

Integrating Wwise with Unity

1. Understanding the Wwise/Unity Integration and Why it's Important

Before diving into integrating Wwise into Unity, it's important to understand how the two software programs work together, and why it is important. Wwise is primarily used for audio creation and mixing, while Unity is used for game development. The integration between the two programs allows for audio to be seamlessly integrated, edited, and mixed into Unity games.

The integration process involves creating something called “audio events” in a Wwise project and then linking them to Unity game objects using Wwise audio sources. This allows for more dynamic audio playback in Unity- Wwise sounds can be triggered by specific game events, such as an enemy appearing on screen or the player collecting a power-up.

KEY FEATURES AND DEFINITIONS:

- **Audio Events** - the heart of Wwise's interactive audio system. An audio event is essentially a container for one or more audio files, along with associated parameters and settings that define how the audio files will be played back in the game. For example, you might create an audio event for a gunshot sound effect. Within the audio event, you could specify the audio file for the gunshot sound effect, along with parameters such as volume, pitch, and spatialization settings. Audio events can be triggered by game events, such as when the player fires a weapon or when an enemy is defeated. When an audio event is triggered, Wwise will play back the audio file(s) within the event.
- **SoundBanks** - containers for audio assets that are used by Wwise to generate the final audio output for the game. When you create an audio event in Wwise, it is not immediately playable in the game. Instead, it must be included in a SoundBank. You create a SoundBank by using Wwise's SoundBank generator, which takes all of the audio assets that have been included in the project and compiles them into a single, optimized package. This SoundBank can then be loaded by Unity at runtime and used to play back the associated audio events. By using SoundBanks, Wwise can optimize the loading and playback of audio assets in the game. The SoundBank generator can compress and package the audio assets in a way that maximizes performance and reduces memory usage.
- **Game syncs** - used to synchronize audio events with game events in real-time. You can define a game sync within the Wwise project and link it with one or more audio events. When

Unity triggers the game sync, Wwise will play the associated audio event in real-time, creating a more dynamic and immersive audio experience for the player. For example, triggering music changes or sound effects based on the player's actions, or in response to environmental events like thunderstorms/earthquakes, or ground material changes.

- **RTPC** (Real-Time Parameter Control) - a key feature of Wwise that allows you to control audio properties in real-time during gameplay. RTPC allows you to adjust audio properties such as volume, pitch, and filter settings based on the state of the game, such as the player's position or the level of ambient noise. RTPC works by defining a parameter within an audio event that can be controlled by the game engine. For example, you might create an RTPC parameter for the intensity of a thunderstorm over time, which can be used to control layering/crossfading multiple sounds (soft/loud rain), and parameters such as volume, distortion, or filtering of the audio source.

2. Organizing Your Audio Assets in Wwise

One of the key benefits of using Wwise is the ability to organize and manage large amounts of audio assets in one place. It's important to take advantage of this feature and keep your audio assets well-organized in Wwise. If you are working on a large game with many audio assets, using Wwise would make it less of a headache to keep track of these sounds and how they interact with each other.

You can create folders in Wwise to group related audio assets together, such as music tracks, sound effects, and dialogue. Even further, you can have subfolders with categories such as weapons, enemies, and players. You can also use Wwise's tagging system to easily find and manage specific audio assets.

3. Creating Interactive Audio in Wwise

Interactive audio is an important part of game audio design, and Wwise offers many powerful tools for enhancing interactive audio. One of the most powerful features of Wwise is the ability to create dynamic audio events that respond to in-game events.

For example, you can create an audio event that plays a different sound effect depending on the distance between the player and an enemy. You can also create an audio event that changes based on the player's actions, such as playing a different sound effect when the player jumps or attacks.

4. Using Wwise's Audio Bus System

Wwise's audio bus system is one of the more powerful features that allows you to control the mix and volume of your game's audio in real-time. With audio buses, you can group related audio assets together and apply effects and volume adjustments to them as a group.

For example, you can create an audio bus for all the music tracks and apply the same reverb effect to the entire group. Moreover, you can create another bus and route all the dialogue to it. You may want the music and the dialogue to have different reverbs in the game, so this can be a useful tool. The goal is creating a cohesive and immersive audio experience for your players, and audio busses definitely are key in that.

5. Optimizing Your Audio Assets for Performance

While Wwise and Unity allow for a high degree of audio complexity in games, it's important to keep in mind the performance impact of your audio assets. Large and complex audio assets can have a significant impact on your game's performance and load times.

To optimize your audio assets, consider compressing your audio files, using shorter sound effects, and avoiding excessive layering and effects processing. You can also use Wwise's profiling tools to identify performance bottlenecks and optimize your audio accordingly. The Profiler provides a real-time view of your game's audio activity, including how much CPU and memory each asset is using. It also allows you to see how many instances of each sound effect are playing at any given time. This is particularly useful for identifying assets that are using too much memory and causing performance issues.

6. Testing Your Audio Design in Real Time

Finally, it's important to note that integrating Unity and Wwise allows the sound designer to alter their audio in real time, experiment, try new things, and create different iterations of assets. The connection allows all of Wwise audio effects to be changed while the game engine is running - eliminating the need to pause, change a parameter, and replay.

HOW TO GET STARTED:

1. Download and install the Wwise Authoring tool from the Audiokinetic website.
<https://www.audiokinetic.com/en/products/wwise/>
2. Create a new Wwise project for your game and import your audio assets into the Default Work Unit.
3. Install the Wwise Unity Integration package in your Unity project.
4. Open the Wwise Launcher and generate a SoundBank for your project.
5. Import the SoundBank and the WwiseSettings.asset file into your Unity project.
6. Add the Wwise Global object to your Unity scene.

7. Use the Wwise Picker window in Unity to select and add audio events to your game objects.
8. Customize the audio settings and properties in the Wwise Authoring tool to achieve the desired audio experience in your game.
9. Test and refine the audio integration as needed.

BONUS: Wwise has lots of easy-to-follow tutorials for extra help – they make it super easy to learn more about specific things that I could not fit into a 10-minute presentation! 😊

https://www.audiokinetic.com/en/courses/wwise101/?source=wwise101&id=quick_start_from_silence_to_sound