READ ME

Audio system

This **Audio System** provides a flexible and centralized way to manage sound in your Unity projects. It is built to simplify the handling of sound effects, music, and ambient audio by offering clean APIs for playback, volume control, and event-driven audio triggers. Instead of scattering audio logic across multiple scripts, the system brings everything into one cohesive framework that is easy to extend, debug, and maintain. Whether you are building a small prototype or a large-scale production, this system helps keep your project's audio organized, scalable, and performant.

System Structure

Core Runtime

AudioManager.cs

Per-SoundClipSO Control

Each sound is defined as a SoundClipSO (with its own output group, volume, loop settings, and clip selection mode). The AudioManager uses these objects as keys for playback, so you can stop, pause, or play multiple instances of the same logical sound without hard-coding clip references.

Emitter Pooling

Instead of creating and destroying AudioSources at runtime, the manager recycles a pool of SoundEmitter components. This keeps performance stable (no garbage collection spikes) and guarantees consistent audio behavior even during heavy sound usage.

Separation of Concerns

The AudioManager only coordinates pooling and playback. Mixer outputs and per-sound settings are stored in SoundClipSO, keeping the manager lean and making it easy to tune sounds without changing code.

AudioReferenceSO.cs

A ScriptableObject that acts as the central registry for the audio system. It automatically collects and lists all available SoundClipSO assets, providing an overview and easy editing of sound clips in one place. In addition to storing the clips themselves, it also:

- Holds a reference to the Audio Mixer.
- Stores the volume data for each output, allowing fine-grained control and balancing directly from the asset.

This makes it the backbone of the system's configuration — keeping all sounds and their mixer routing organized in a single asset.

SoundClipSO.cs

- ScriptableObject that defines individual sound settings (clip, volume, pitch, looping).

SoundEmitter.cs

- Component that plays sounds via pooled AudioSource instances for efficient performance.

Editor Extensions

AudioReferenceSOEditor.cs

- Custom inspector for AudioReferenceSO assets.

AudioReferenceWindow.cs

- Dedicated window for browsing and managing soundClips.

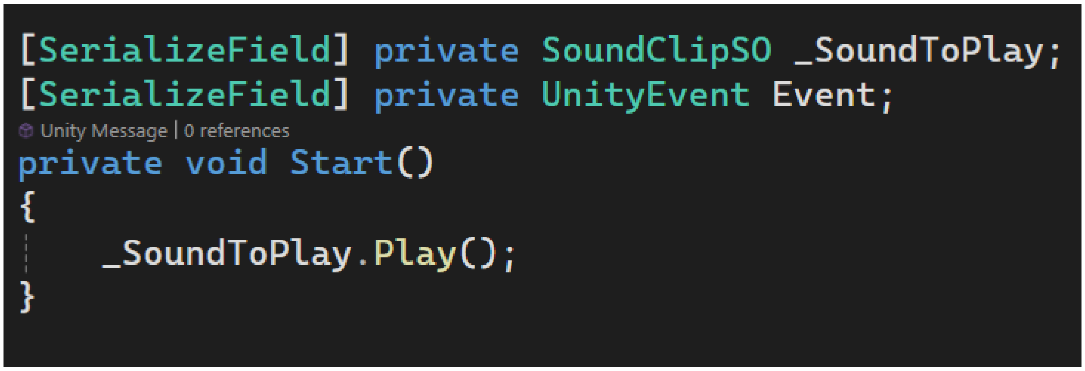
EditorAudioPlayer.cs

- Utility for previewing soundClips directly in the Editor.

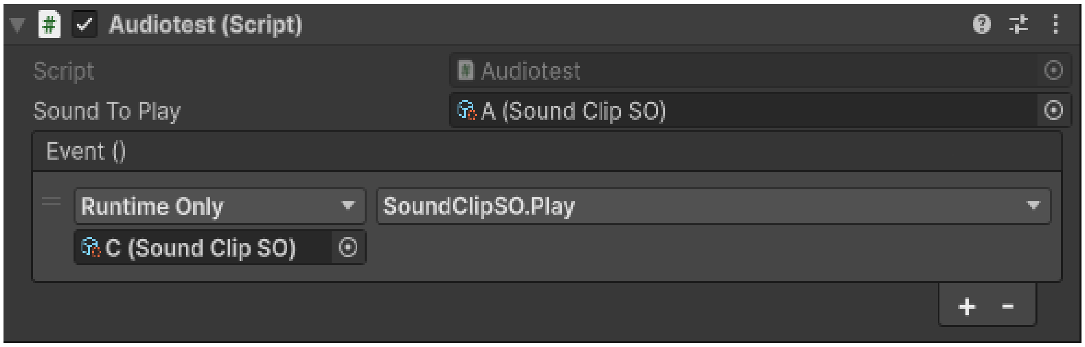
Quick Start

1. Drag AudioManager.prefab from Prefabs folder into your scene.
2. Create a AudioReference, find the folder you want to save it and right-click => Create => Audio => AudioReference

3. Create a SoundClip, Find the folder you want to save it and right-click => Create => Audio => SoundClip
4. Play from Script or event



```
[SerializeField] private SoundClipSO _SoundToPlay;
[SerializeField] private UnityEvent Event;
Unity Message | 0 references
private void Start()
{
    _SoundToPlay.Play();
}
```



Audiotest (Script)

Script: Audiotest

Sound To Play: A (Sound Clip SO)

Event ()

Runtime Only | SoundClipSO.Play

C (Sound Clip SO)

Demo scene

A working demo scene is included to showcase how the Audio System works out of the box.

Location

- ExampleScene/Demo.unity

Contents

AudioManager.prefab

- in the scene, configured to manage audio playback.

AudioTest.cs

- Shows that SoundClipSO can be played from script and event.
- Uses a coroutine to play a sound clip every 2 seconds.

Camera

- For the audio listener

License

This project is released under the **MIT License** when distributed through GitHub. You are free to use, modify, and integrate it into both personal and commercial projects. See the LICENSE file for full details.

When downloaded from the **Unity Asset Store**, usage is covered by the standard **Unity Asset Store EULA**. This grants the right to use the asset in your Unity projects but does not allow redistribution outside the Asset Store.