Simple Audio Manager

User Guide

Table of Contents

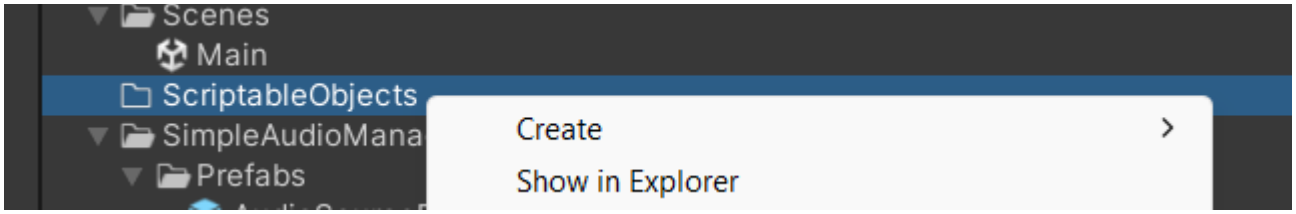
Audio Manager Configuration

Creating Songs

“Songs” contain a collection of audio clips known as “Intensity”. This configuration is designed to allow you to blend between multiple variations of the same composition at your discretion. Additionally, Songs use a field called “Reverb Tail” to allow for seamlessly looping tracks prior to their completion. In order to use the Simple Audio Manager, you must create one or more Songs with each containing at least one Intensity audio clip.

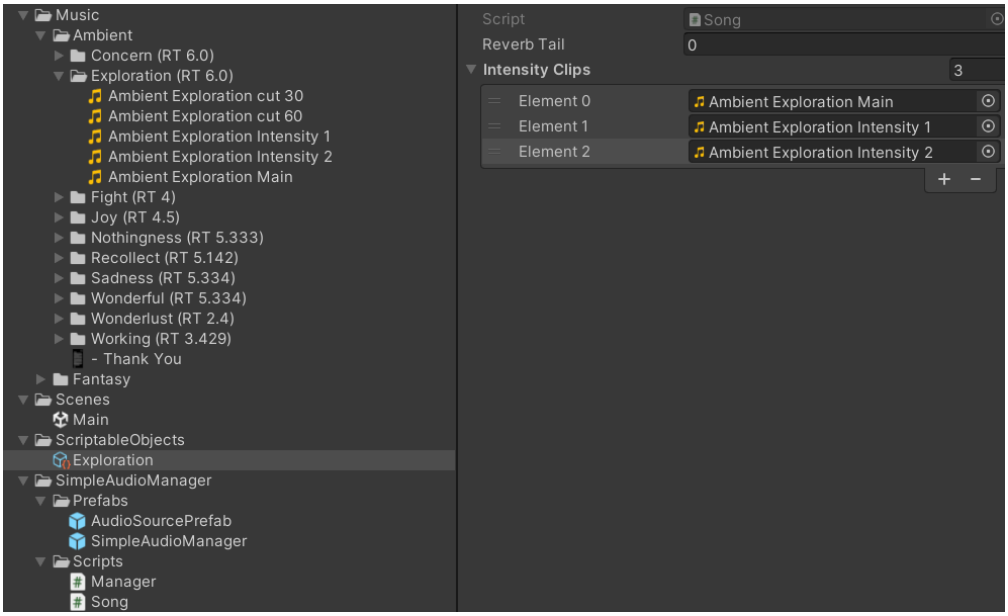
To create a new song:

1. Right Click inside of your project window on the desired folder.

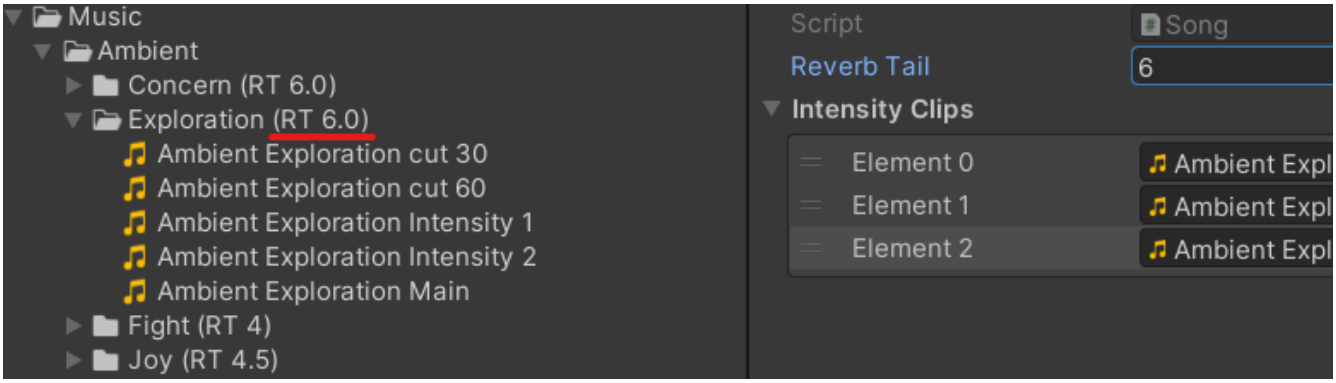


2. Select “Create→SimpleAudioManager→Song” from the context menu.

3. Populate the “Intensity” list with the variations of audio clips.



4. Optionally specify the Reverb Tail. (For Ovani Sound Assets, you will see the RT specified in parentheses at the end of the composition folder)

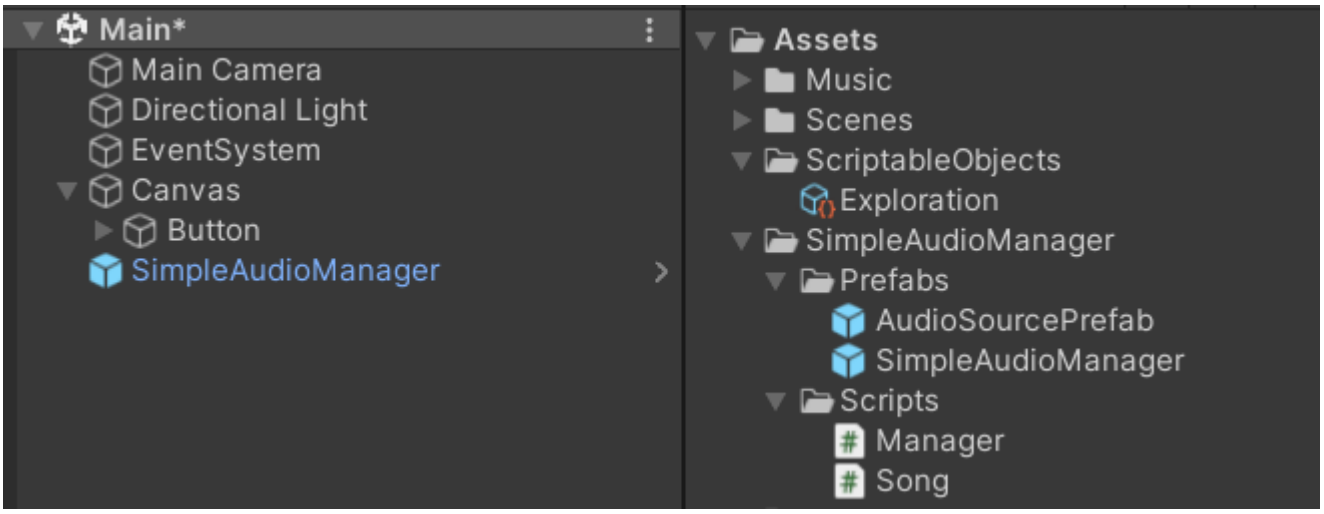


Applying Songs to the Manager

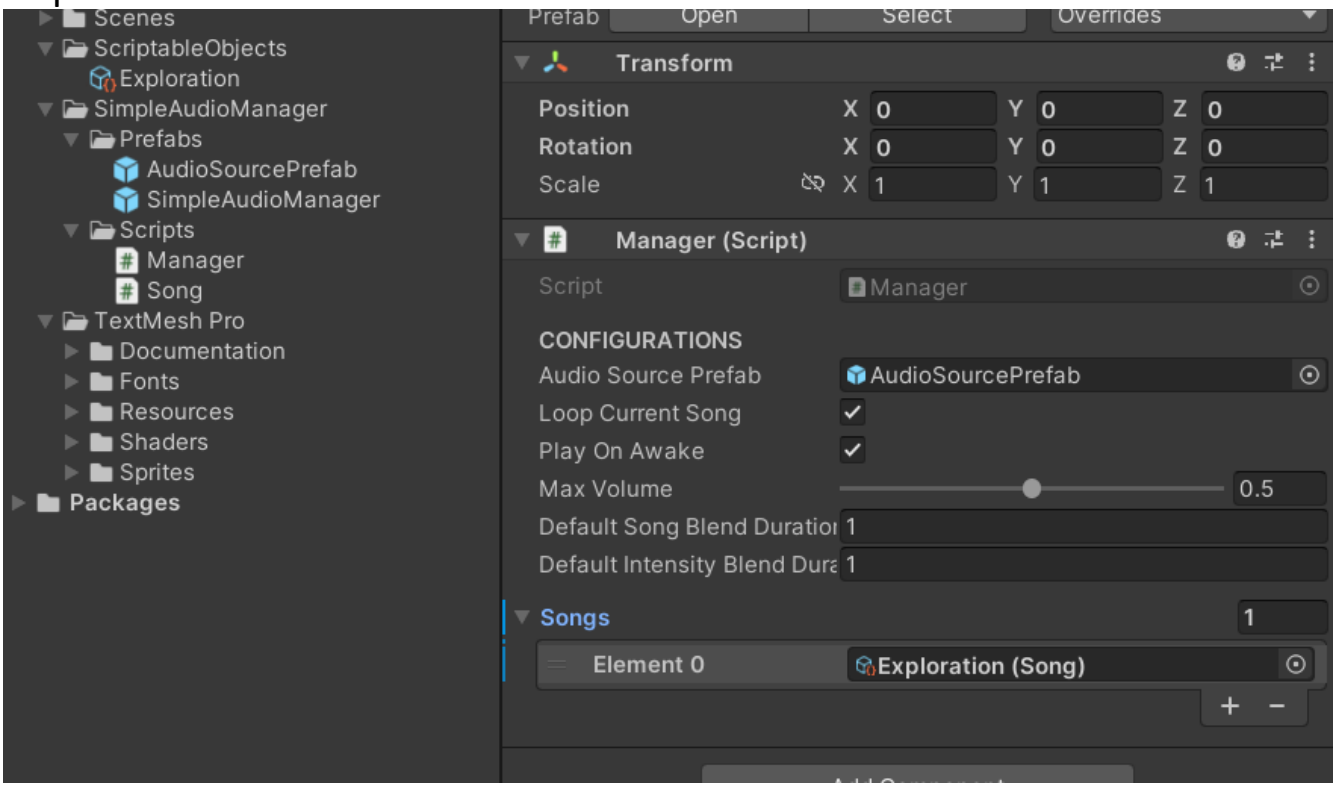
Once you have created at least one Song, you can apply it to the manager. This is what allows you to play songs at runtime.

To apply the Songs to the Manager:

1. Drag & Drop the “SimpleAudioManager” prefab into your scene.



2. Select the Simple Audio Manager and begin Dragging & Dropping your Songs onto the “Songs” list located at the bottom of the component in the inspector.



At this point, your Songs are ready to be played from the manager.

Explaining the Manager and the Options

The Simple Audio Manager is designed to be a simple but elegant solution for managing small-scale projects or simple audio needs. As such, the options are extremely limited and easily explained.

Audio Source Prefab

This is the prefab used in the pooling pattern. It may be replaced with a custom prefab as long as the top level object contains an Audio Source.

Loop Current Song

This toggle determines whether or not the current Song being played will loop when it reaches the end of its play time.

Play On Awake

This toggle specifies whether or not the manager will play the first Song in the list when gameplay begins.

Max Volume

This slider specifies the volume of the clip once it has completed blending into play.

Default Song Blend Duration

This is the default amount of time it will take for the previous Song to fade out and the next Song to fade in when changing Songs. **This does not apply when looping or when modifying the Intensity.**

Default Intensity Blend Duration

This is the default amount of time it will take for the previous Intensity to fade out and the next Intensity to fade in when modifying the Intensity.

Using the Audio Manager

Playing a Song

To play an available Song with the Simple Audio Manager, simply call a variation of the “Play Song” method.

Variation 1:
PlaySong (*int* pSong) : This method will call the following Play Song method with all default values sans the song index. This method is designed for quick calls from the inspector with Unity Events.

Variation 2:
PlaySong (*PlaySongOptions* pOptions) : This method accepts a struct called “PlaySongOptions”. The struct has the following modifiable fields:

- “**song**”: Used to specify the index of the Song in the Song list.
- “**intensity**”: Used to specify the target intensity for the target Song.
- “**startTime**”: Used to specify the point in the clip at which to begin.
- “**blendOutTime**”: How long it will take for the previous Song to fade out.
- “**blendInTime**”: How long it will take for the next Song to fade in.

NOTE: Please note that playing a song or modifying the intensity (which is an alias for “PlaySong”) uses an object pooling pattern which will create new gameobjects if the existing audio sources are busy playing clips.

Stopping a Song

To stop a song, simply call the “Stop Song” method.

StopSong() : Stops the current audio source as well as the looping coroutine.

Changing the Intensity of the Current Song

To change the Intensity of the current Song, call one of the two variations of the "Set Intensity" method.

Variation 1:
SetIntensity (*int* pIntensity) : This method calls the second variation of this method with default values sans the intensity provided. This method is designed for quick calls from the inspector with Unity Events.

Variation 2:
SetIntensity (*int* pIntensity, *float* pBlendOutDuration, *float* pBlendInDuration) : This method calls the "Play Song" method but passes in a default start time of 0 seconds as well as the supplied blend durations.

Miscellaneous Properties

SimpleAudioManager.Manager.instance → returns the singleton instance.

Within the Manager Class

audioSourcePrefab A reference to the prefab used for object pooling.

clipTimeRemaining The time before either a non-looping clip ends or the next loop of a looping clip begins.

Within the Song Class

reverbTail The time before the end of a clip where a new loop of the clip will begin to play.

intensityClips The list of audio clips which make up the varying intensities of the Song.

Notes

Thank you for purchasing the Simple Audio Manager plugin.
We greatly appreciate it!

Please remember that this tool was designed with simplicity and ease-of-use in mind. We strongly recommend implementing more complex or dynamic tools if greater flexibility is required. Alternatively, if you like how we built the tool, feel free to use it as a starting point for your own custom audio solution.

Keep an eye out for more Ovani Sound plugins in the future.
If you have any feedback or suggestions, feel free to reach out to us!