Name: Gurneet Kaur Mark \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/50

# Brief introduction \_\_/3

This feature implements the Audio System for the Orange Ninja Game, adding an immersive layer to gameplay. The system handles all sound effects (SFX), background music (BGM), and event-triggered sounds to enhance player experience. Every key player action—such as jumping, attacking, or collecting items—has an associated sound cue, providing feedback and reinforcing the game’s fast-paced and humorous tone. The background music dynamically shifts between calm exploration tracks and intense boss battle themes, maintaining excitement. Volume controls and mixing ensure a balanced audio experience across different devices and player preferences.

# Use case diagram with scenario \_\_14

**Actors**

* Player (primary)
* Game System (secondary)

**Use Cases**

* Play SFX
* Play BGM
* Adjust Volume
* Trigger Event Sound <<include>> Play SFX (dynamic binding: event→specific SFX)
* Switch to Boss BGM <<extend>> Play BGM (exception at boss start)
* Pause/Resume Audio <<include>> Save Audio State
* Mute/Unmute

**Scenarios (one for each key use case)**

**2.1 Scenario: Play SFX**

**Summary:** Play a short sound when a player or enemy event occurs.  
**Actors:** Player, Game System  
**Preconditions:** Audio system initialized; SFX asset mapped to event ID.  
**Basic Sequence:**

1. Game System detects an event (e.g., player jumps, attacks, item pickup).
2. Audio System looks up SFX mapped to the event.
3. Audio System plays the SFX via mixer.  
   **Exceptions:**

* **Step 2:** Mapping missing → play default fallback SFX.
* **Step 3:** Output device unavailable → queue SFX and log error.  
  **Postconditions:** Appropriate SFX feedback produced.  
  **Priority:** 1 (must have) • **ID:** A01

**2.2 Scenario: Play BGM**

**Summary:** Start and loop background music for the current context (menu/level).  
**Actors:** Game System  
**Preconditions:** Context (menu/level) loaded; BGM track available.  
**Basic Sequence:**

1. Context start event fires.
2. Audio System loads the context BGM.
3. Mixer starts playback with loop enabled.  
   **Exceptions:**

* **Step 2:** Track missing/corrupt → continue without BGM, log error.
* **Step 3:** Loop boundary pops/clicks detected → switch to seamless loop points.  
  **Postconditions:** Continuous BGM plays until context changes.  
  **Priority:** 1 (must have) • **ID:** A02

**2.3 Scenario: Adjust Volume**

**Summary:** Player changes SFX/BGM volume sliders in Settings.  
**Actors:** Player, Game System  
**Preconditions:** Settings menu open; mixer initialized.  
**Basic Sequence:**

1. Player opens “Audio” settings.
2. Player moves SFX and/or BGM slider(s).
3. Audio System updates corresponding mixer bus levels in real time.
4. System persists settings to profile.  
   **Exceptions:**

* **Step 3:** Volume set to 0 → bus is muted (no output).
* **Step 4:** Save fails → keep changes in session memory; re-prompt later.  
  **Postconditions:** New volume levels applied and saved.  
  **Priority:** 2 (essential) • **ID:** A03

**2.4 Scenario: Trigger Event Sound <<include>> Play SFX**

**Summary:** Route a game event to its audio cue (dynamic dispatch to correct SFX).  
**Actors:** Game System  
**Preconditions:** Event bus active; event→SFX map exists.  
**Basic Sequence:**

1. Event bus publishes gameplay event with parameters (type, intensity).
2. Audio System selects SFX variant (e.g., pitch/volume) based on parameters.
3. **<<include>> Play SFX** to render the chosen cue.  
   **Exceptions:**

* **Step 2:** No variant matches → use default variant.  
  **Postconditions:** Context-appropriate SFX plays.  
  **Priority:** 1 (must have) • **ID:** A04

**2.5 Scenario: Switch to Boss BGM <<extend>> Play BGM**

**Summary:** On boss encounter, transition from level BGM to boss theme.  
**Actors:** Game System  
**Preconditions:** Boss encounter triggered; boss track available OR fallback defined.  
**Basic Sequence:**

1. Boss start signal received.
2. Fade out current BGM over configured duration.
3. Load boss BGM track.
4. Fade in boss BGM and loop it.  
   **Exceptions:**

* **Step 3:** Boss track unavailable → **<<extend>>** falls back to continuing current BGM.
* **Step 2:** Fade fails (latency spike) → perform immediate crossfade.  
  **Postconditions:** Boss BGM active until boss defeated or encounter ends.  
  **Priority:** 2 (essential) • **ID:** A05

**2.6 Scenario: Pause/Resume Audio <<include>> Save Audio State**

**Summary:** Pausing the game suspends audio; resuming restores it seamlessly.  
**Actors:** Player, Game System  
**Preconditions:** Active playback (SFX/BGM).  
**Basic Sequence:**

1. Player presses Pause.
2. **<<include>> Save Audio State** (store BGM position, mixer levels, mutes).
3. Mixer suspends all buses or ducks to near-zero.
4. Player presses Resume; state is restored; playback continues.  
   **Exceptions:**

* **Step 2:** State write error → resume with best-effort defaults (restart BGM from loop start).  
  **Postconditions:** Audio matches pre-pause state as closely as possible.  
  **Priority:** 2 (essential) • **ID:** A06

**2.7 Scenario: Mute/Unmute**

**Summary:** Toggle global mute quickly during gameplay.  
**Actors:** Player  
**Preconditions:** Mixer supports master bus mute.  
**Basic Sequence:**

1. Player toggles mute.
2. Audio System mutes/unmutes master bus.
3. UI reflects current mute status.  
   **Exceptions:**

* **Step 2:** Hardware mute already engaged → show notice; keep UI in sync.  
  **Postconditions:** Global output matches player preference.  
  **Priority:** 3 (nice to have) • **ID:** A07

Player

Switch to Boss BGM

Play BGM

Game System

Adjust Volume

Pause/Resume Audio

Mute/Unmute



## Data flow Diagram

Player/Enemy Action

Game System Detects

Event (Jump, Attack

Audio System Selects

Correct SFX/BGM

Mixer Plays Sound

(Apply Volume/Mix)

Speakers Output

Sound to Player

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

|  |  |  |
| --- | --- | --- |
| Test | Input/Steps | Expected Output |
| 1 -Jump SFX | Press Jump key once | Jump SFX plays instantly(<=50 ms delay) |
| 2 -Attack Spam | Press Attack 10x quickly | 10 distinct attack sounds, no drops |
| 3 – Item Pickup | Collect 3 items in 200 ms | 3 overlapping pickup sounds play |
| 4-Level BGM | Load Level 1 | Level music starts looping in ≤0.5 s |
| |  | | --- | |  | | 5 – Pause/Resume | |  |  | | --- | |  | | Pause at 01:23, wait 2s, resume | Music resumes from same spot. |
| |  | | --- | | 6 – Volume Adjust |  |  | | --- | |  |  |  | | --- | |  | | |  | | --- | | Set SFX=30%, BGM=70% |  |  | | --- | |  | | |  |  |  | | --- | --- | --- | | |  | | --- | | Mixer reflects new levels, saved |  |  | | --- | |  | | |
| |  | | --- | | 7 – Mute Toggle |  |  | | --- | |  |  |  | | --- | |  | | |  | | --- | | Toggle Mute ON/OFF |  |  | | --- | |  | | |  |  |  | | --- | --- | --- | | |  | | --- | | Output silences and restores |  |  | | --- | |  | | |
| |  | | --- | | 8 – Missing Asset |  |  | | --- | |  | | |  | | --- | | Trigger sound that doesn’t exist |  |  | | --- | |  | | |  | | --- | |  | | Fallback sound plays, log warning | |  |  | | --- | |  |  |  | | --- | |  |  |  | | --- | |  | |
| |  | | --- | | 9 – Boss Switch |  |  | | --- | |  |  |  | | --- | |  | | |  | | --- | | Trigger boss event mid-level |  |  | | --- | |  | | Crossfade to boss BGM, no silence |

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

|  |  |  |
| --- | --- | --- |
| Task | Duration (PWks) | Predecessor Task(s) |
| |  | | --- | | 1. Define Audio Requirements |  |  | | --- | |  | | 1 | - |
| 2. Collect/Design Sound Assets | 2 | 1 |
| |  | | --- | | 3 Build Audio Engine & Mixer |  |  | | --- | |  | | 3 | 1 |
| 4.Integrate with Game Events | 2 | 3 |
| 5. Implement Boss Music Switch | 1 | 4 |
| 6. Test(Latency, Volume, Loops) | 1 | 5 |
| 7. Final Mix&Adjustments | 1 | 6 |
| 8. Debugging & Documentation | 1 | 7 |

## Gantt timeline

