# Samurai Village (2023-24)

# Sound Implementation

This section will guide you through what sounds and elements you should look to implement. While this guide tries to be as comprehensive as possible, you should use your own intuition to gauge what needs to be done.

**1.1 - Environmental Sounds**

You should aim to add all environmental sound to the game including but not limited to:

* Trees
* Birds Tweeting
* Fires
* Water
* Random Conversations (not dialogue)
* Other Character Sounds (should be attached to their animations if animated)
* Interactable environmental objects such as gongs etc.
* General Ambience (Area Loops)

Objects that are repeated across the game and have the same sounds should have sound added to the prefab (perfricated object). Single sounds or elements of selected variation should be implemented in the game world, not the prefab.

You should look to make sure that heavy amounts of randomization are utilized to create the illusion of a real world. See the module videos for more details.

**1.2 - Environmental Collision**

Environmental collision includes any element with which the player can collide. For example, grass, bamboo bushes. You should ensure all objects that the player can intentionally walk through have some form of collision sound.

**1.3 - Quests**

There are 2 quests in the game world, the first is the Leek Quest and the second is the Sword Quest. These quests test your ability to read more complex scripts and find points of interest to implement sound. The quests primarily consist of if statements with **sound required on object pickups and drop-offs**. You may also wish to **implement some dialogue on the characters.**

**1.3.1 - The Leek Quest**

The leek quest starts with the Market Seller at this location: 

The premise of the quest is to collect a box of Leeks from the Farm area and drop back off the market seller.

The Box of Leeks are located somewhere in this area.

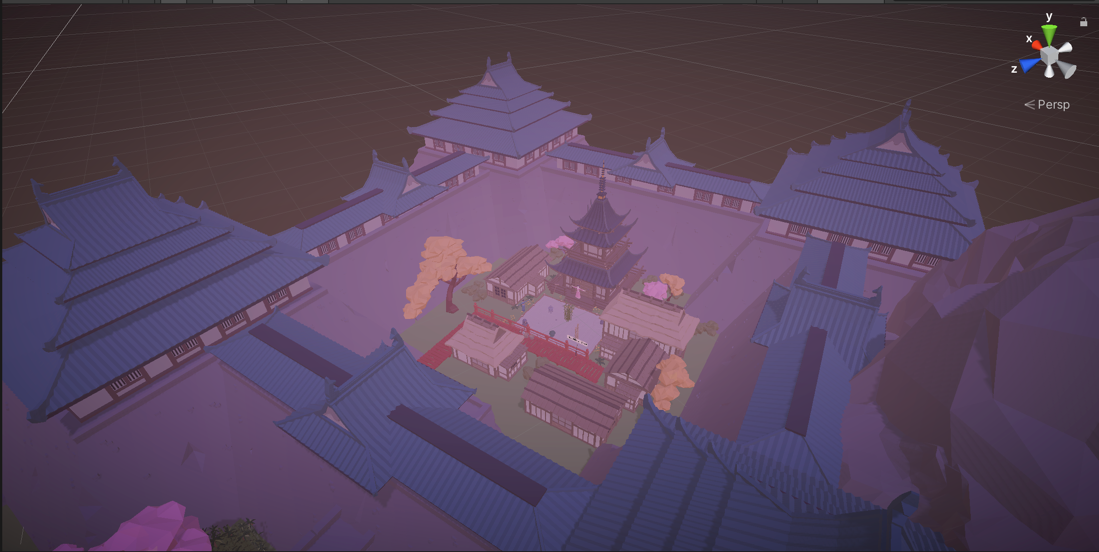


**Quest Bug!** There is a bug in the quest that allows the player to recollect the Leeks from the Farm. You should look to solve this issue by looking at the Sword Quest or by thinking about basic if statement logic.

**1.3.2 - The Sword Quest**

The Sword Quest does not have an obvious start point and you may wish to create a starting point by utilizing one of the characters.

The Quest should guide to the player to the location of the Emperor’s Sword located somewhere in this area:



The sword should then be dropped off to the emperor himself at this residence:



**1.3.3 - Dialogue & UI**

Dialogue can be implemented on the characters during key stroke activations/character interactions. You should aim to capture quality dialogue for the highest marks, although you may utilise quality text to speech or other solutions.

Alternatively, the dialogue could be completely gibberish in a Nintendo-like fashion with some text appearing on-screen. Implementing text on the UI is extremely easy and you should look to research this on your own.

**1.4 - Footsteps**

The footstep system is possibly the most complex part of this assessment with several steps needed to design and implement a fully immersive and natural playback and switching system. The player uses an animator to move which should aim to call the sound. The player can also jump, walk, and crouch and you may wish to implement solutions for these as well to push your grade further.

**1.4.1 - Animations**

The animation system in Unity is very comprehensive allowing the full control of almost any aspect of an object over time. You can use this animation controller to call a function in script and playback a sound that is time matched to the footstep landing position.

**1.4.2 - Surface Switching**

During the module you will explore several different solutions to footsteps and surface switching. Which surface switching solution you utilise is up to you, however, it should naturally change based on the player contact with a particular surface. This can be done by checking the tag, layer, or physical material. In the module we have covered the tag solution however there are plenty of guides online to explore other solutions if you wish.

It is recommended after significant testing to use a RayCast **not a collider** for the information retrieval from the object the player is contacting.

**1.4.3 - Jumping**

The player can jump using the Space Bar and should have a start and landing jumping sound. You should look to create a similar system to the footstep script and playback with surface switching.

**1.4.4 – Footstep State Switching**

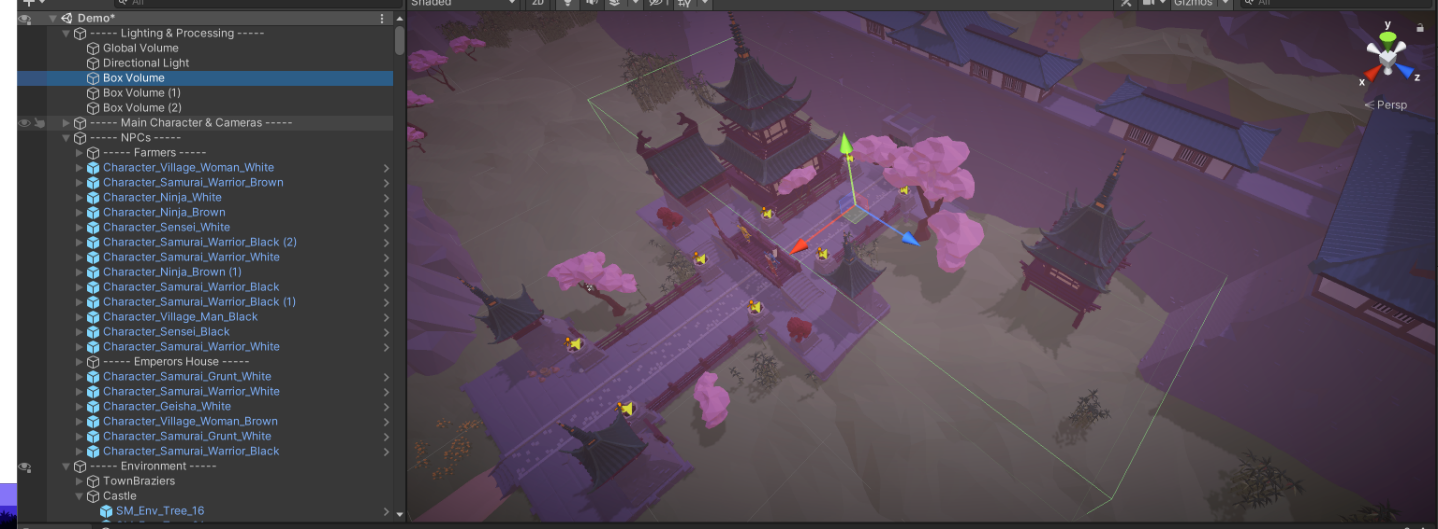
The player can Run, Walk or Crouch, with these states needing slightly different sounds. Due to the way the blend tree works the function call from the Animator can accidentally call walk or run when not in that state.

It is far more effective to use a system utilising the player velocity to gauge what speed the player is going and then switching the volume parameter to your Run, Walk and Crouch sounds. You could also help gauge the players state by looking for the corresponding keycode.

**1.5 - Volumes (Rendering)**

There is an interactable Rendering Volume in the game near the entrance to the main castle. When the player enters this volume the rendering to the camera will change with a UI element appearing on the screen. You should look to utilise a master bus effect on the FMOD master bus to signify this change as well as the addition of another sound such as a heartbeat or similar.

The volume is located in -----Lighting & Processing ------. The screenshot below shows the selected Box Volume.

There is a script for the control of the volume called **NoPlayerZone.** 

**1.6 - Music**

Some tips on how to implement music in the level:

* Trigger start the music upon arrival to the town.
* Loop sections until items are collected in the quests.
* Add drums the closer you are to the castle.

# Mixing and Project Finalisation

Mixing your game is similar to the way you would approach post-production audio in film, however, the difference here is the majority of the sound is interactive and exists in a 3D space.

**2.1 - Mix Considerations**

When mixing you should consider both the real-world amplitude of the sounds surrounding you as well as the volume of objects/elements that you wish to stand out such as speech or environmental collision which is typically slightly louder than in real life in a lot of games (but not always).

The mix will feel balanced if:

* Some objects are not wildly louder than others.
* Reverb is at a natural level in the relation to its location.
* Speech is natural sounding, no EQ coloration, and sits above the main mix but just enough to be heard clearly and should not dominate.
* Music lowers in volume during speech. You could use an if statement and an FMOD parameter here to find out when dialogue is active and if so, lower volume, if not, normal volume.
* Music is not overpowering and sits just below the main environmental sounds. To push your grade further, you could investigate creating a music volume switcher using the + & - keys.
* Sounds should not be compressed heavily, and dynamic range should be as natural as possible. Compression can be used for speech to help consistency.
* The overall volume should be approx. 18-25 LUFS in regular situations but can reach much higher levels during action or similar. Assassins Creed Odyssey for example as an average LUFS of 26 LUFS with speech elements and action coming in louder at approx. 15-20 LUFS. This is just a guide and there is no regulation set in stone and use common sense when choosing an appropriate loudness.

This is by no means a complete guide to the perfect mix but tools to follow to help guide toward a good mix. There is no replacement for listening and tweaking. Mixing is not something you can rush but something that will require you to take your time.