



CUTTLE

CUSTOM AUDIO TOOLSET FOR UNITY3D

What is Cuttle?

Cuttle is an audio system built upon Unity's native audio system and API. While traditional middleware options like FMOD and Wwise introduce a new and separate workflow for audio implementation, Cuttle expands your existing Unity workflow with more audio capabilities and also allows programmers to jump in and change things up to suit their specific needs.

Cuttle Features

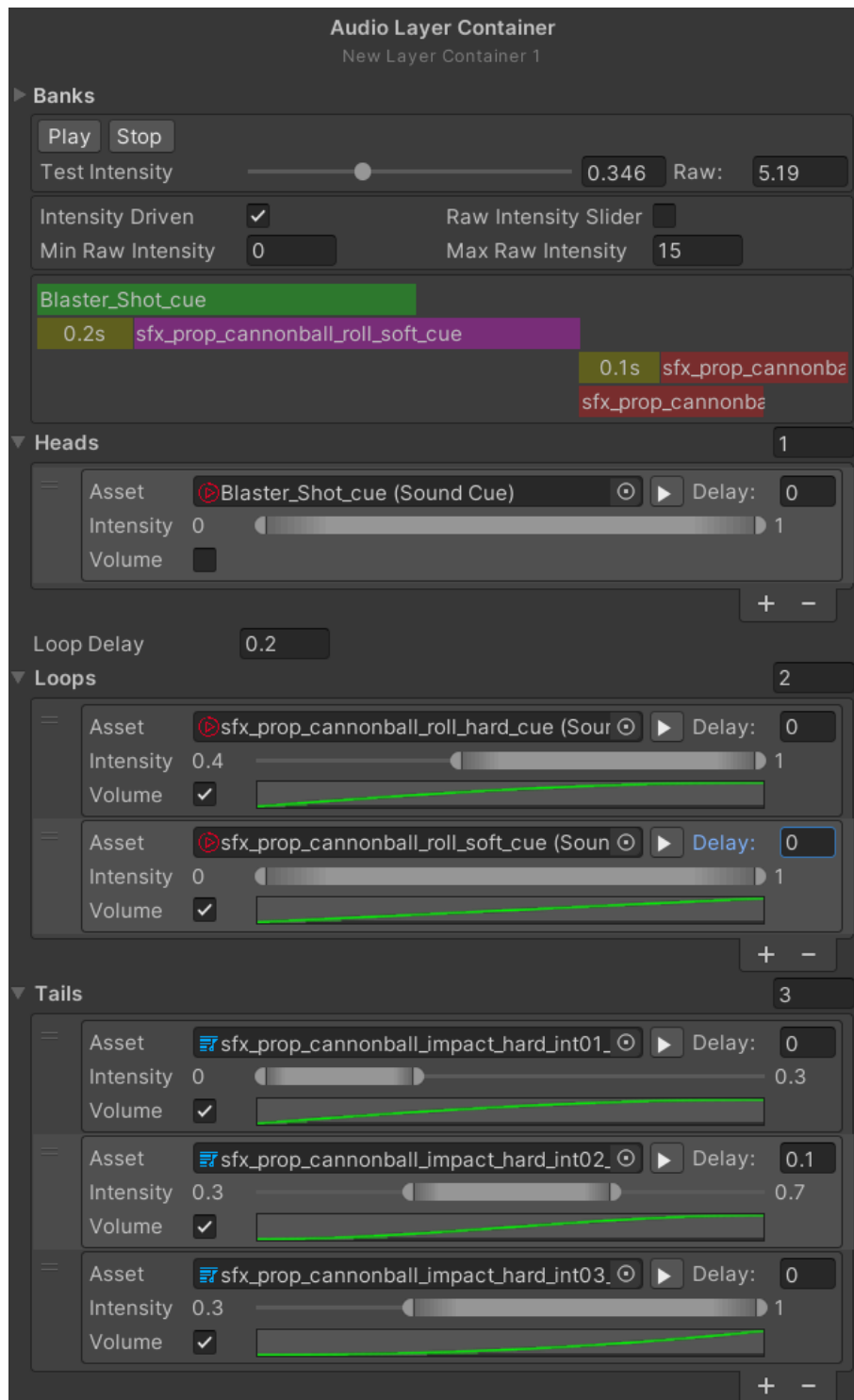
- SoundCues
- Containers
- Real-Time Parameter Controls
- Audio Occlusion
- Dynamic Reverb
- Real-Time Debugging
- AudioSource, Pooling & Voice Management
- Audio Memory Management & Sound Banks
- Digital Signal Processors
- Poly Lines

SoundCues and Containers

Control audio events at the asset level without having to go into and make adjustments to scenes and scripts using our SoundCues and Containers.

SoundCues builds upon the properties of a Unity AudioSource and add functionality such as:

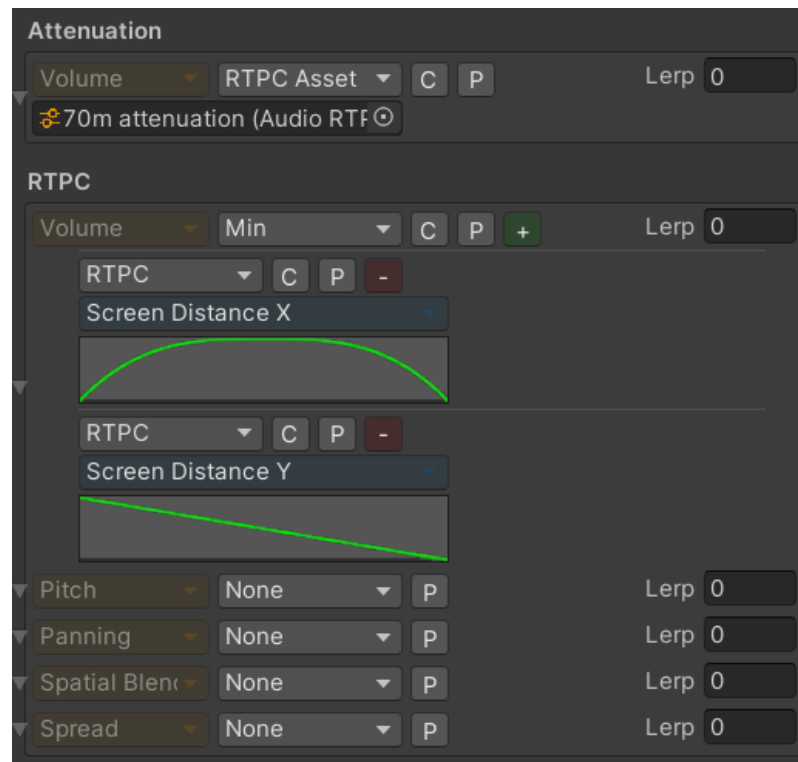
- Random Offset
- Fade In
- Fade Out
- Voice Limit
- Cooldown
- Delay
- Occlusion
- RTPCs



Containers define how a SoundCue is selected and triggered. Each Container applies the following logic:

- Random: randomises cue selection and its pitch and volume
- Scatter: a Random container that also randomises trigger interval and audio source position
- Minigun: a Random container that also moves the sound in one or more directions
- Layer: plays multiple sounds together, with optional lead-in and lead-out layers. The mix of the layers is driven by an intensity parameter

Real-Time Parameter Controls (RTPCs)



Dynamically change a sound in real time, driven by in-game parameters, with our **RTPCs** that don't require any hardcoding to set up.

- Dynamic real-time changes, driven by game parameters
- Audio attenuation
- Stackable RTPCs for complex designs
- Can be shared, or copied and pasted for easy replication across SoundCues

Audio Occlusion

Use **Audio Occlusion** to muffle or block sounds when there are walls or barriers between the sound source and listener.

- Natural sounding occlusion
- Light on system resources
- Smooth transitions
- Can be used to drive other parameters
- Can accommodate different material types

Dynamic Reverb

Use **Dynamic Reverb** to help situate the player in the world around them. With minimal setup, players can hear realistic reverbs that dynamically change as they move from one space to the next.

- Ready to use dynamic reverb system
- Instant reverb coverage without the need to set up numerous reverb zones
- Easy to customise via config files or directly in code
- Can accommodate different material types
- Can accommodate manual reverb zones for special zones

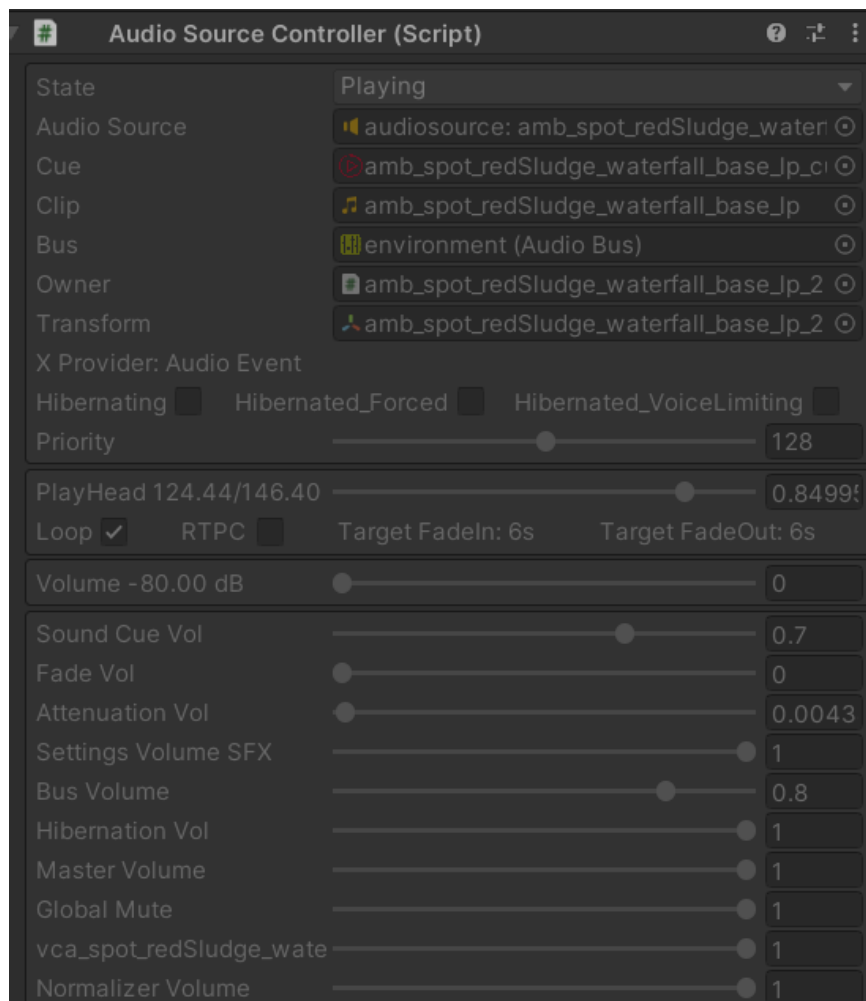
Real-Time Debugging

TEXT FILTER:		BUS FILTER:		None (Audio Bus)		SORT: None		include virtual		final volume		reverse		unsolo all		unmute all		logger		GUI					
SHOW:		clip		owner		loop		bus		distance		transform		priority		pitch		pan		spatial blend		spread			
PLAYING LIST [7/32]																									
> M S		voxJp_01_cue_delayTest				voxJp_01				vox (2)		3D 1.0		loop		24.27 / 42.2s		25.22m		-12.60 dB					
> M S		voxJp_01_cue_delayTest				voxJp_01				vox		3D 1.0		loop		24.27 / 42.2s		24.42m		-12.11 dB					
> M S		voxJp_01_cue_delayTest				voxJp_01				vox (1)		3D 1.0		loop		23.62 / 42.2s		26.32m		-80.00 dB					
> M S		Blaster_Shot_cue				Blaster_Shot				Weapon_Blaster(Clone)		2D 0.0		oneshot		0.45 / 1.5s		0.57m		-2.19 dB					
> M S		Blaster_Shot_cue				Blaster_Shot				Weapon_Blaster(Clone)		2D 0.0		oneshot		0.32 / 1.5s		0.57m		0.00 dB					
> M S		Blaster_Shot_cue				Blaster_Shot				Weapon_Blaster(Clone)		2D 0.0		oneshot		0.21 / 1.5s		0.57m		0.00 dB					
> M S		Blaster_Shot_cue				Blaster_Shot				Weapon_Blaster(Clone)		2D 0.0		oneshot		0.11 / 1.5s		0.57m		-0.77 dB					
AUDIO LOGGER [61]																									
		Recording		clear		TEXT FILTER:																			
#57		122.965		932		OnEditorApplicationFocus(False)																		>	
#56		122.859		929		Blaster_Shot_cue		Weapon_Blaster(Clone)		Starting Hibernating (Voice Limiting)! immediate: False, already hibernating? False														>	
#55		122.859		929		Blaster_Shot_cue		Weapon_Blaster(Clone)		Played.														>	
#54		122.859		929		Blaster_Shot_cue		Weapon_Blaster(Clone)		Fading out!														>	
#53		122.752		925		Blaster_Shot_cue		Weapon_Blaster(Clone)		Played.														>	
#52		122.645		920		Blaster_Shot_cue		Weapon_Blaster(Clone)		Played.														>	
#51		122.517		916		Blaster_Shot_cue		Weapon_Blaster(Clone)		Played.														>	
#50		121.408		871		Weapon_Cooling_cue		Weapon_Blaster(Clone)		Stopping!														>	
#49		121.280		866		Weapon_Cooling_cue		Weapon_Blaster(Clone)		Fading out!														>	
#48		120.911		849		Blaster_Shot_cue		Weapon_Blaster(Clone)		Stopped!														>	
Frame: 932 DSP Clock: 122.965 Memory: 5.5 MB Real: 6/16 Virtual: 1/16 Compressed: 2 Streaming: 0																									

Easily identify what sounds are triggered at any given moment with our **Audio Debug Window**. Find, sort and monitor specific sounds for debugging and mixing purposes.

- Real-time data display of frame count, audio memory usage, real/virtual voice count, and more
- Important information about SoundCues and AudioClips at a glance
- Direct links to the audio asset or GameObject
- Mute and Solo buttons for easy testing
- Sort by volume to see what sounds are most prominent at any given moment
- Name filtering to focus on certain sounds
- 3D GUI text for in-game debugging
- Independent logger that logs every audio event with stacktrace information

AudioSource, Pooling & Voice Management



Use our **AudioSourceController (ASC)** to have total control over audio voices and bypass the limitations of the Unity AudioSource API, including voice management.

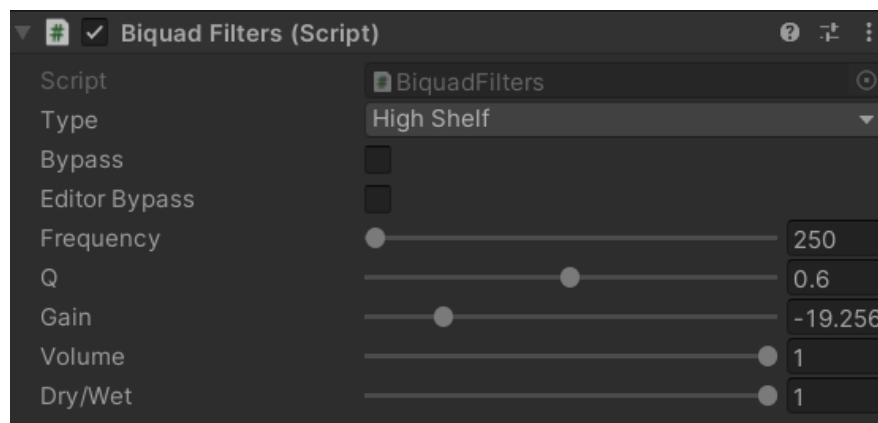
- Highly configurable playback calls for various situations
- Sensible state flags: Playing, Stopped, Standby
- Showing what script is responsible for a sound
- Transparency of volume stacking to see everything that contributes to a sound's final volume
- Callbacks on Play, Stop, Pause, FadeOut and Virtualize/Devirtualize, allowing game events to react to a sound's playback status
- Built-in voice management system that can accommodate large games and also maintain a small AudioSource pool
- Fadable virtualization that avoids hard cut-offs due to voice limit
- Flexible voice limit that can be applied to any group of sounds

Audio Memory Management & SoundBanks

Take advantage of automatic **Audio Memory Management** with Cuttle, and assign SoundCues to designated SoundBanks for manual loading and unloading.

- Timed audio clips, where clips are automatically unloaded to release memory if they stay unplayed
- Management of clips so they are not loaded and/or unloaded too frequently
- SoundBanks where clips can be assigned for manually loading/unloading
- Debug window to monitor what clips are loaded into memory in real-time

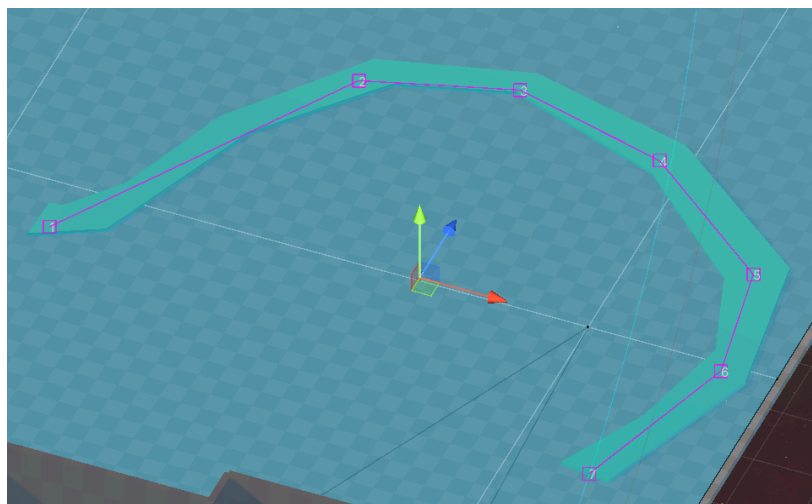
Digital Signal Processors (DSPs)



Expand the range of available audio effects in your projects with our custom DSP plugins. Unity has great DSP plugins but only for the AudioMixer. With Cuttle you can have additional AudioSource plugins as well.

- Applicable on a per-source level
- Bit Crusher, Tape Delay, and Biquad Filters currently available
- Efficient use of OnAudioFilterRead calls per source
- Synthesizer Mode to allow the DSP to occur before the original data is even rendered

Poly Lines



Use our neat PolyLine tool to create line-shaped audio emitters without having to place numerous audio sources.

- Minimal audio sources to cover long lines, like rivers and water banks
- Smooth, continuous playback of an audio source
- Freely move, add, or remove nodes from the polyline
- Inexpensive solution for lengthy emitter objects
- Can be used for non-audio assets as well