The game is muted, so no Main Music plays.

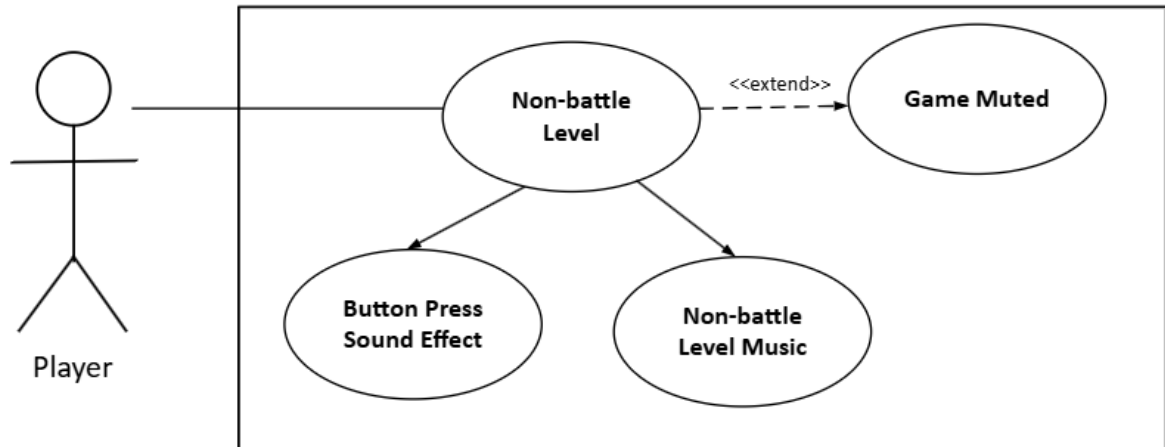
Step 2: The game is muted, so no Sound Effects play.

Post conditions: None

Priority: 2*

ID: C01

Use Case Diagrams



Scenarios

Name: Non-battle Levels

Summary: These are the levels the player encounters that are not battles.

Actors: Player.

Preconditions: A non-battle level has been accessed.

Basic sequence:

Step 1: The non-battle level begins.

Step 2: The non-battle level music plays.

Step 3: Anytime a button is pressed, the button press sound effect plays.

Exceptions:

Step 1: If the game is muted, no music.

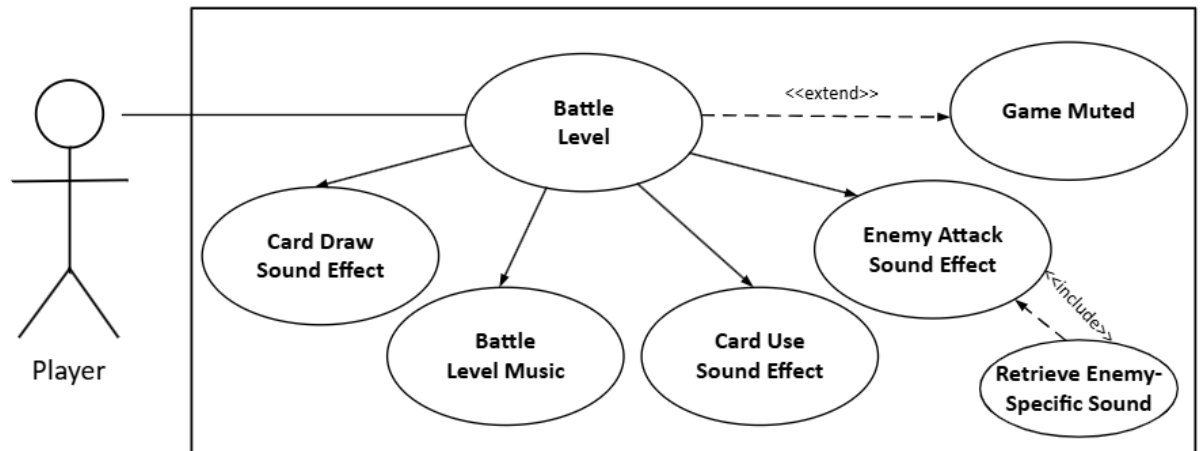
Step 2: If the game is muted, no sound effects.

Post conditions: None.

Priority: 2*

ID: C02

Use Case Diagrams



Scenarios

Name: Battle Level

Summary: These are the levels that include enemies for the player to battle using their deck of cards.

Actors: Player.

Preconditions: A battle level has been reached.

Basic sequence:

Step 1: The battle level begins.

Step 2: Battle music begins playing.

Step 3: The player draws a card (Sound Effect).

Step 4: The player selects and plays a card (Sound Effect).

Step 5: The enemy takes its turn.

Step 6: Determine what enemy group is active and what sound will be used.

Exceptions:

Step 1: If the game is muted, do not play music.

Step 2: If the game is muted, do not play sound effects.

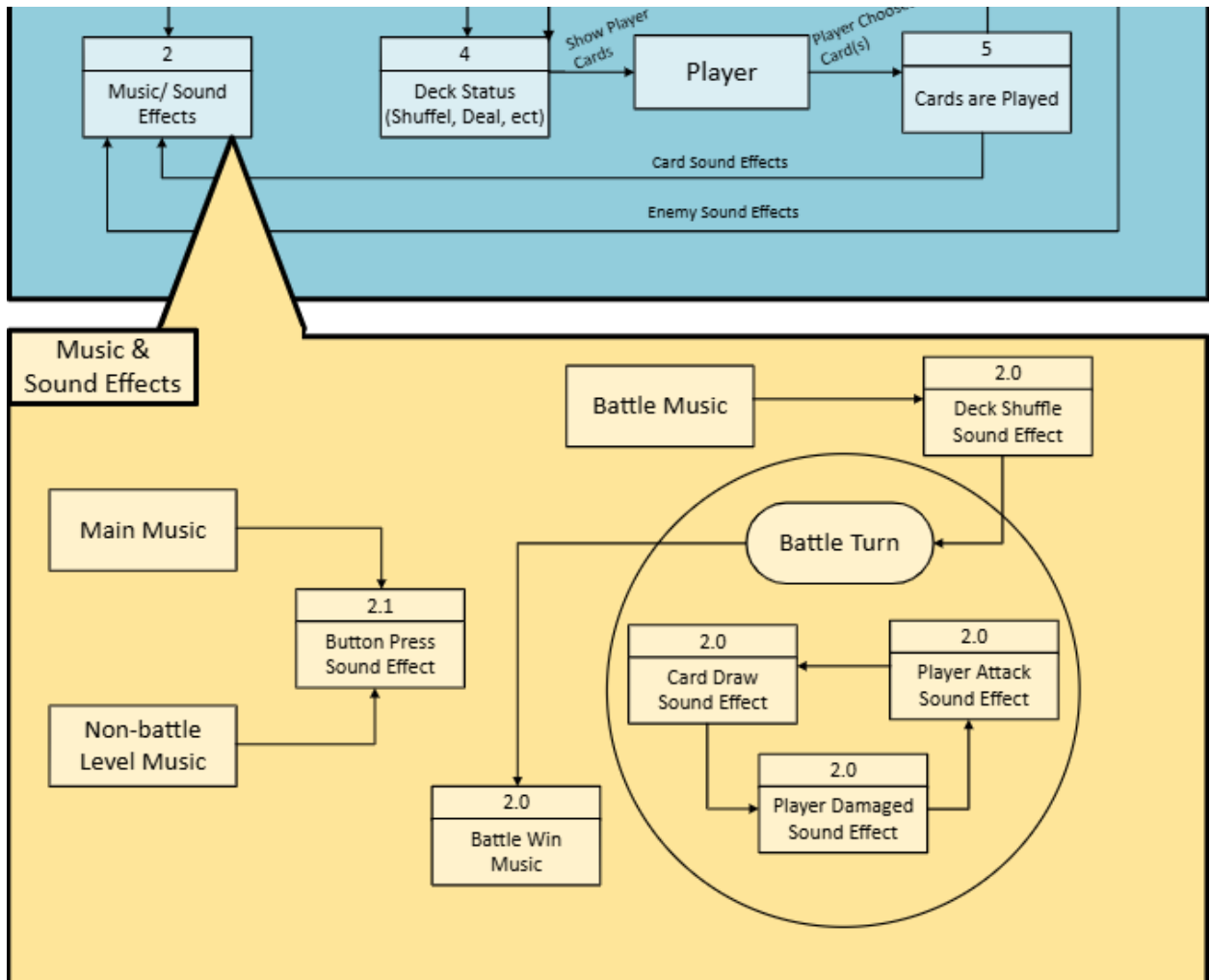
Post conditions: None.

Priority: 1*

ID: C03

3. Data Flow diagram(s) from Level 0 to process description for your feature ____14

Data Flow Diagrams



Process Descriptions

Assign rooms*:

WHILE teacher in two places at once OR two classes in the same room
 Randomly redistribute classes
 END WHILE

Main Menu:

WHILE on main menu:
 Play main music
 IF button pressed:
 Play Button Press Sound Effect
 END IF
 END WHILE

Non-Battle Level:

```
WHILE on non-battle level:  
    Play non-battle level music  
    IF button pressed:  
        Play button press sound effect  
    END IF  
END WHILE
```

Battle Level:

```
WHILE on battle level:  
    Play battle level music  
    WHEN deck shuffling:  
        Play deck shuffle sound effect  
    END WHEN  
    WHILE player is alive AND enemy is alive:  
        WHEN player draws card:  
            Play draw card sound effect  
        END WHEN  
        WHEN player uses card:  
            Play card use sound effect  
        END WHEN  
        WHEN enemy attacks:  
            IDENTIFY correct enemy sound effect  
            Play correct enemy sound effect  
        END WHEN  
    END WHILE  
    Play battle win music  
END WHILE
```

4. Acceptance Tests 9

There are no inputs specific to the sound of the game. All sound scripts will be called at the same time as other scripts, making the input tests for those features dependent on other team members.

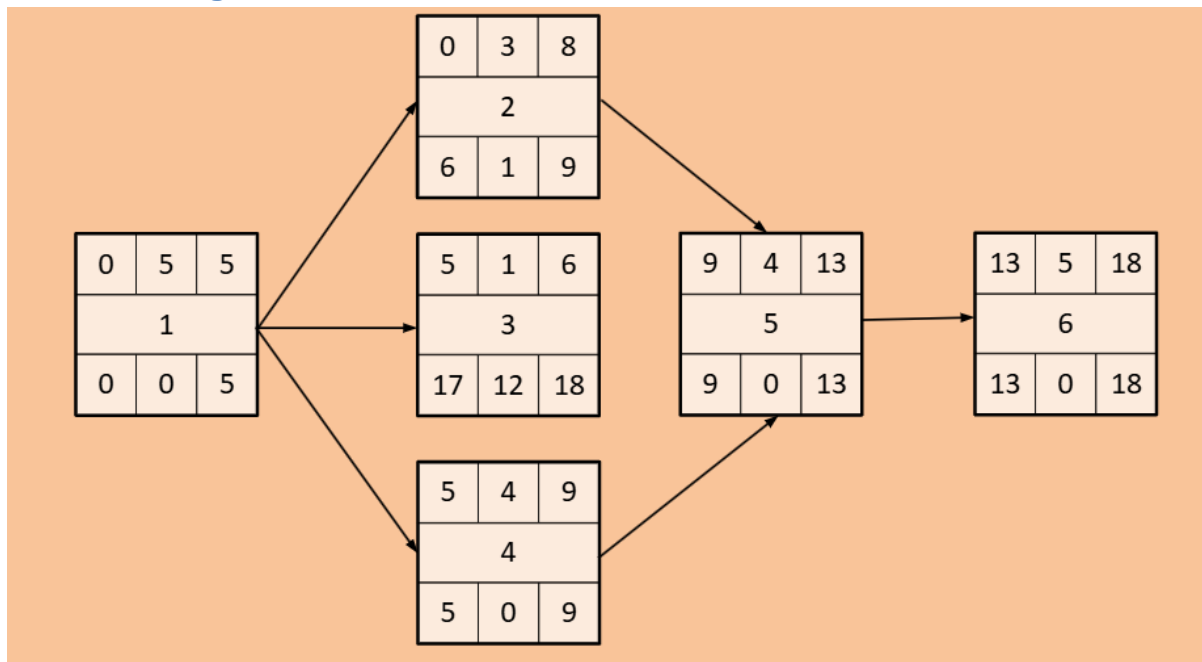
There is one thing to test, that being the background music that must loop. Ensuring each music file can loop for at least as long as the rest of the program can run is important. This can be done by letting the game sit idle for 2 hours for each song, ensuring the music loop correctly.

5. Timeline ____/10

Work items

Task	Duration (PWks)	Predecessor Task(s)
1. Collect Music + Sound Files	5	-
2. Implement Main Music	3	1
3. Implement Button Press SE	1	1
4. Implement Other Music	4	1
5. Test Music Transitions	4	2,4
6. Implement Other SE's	5	5

Pert diagram



Gantt timeline

