SOUND SHAPES

Welcome!

NEWS: Join the brand new Telepresent Games Discord Server!

Thank you for choosing the Sound Shapes! I hope it'll make a big difference in your game audio workflow!!

In this documentation, you'll find everything you need to get started, from installation to feature explanations and usage tips

Let's dive in!

Why Sound Shapes?

Real-world sound sources rarely originate from a single point. Flowing rivers, gentle rainfall, or industrial machinery all generate audio across extended areas, making realistic simulation difficult with traditional single-point AudioSources.

Sound Shapes lets you effortlessly simulate realistic audio by dynamically tracking sounds within user-defined areas—whether they're shapes, meshes, or points. Easily preview and refine your soundscapes directly within Unity's editor, with instant feedback respecting all your custom parameters.

Additionally, Sound Shapes includes an integrated occlusion system, adding depth and realism by accurately simulating how sound interacts with the environment. The intuitive, no-code-required interface ensures you can quickly create immersive audio experiences without hassle.

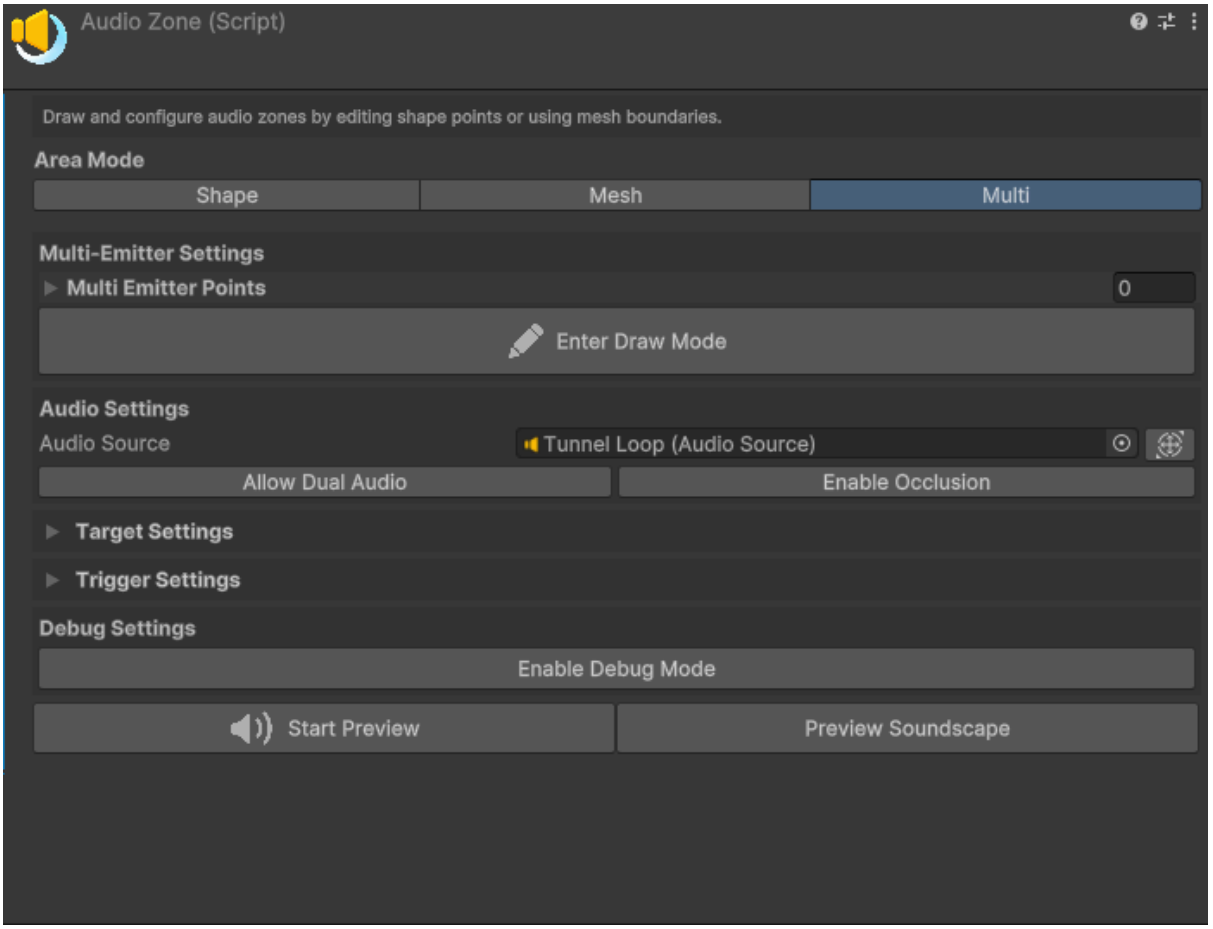
Installation

Not much to say here! Simply import the asset from the Package Manager, and you'll gain immediate access to the Sound Shapes Audio Zone System!

How To Use

Once installed, you'll be able to add the **Audio Zone Component** to your GameObjects in your scenes.

Audio Zone Component



The Audio Zone Component is the main system of Sound Shapes. It allows you to define an area in 3D space within which your AudioSource can travel, following its Tracking Target.

Area Modes:
The Area Mode defines the type of area you want your AudioSource to move within. There are three different Area Modes:

- Shape Mode
- Mesh Mode
- Multi-Emitter Mode

Shape Mode:
In Shape Mode, you'll be able to draw a series of points in 3D space. You can do this by manually adding them to the Points list, drawing them in 3D space using Drawing Mode (more info later), or during runtime with code.

Shape Mode has a unique "Closed Shape" toggle, which defines whether you are drawing a filled shape or merely a path. With Closed Shape toggled **On**, the AudioSource will be able to move within your defined shape. With the setting **Off**, the AudioSource can only follow the path you've defined.

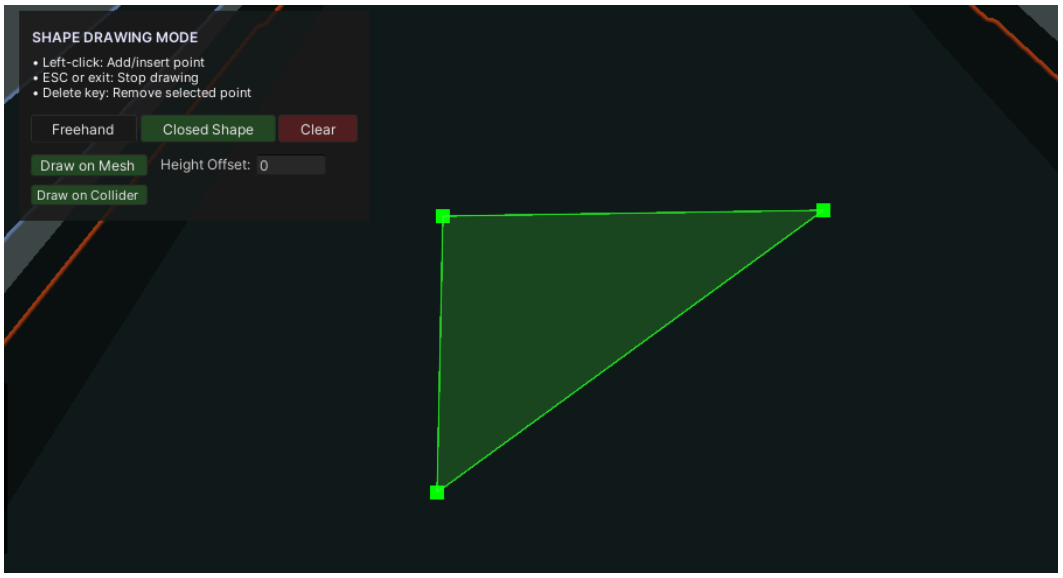
Mesh Mode:
In Mesh Mode, your AudioSource will be able to move all around the surface of one or more defined meshes from the Mesh Filters List. Depending on your mesh complexity, this is the most performance-intensive Area Mode, especially in Edit Mode.

Mesh Mode has a unique "Audio Offset" setting, allowing you to define a Vector3 position offset for your AudioSource. This might be preferable, especially when it comes to supporting proper occlusion behavior.

Multi-Emitter Mode:
In Multi-Emitter Mode, you can define a series of Multi-Emitter Points, either manually through the list, through Drawing Mode, or during runtime using code.

In Multi-Emitter Mode, an AudioSource is added to each of the defined points when the Player is in range and despawned once the Player is out of range.

Drawing Mode



In **Drawing Mode**, you're able to add points by drawing them directly in the scene. Simply click your desired point in the scene, and the system will automatically place a point directly onto the mesh at the clicked position—no colliders needed!

Drawing Mode has several settings you might find helpful:

- **Freehand Mode**: Allows you to hold and draw freehand to add multiple points in one stroke.
- **Closed Shape Toggle**: Toggles the Closed Shape setting in Shape Mode.
- **Clear**: Clears all points—don't worry, CTRL+Z works!

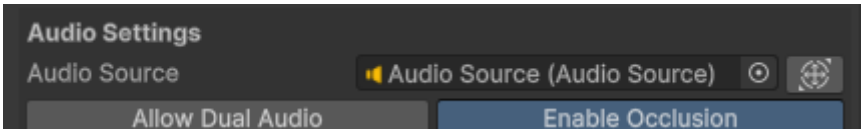
Additionally, you can adjust the behavior of Drawing Mode with these settings:

- **Draw On Mesh**: When enabled, allows you to draw directly on mesh data.
- **Draw On Collider**: When enabled, the system will prioritize and register points onto Colliders over Mesh data.
- **Height Offset**: Adds a vertical offset to any drawn point (useful for flat surfaces when using occlusion).

Tips:

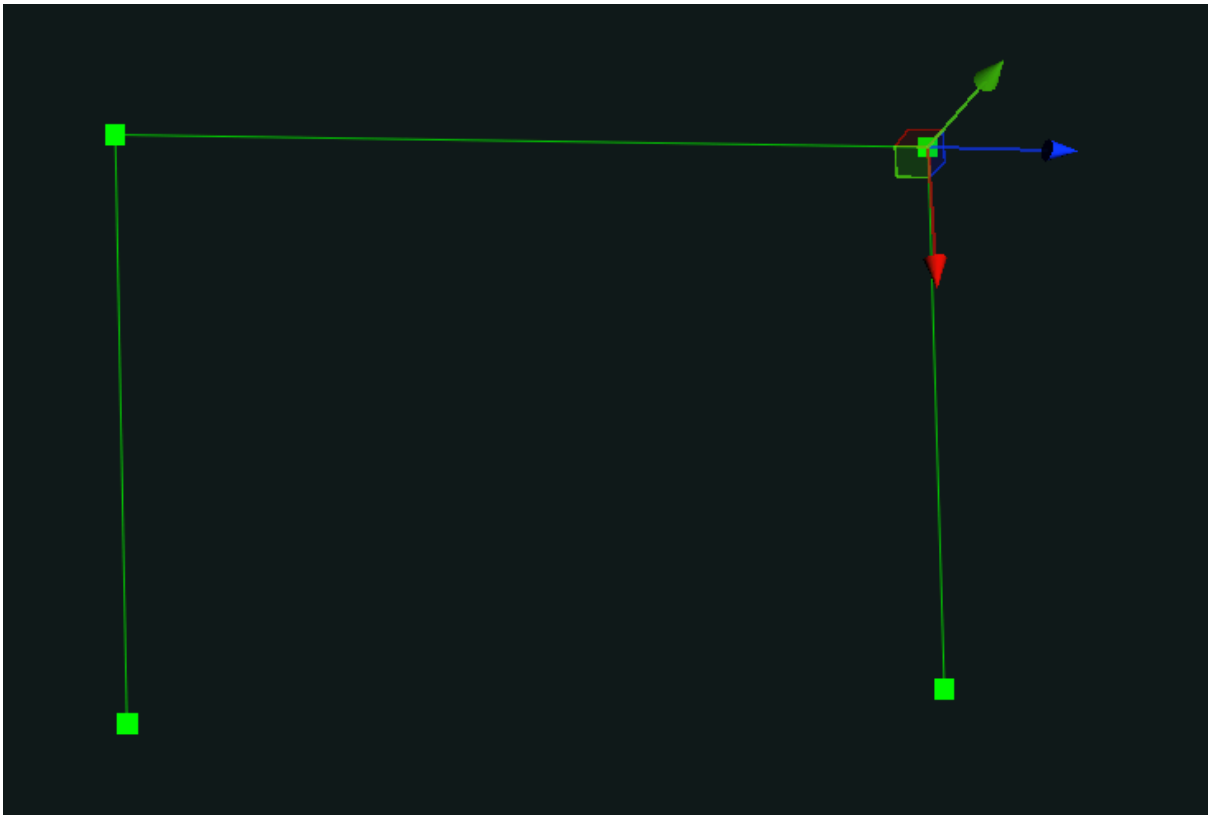
- **CTRL+Z** and **CTRL+Y** perform undo and redo actions.
- Individual points can be selected and dragged.
- Clicking on the line between two points will insert a new point between them.

Audio Source Setting



This Setting allows you to assign the AudioSource to the Audio Zone. The rightmost Button will select and focus on the AudioSource in the scene.

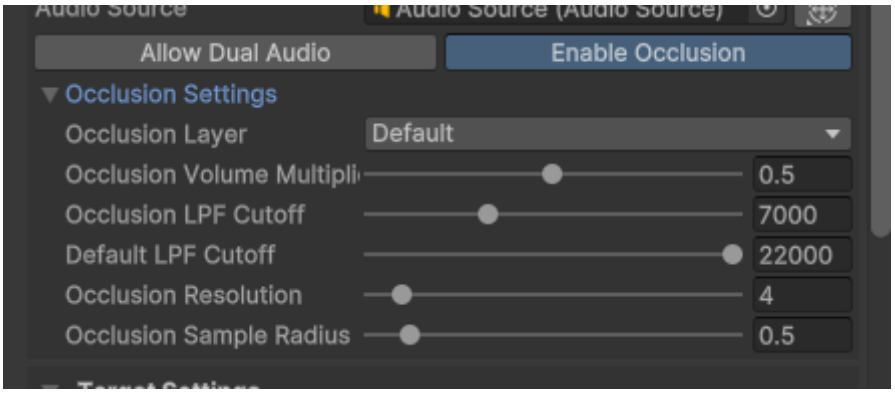
Dual Audio Setting



Dual Audio Mode is designed to support cases where your Audio Zone has two parallel points positioned to the left and right of your Player. In this scenario, the system will attempt to add an AudioSource to both sides, accurately simulating the player being surrounded by multiple audio sources.

This mode works best when **Closed Shape** is set to **Off**, especially in cases similar to the one shown above. However, Dual Audio Mode can produce somewhat unreliable results, particularly with closed shapes, so use this feature with caution.

Occlusion Setting



When enabled, the **Occlusion Setting** allows your AudioSource to automatically determine whether the AudioSource and Listener are currently separated by a Collider, muffling the audio accordingly.

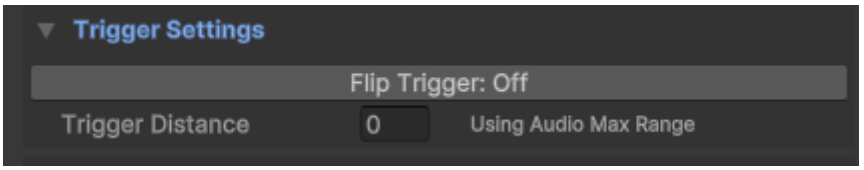
Within the Occlusion Settings, you can tweak the following:

- **Occlusion Layer**: Controls which layers the occlusion system checks.
- **Occlusion Volume Multiplier**: Controls the volume multiplier applied to the AudioSource when it becomes occluded.
- **Occlusion LPF Cutoff**: Adjusts the LowPassFilter cutoff frequency (the "muffle" level) applied to the AudioSource when occluded.
- **Default LPF Cutoff**: Sets the default LowPassFilter cutoff frequency for the AudioSource when not occluded.
- **Occlusion Resolution**: Controls how many raycasts are performed to determine the current occlusion level. Higher values provide more detailed and smoother occlusion effects.
- **Occlusion Sample Radius**: Defines the radius around which occlusion checks are performed.

Target Settings

These settings control which GameObject the system tracks for the Audio Zone. By default, it scans for objects with the **Player** tag, but it can also be configured to track other tags, as well as specific GameObjects. These settings can also be adjusted during runtime.

Trigger Settings

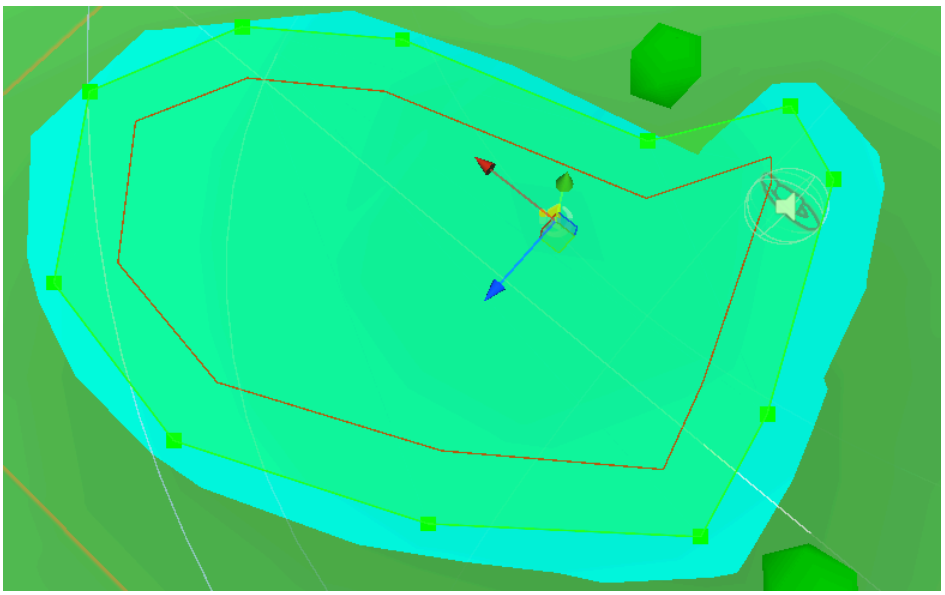


The **Trigger Settings** define how and when the Audio Zone should start and stop tracking its target.

By default, when **Trigger Distance** is set to `0`, the Audio Zone will use the **Max Distance** value from the assigned AudioSource component. This provides great results since it ensures tracking only occurs when the player is within earshot 😁.

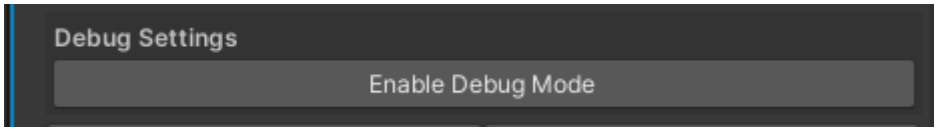
However, you can override the **Trigger Distance** manually if you'd like to tweak your own trigger range.

When working with a **Closed Shape**, an additional **"Flip Trigger"** toggle will appear. Toggling this setting reverses whether the Trigger range is inset or outset relative to the shape.



The Red inner perimeter shows Trigger Distance

Debug Toggle:



When toggled, the following information will be drawn in the scene for your convenience:

- **AudioSource Location**: A small widget will appear, indicating the current location of the AudioSource.
- **Occlusion Raycast Information**: When Occlusion is enabled, raycast lines will be displayed in two different states:
    - **Unoccluded**: Yellow.
    - **Occluded**: Red.

Preview Buttons:

Lastly, two buttons in the inspector allow you to preview your soundscape:

- **Start Preview**: Previews only the selected Audio Zone.
- **Preview Soundscape**: Previews your entire soundscape.

Both buttons function as toggles, allowing you to easily enable or disable previews at any time in edit mode.

API and Method Information

Sound Shapes' Audio Zone system is designed to work seamlessly, both in-editor and at runtime during play mode.

Below you'll find a list of accessible public methods you can use to suit your specific use case—whether you're procedurally generating Audio Shapes or dynamically changing targets.

AudioZone Method Documentation

🎯 Tracking Methods

- **SetTarget(Transform trackingTarget)**
  Sets the specific target object for tracking.
  Parameters:
  • `trackingTarget` *(Transform)* – The transform of the object to track.
- **SetTrackingMode(TrackingMode newMode)**
  Sets the tracking mode for the audio zone (Tag-based or Object-based tracking).
  Parameters:
  • `newMode` *(TrackingMode)* – The new tracking mode to apply.
- **ToggleShouldTrack(bool shouldTrack)**
  Enables or disables target tracking functionality.
  Parameters:
  • `shouldTrack` *(bool)* – True to enable tracking; false to disable.
- **SetTrackingTag(string newTag)**
  Sets the tag used for identifying the object to track when in tag-based tracking mode.
  Parameters:
  • `newTag` *(string)* – The tag of the GameObject to track.

Shape Management

- **ToggleClosedLoop(bool closedShape)**
  Sets whether the audio zone shape should be closed (forming a loop). Only applicable in Shape mode.
  Parameters:
  • `closedShape` *(bool)* – True if the shape should be closed.
- **AddShapePoint(Vector3 location)**
  Adds a new point to define or extend the audio zone shape.
  Parameters:
  • `location` *(Vector3)* – Position of the new shape point.

- **RemoveShapePoint(int pointIndex)**
  Removes a specific shape point at the given index if the index is within range.
  Parameters:
  • `pointIndex` *(int)* – Index of the shape point to remove.
- **ClearShapePoints()**
  Clears all defined shape points.
- **SetShapePointLocation(int index, Vector3 location)**
  Updates the position of a specific shape point by index.
  Parameters:
  • `index` *(int)* – Index of the shape point to update.
  • `location` *(Vector3)* – New location for the point.
- **PopulateShapePoints(List transforms)**
  Adds multiple shape points based on a provided list of transforms.
  Parameters:
  • `transforms` *(List)* – Transforms whose positions define new shape points.
- **PopulateShapePoints(List positions)**
  Adds multiple shape points from a list of Vector3 positions.
  Parameters:
  • `positions` *(List)* – Positions to use as shape points.

Multi-Emitter Management

- **AddMultiPoint(Vector3 location)**
  Adds a new emitter point at a specified location within the multi-emitter zone.
  Parameters:
  • `location` *(Vector3)* – Position for the new emitter point.
- **RemoveMultiPoint(int pointIndex)**
  Removes a specific emitter point by index.
  Parameters:
  • `pointIndex` *(int)* – Index of the emitter point to remove.
- **ClearMultiPoints()**
  Clears all defined multi-emitter points.
- **SetMultiPointLocation(int index, Vector3 location)**
  Updates the location of a specific multi-emitter point.
  Parameters:
  • `index` *(int)* – Index of the emitter point to update.
  • `location` *(Vector3)* – New position for the emitter point.
- **PopulateMultiPoints(List transforms)**
  Adds multiple emitter points from a provided list of transforms.
  Parameters:
  • `transforms` *(List)* – Transforms whose positions define new emitter points.
- **PopulateMultiPoints(List positions)**
  Adds multiple emitter points from a list of positions.

Parameters:

• `positions` *(List)* – Positions for new emitter points.

Mesh Management

- **AddMeshTarget(MeshFilter filter)**
  Adds a mesh to define the audio zone shape.
  Parameters:
  • `filter` *(MeshFilter)* – MeshFilter component contributing to the zone definition.
- **ClearMeshTargets()**
  Clears all assigned mesh targets and cached mesh data.

Thank You for Choosing Sound Shapes!

I hope this tool makes it more fun and smooth for you to work with sound in your project!

If you have any questions, feedback, or run into any issues, please don't hesitate to reach out at **TelePresentGames@gmail.com**, or **join the Discord!**. Your input is invaluable.

Happy developing!

Kind regards,
Martin

Changelog:

Original Release:

- Original release.