AMBIENT SOUNDS

By Procedural Worlds

Ambient Sounds is a system to create and play back interactive soundscapes in defined locations.

Contents

Welcome!

Thank you for purchasing Ambient Sounds!

Ambient Sounds is a sophisticated tool with a lot of options and while you can go as deep as you like to create very complex soundscapes, you can also start easily and quickly.

To get up and running quickly we strongly recommend that you review and follow the Workflow / Quick start section in this manual. It will guide you through the setup of your first Ambient Sound Area.

And finally, we have also created an awesome support network for you which you can access from the Window > Procedural Worlds menu in Unity.

PRO TIP:

Did you know that we also have a range of other products to enhance your environments in Unity? For example, Gaia can help you to create entire terrains from scratch, or with Gena Pro you can create roads and rivers on your terrain in minutes!

Check out our other products on the next page to learn more!

About Procedural Worlds

Powerful, simple, beautiful. Friendly tools, gorgeous games!

Procedural Worlds empowers artists and developers to bring their vision to life by making it easy to create beautiful worlds. Leverage the latest procedural generation techniques to take the pain out of creating stunning environments and focus on creating amazing games.

NEW – Canopy community site

Canopy is our new community site for all Procedural Worlds Tools with support forums, knowledge base library, tutorials, and much more.
Register with Canopy to receive FREE stamp packs for Gaia and get the best out of Ambient Sounds.

The only end to end environmental generation and delivery suite:

Gaia Pro 2021 - A world generation system for creating, texturing, planting and populating scenes from low poly mobile, VR and through to high end desktop.

GeNa Pro - A sophisticated localised level design tool that augments Gaia's broad-brush strokes, by working intuitively to give fine grained control.

SECTR - A suite of performance-enhancing tools that enable open world streaming, massive mobile games and includes the latest techniques in audio occlusion and propagation.

Ambient Sounds - Lets you configure music and sounds to create a unique atmosphere for each region in your game, which can react to changes in your gameplay instantly.

Pegasus - A cut scene and fly through creator that makes it easy to show off gorgeous environments and drive characters through scenes with localised avoidance and Mecanim animation support.

Spawner Packs – You can save time by using our pre-configured Procedural Worlds Spawner packs (PWS). The packs contain configurations for our tools Gaia and GeNa, and are designed to work with popular asset packs from the Unity Asset Store. Currently available:

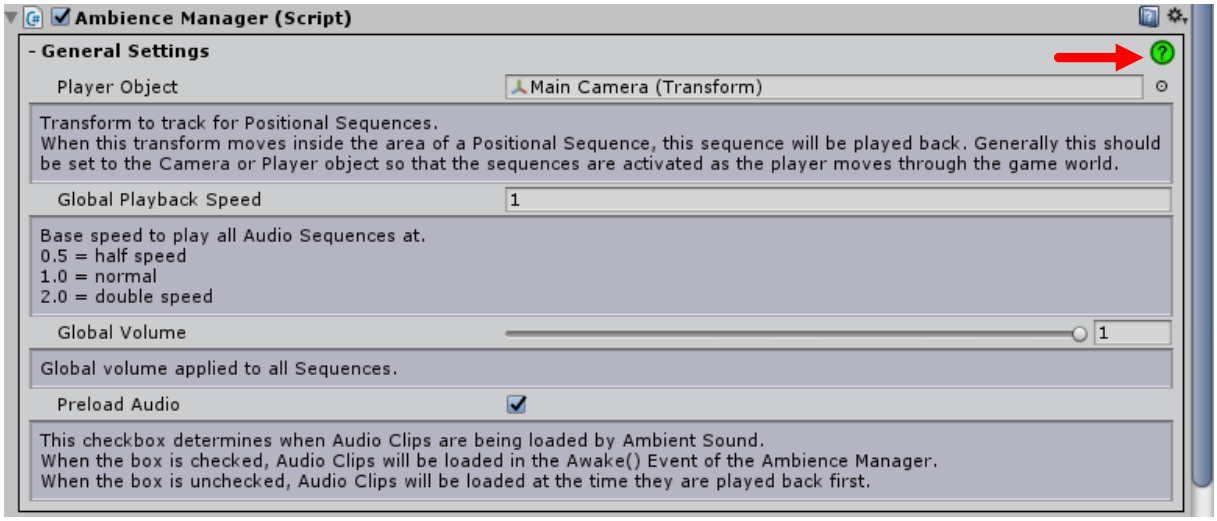
PWS – POLYGON Fantasy Kingdom - Spawner Pack
PWS – POLYGON Nature - Spawner Pack
PWS – SUNTAIL Stylized Fantasy Village – Spawner Pack

Micro Biomes - Let us inspire you with our new Micro Biome series where we put together groups of matching assets that cover one specific aspect of environmental design and release them in targeted high quality packs:

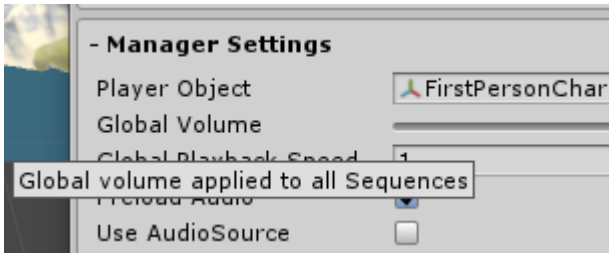
Micro Biomes – Deep Desert
Micro Biomes – Fields of Color

Tutorials, Chat, Support

In general Ambient Sounds is self-documenting – to understand a control click on the small help button in the control and help messages will appear.



Most input fields also offer a help text if you move the mouse over the name of the field:



For a Tutorial you can follow along with the Work Flow / Quick start Guide of this manual, or check the Ambient Sounds tutorial section in the Canopy Library.

Still stuck even after reading the quickstart guide? Please follow the support workflow as described here: https://www.procedural-worlds.com/support/

Subscribe to our newsletter on our website: https://www.procedural-worlds.com/

Installation

Installing Ambient Sound will create this folder structure with the following contents:

Procedural Worlds – root folder for all Procedural Worlds Assets

- **Ambient Sounds** – root folder for the Ambient Sounds Asset
    - **Audio** – Example Audio files and preconfigured Ambient Sounds assets. If you want to use your own Audio files exclusively, you can safely delete this folder.
    - **Demo** – demo scene that will make you familiar with the core concepts of Ambient Sounds. You can safely delete this folder if you don't need the demo scene and its contents in your project.
    - **Documentation** – Ambient Sounds documentation
    - **Localization** – Localized UI texts
    - **Scripts** – The core logic of Ambient Sounds
- **Framework** – shared functionality between Procedural Worlds assets

What is Ambient Sounds?

Ambient Sounds is a system to create and play back interactive soundscapes in defined locations.

With Ambient Sounds you will:

- Organize your audio clips into Sequences with combined playback settings
- Playback these Sequences either globally ("everywhere") or locally in a certain area of your game world
- Pair multiple Sequences of background noises and music to create an unique soundscape for each location in your game
- Make these Sequences interactive by controlling them through Requirements
- Modify these Sequences on the fly to react to Events in your game
- Develop Sequences quickly by stopping and starting Sequence playback during runtime at will while testing different playback parameters
- Synchronize Playback between Sequences to make them start and end at the same time even if they are different in duration.
- Keep control over your Ambient Audio setup with comprehensive lists of all Sequences and conditions in your scene in the Ambient Sounds Manager Window.

Ambient Sounds Concepts & Components

Ambient Sounds allows you to combine sounds into Sequences that will play those sounds in random or non-random order. Multiple Sequences can be played back at the same time at a defined location to create a unique and interactive soundscape for that location.

The main Ambient Sounds components are:

Ambience Manager – the main playback component that needs to be present somewhere in your scene for Ambient Sounds to function. It will track the player / camera for you and evaluate which Sequences need to be played and which modifiers are currently applied to them. Select this component during runtime to get an overview which Sequences are currently playing and which Sliders and Events are currently active.

Sequence – A Sequence is a collection of audio clips combined with rules and settings for playback. A Sequence can have certain requirements before it will play, and can have multiple Modifiers attached to it.

Modifier – A modifier will modify the playback parameters of a Sequence it is attached to when its requirements are met.

Requirements: Values and Events – Both Sequences and modifiers can have requirements that need to be fulfilled before they become active. Requirements for Sequence and modifier activation can be defined in two ways:

- Values – Allow you to set up a requirement based on a range of numerical values, e.g. "Only be active when the variable 'Hours' is between 0.4 and 0.6.

- Events – Allow you to set up a requirement based on a named Event, that can either be active or not, e.g. "Only be active if the Event 'Combat' is currently active"

It is also possible to have a combination of multiple Values and Events as requirements. Each Sequence and each modifier on it can have its own requirements, which allows you to set up complex conditions for your soundscapes.

Audio Areas – Multiple Sequences can be combined in an Audio Area with a defined shape. Audio Areas will play back the associated Sequences if:

- The requirements for the Sequence are met
- The player / camera has entered the associated area

The areas can be one, two or three dimensional and have different shapes as well.

Sync Groups – Allow you to synchronize two or more Sequences together by adjusting their playback speed or repeating shorter Sequences while the longer Sequence is still playing.

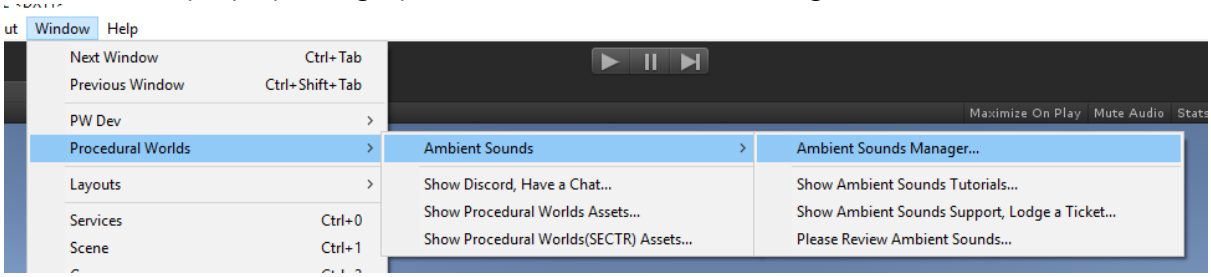
Work Flow / Quick start Guide

This chapter gives an overview about how to set up Ambient Sounds playback for a project. It is recommended to follow along with these instructions to get started quickly with using Ambient Sounds.

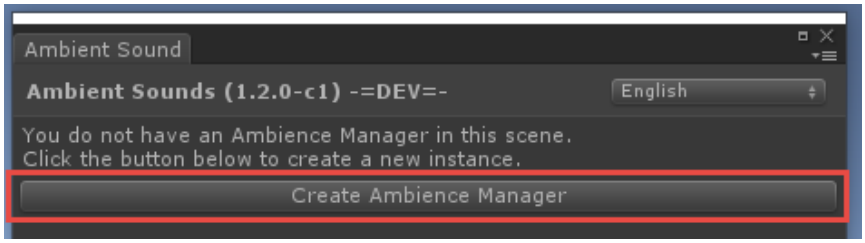
Setting up the Ambience Manager

The Ambience Manager is the core component responsible for Ambient Sound Audio Playback. You can attach it anywhere in the scene, but during playback it will parent itself automatically to the tracked player object.

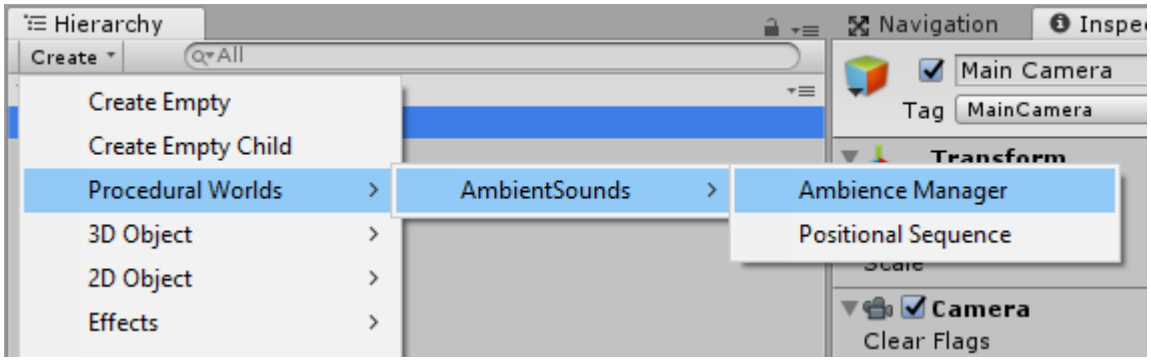
You can let Ambient Sounds attach an Ambience Manager directly to your player automatically by opening up the Ambient Sounds Manager:



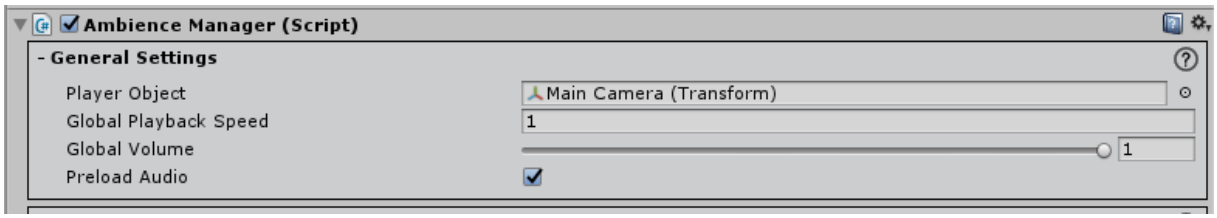
and then pressing the Create Ambience Manager button:



OR you can create a container for it in the scene hierarchy:



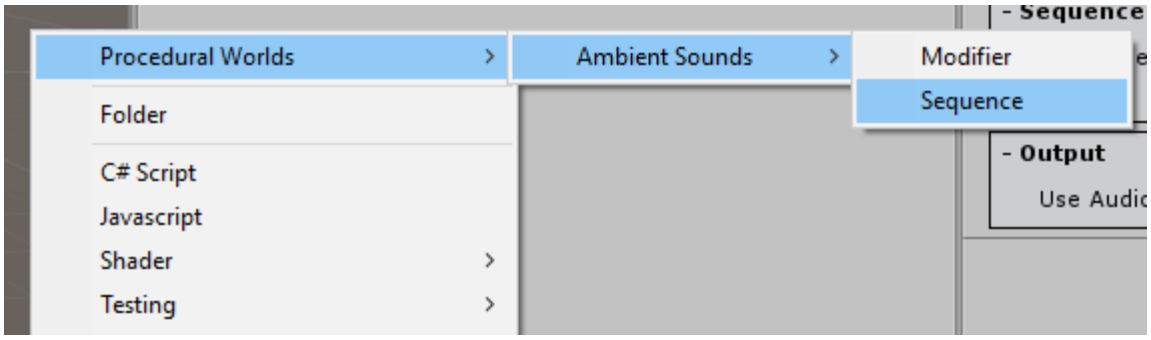
In the Ambience Manager you need to set up a player object. This is the transform that is tracked to evaluate positional audio Sequences. Usually this should be the transform where your audio listener is attached to; most likely this is the player or the camera.



This is all initial setup the Ambience Manager requires.

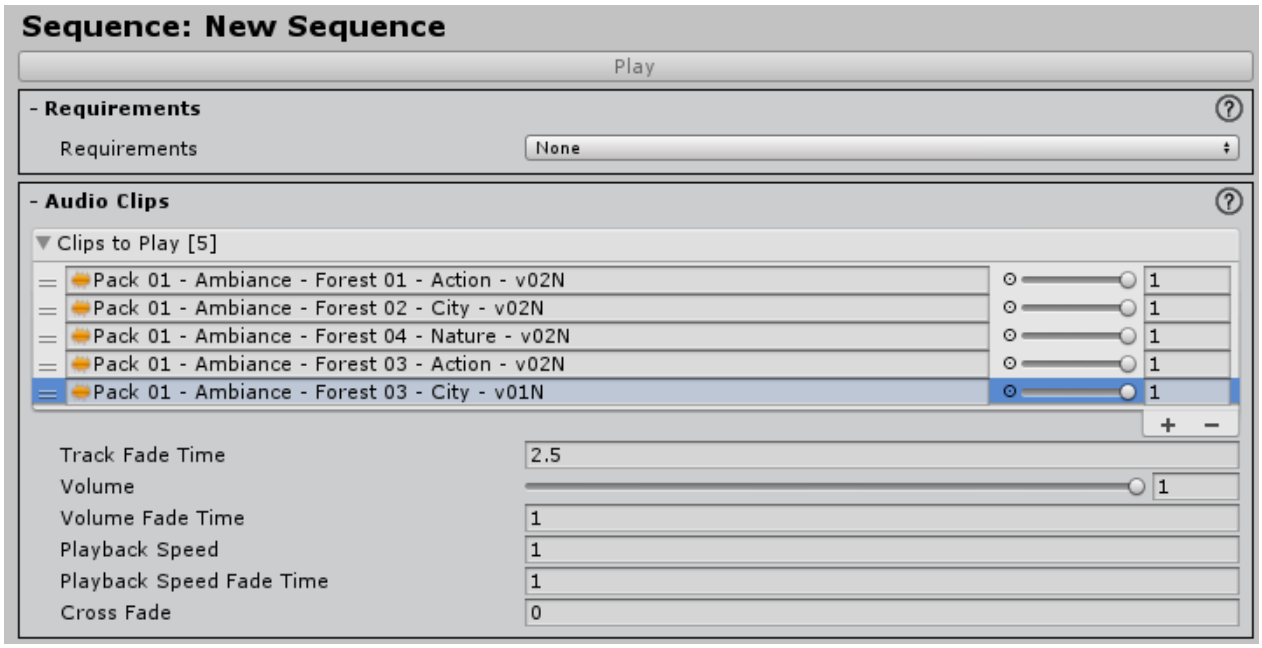
Setting up a Global Sequence

We need to define a Sequence as well so we have something that the Ambience Manager can play back. Create a new Sequence in your Asset Hierarchy:

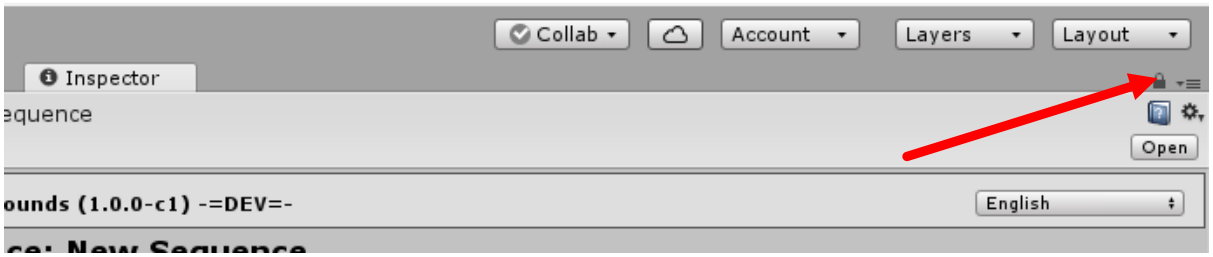


Select the newly created Sequence and add some Audio Clips to it. You can take content from the Assets\Procedural Worlds\Ambient Sounds\Demo\Audio\Music folder if you don't have any Audio Clips in your project yet.

Just click the plus button at the bottom of the list to increase the number of clips and drag and drop some clips in the list. Your result should look something like this:

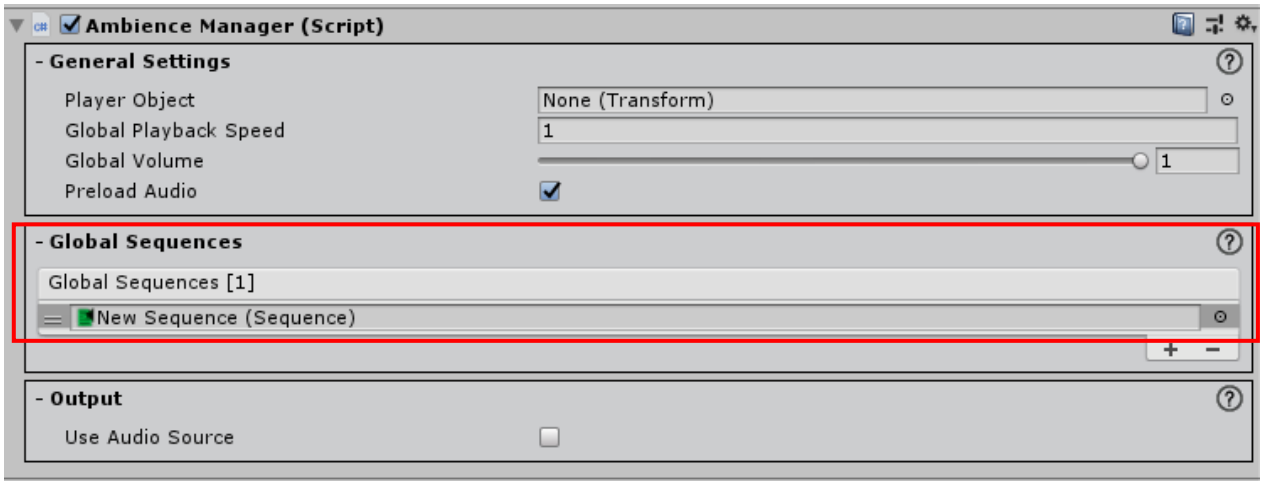


Tip: It can be difficult to hunt for clips in the asset hierarchy because your inspector window will lose focus on the Sequence every time you select a clip in the asset hierarchy. You can prevent this by locking the inspector window down with the little lock icon in the top right corner:

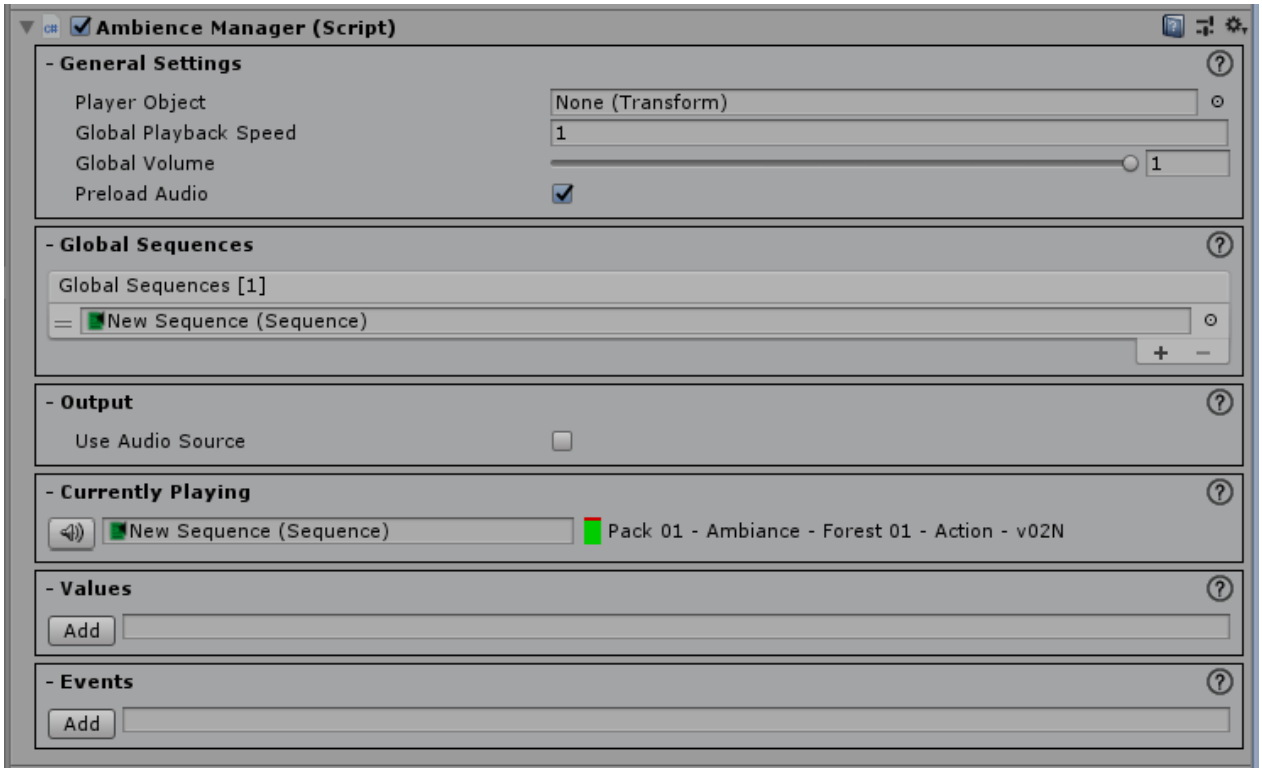


In this way the inspector window will stay locked on your Sequence until you unlock it again.

The easiest way to get this Sequence played back is to set this scene up as a Global Sequence in the Ambience Manager. Select the Ambience Manager you created before and add your new Sequence as a Global Sequence:



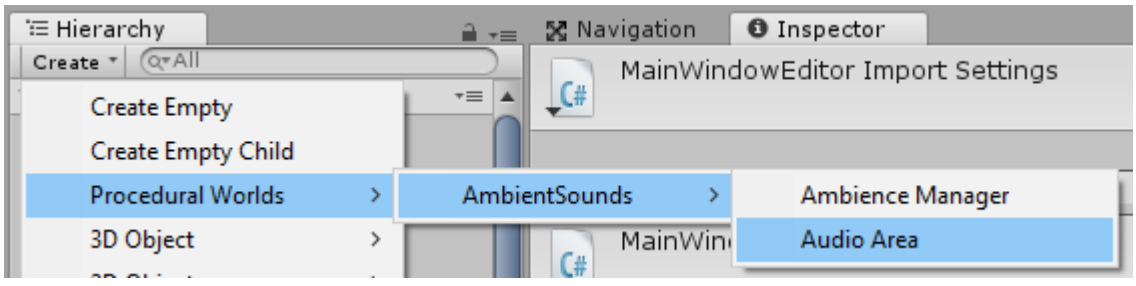
If you enter play mode, the audio clips will be played back immediately since Global Sequences are played back regardless of the player position in the scene. If you select the player / camera object while running your scene, you can see that the Ambience Manager is playing back your Sequence in the active Sequences list.



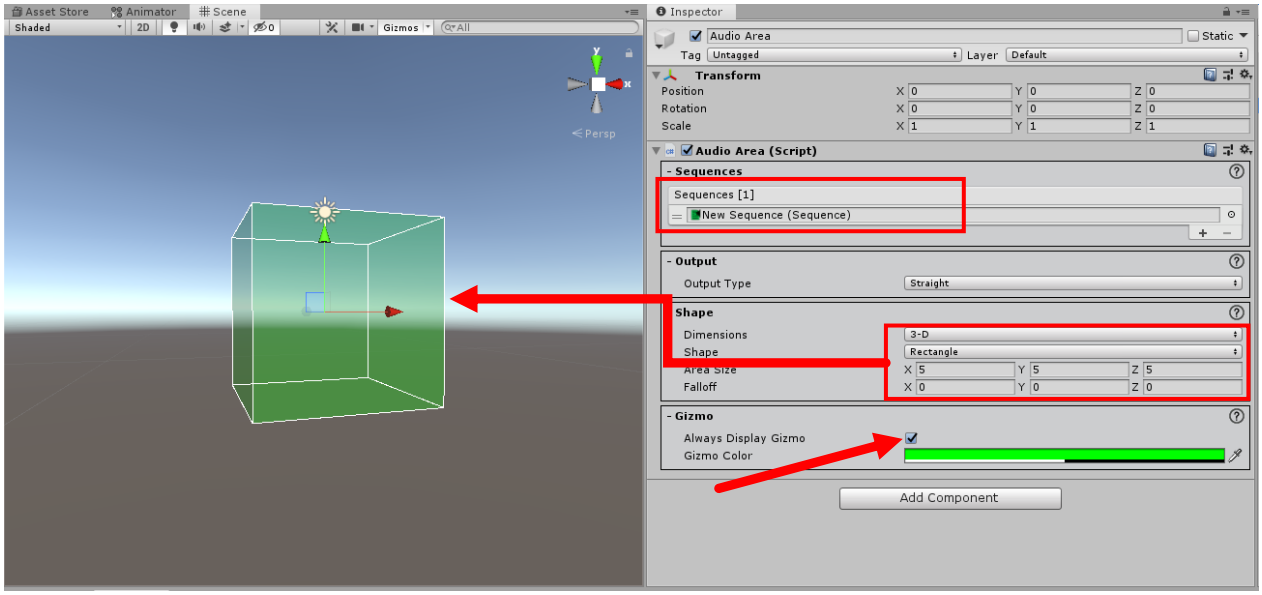
Note that you can double click on the Sequence in "Currently Playing" to select it quickly in the Asset hierarchy. The Speaker Button allows you to mute the playing track which is useful for debugging during runtime. The red line in the Bar Graph on the right side represents the "fade state" of this sequence and the green part represents the volume setting in the sequence for the current clip.

Restricting Playback to an Area: Setting up an Audio Area

Now that we know how we can define a Sequence and play it back globally, let's try to play back the Sequence in a certain area of the game world only. Remove the Sequence from the Global Sequences again to set up an Audio Area instead. Create a new Audio Area from the Scene Hierarchy:

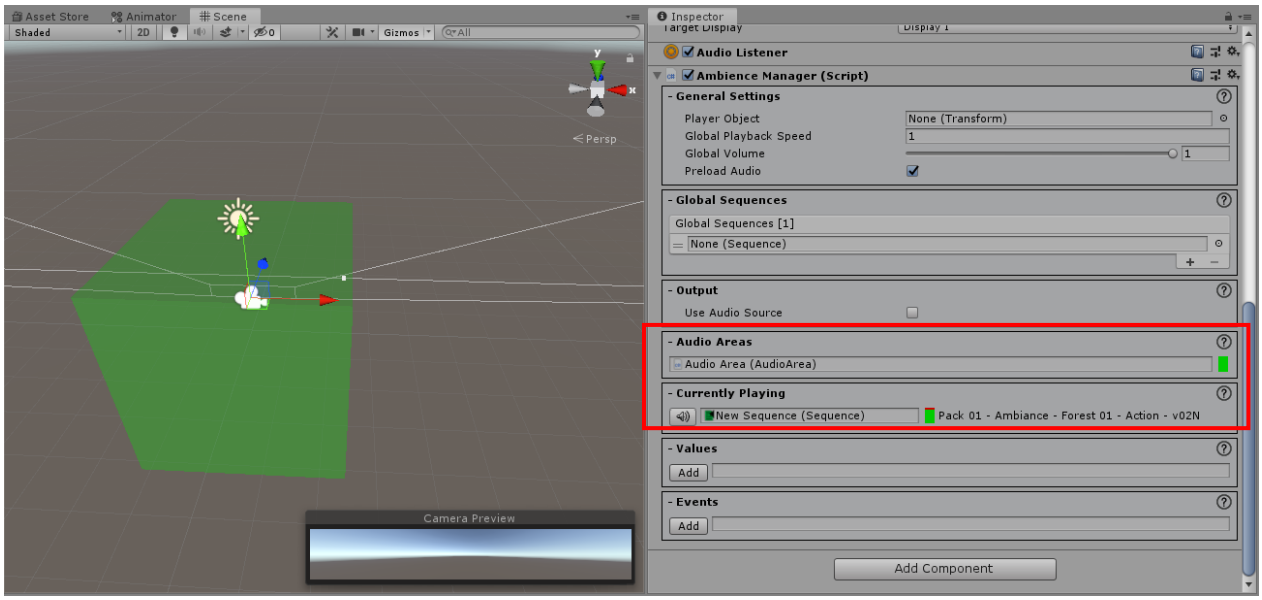


Add the Sequence you created before to the list of Sequences for this area. Then set up a shape for this positional Sequence. For your first test a 3D box should do fine:



Select "Always Display Gizmo" at the bottom as well so you can still see the Shape when entering play mode.

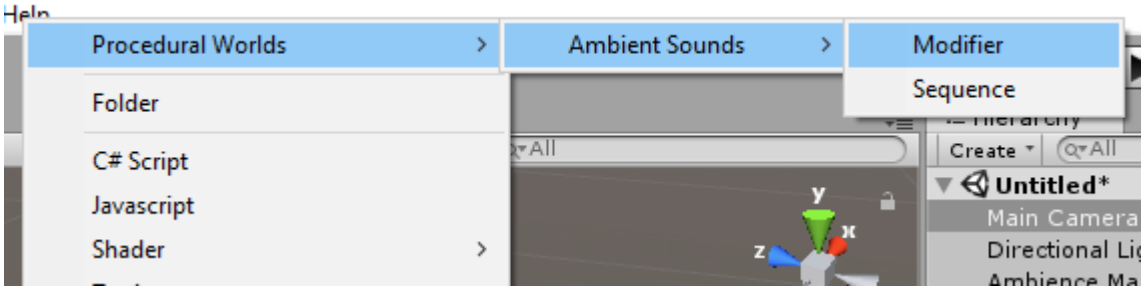
If you start play mode and enter the shape that you defined before with your player / camera, the audio clips from the test Sequence start playing. If you select the player / camera object you can see that the Ambience Manager is updated with current information during runtime again. It displays the currently active Audio Areas and the Sequences playing therein:



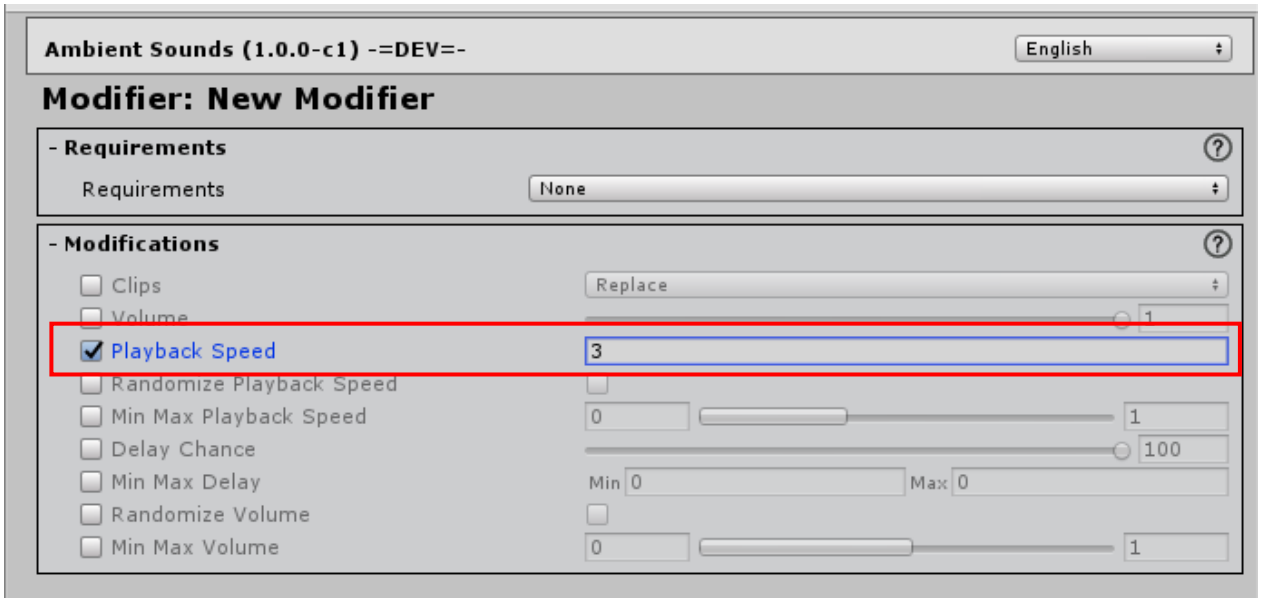
Note that you can double click both on the Audio Area as on the Sequence to get the corresponding object selected in the scene / asset hierarchy.

Using a Modifier

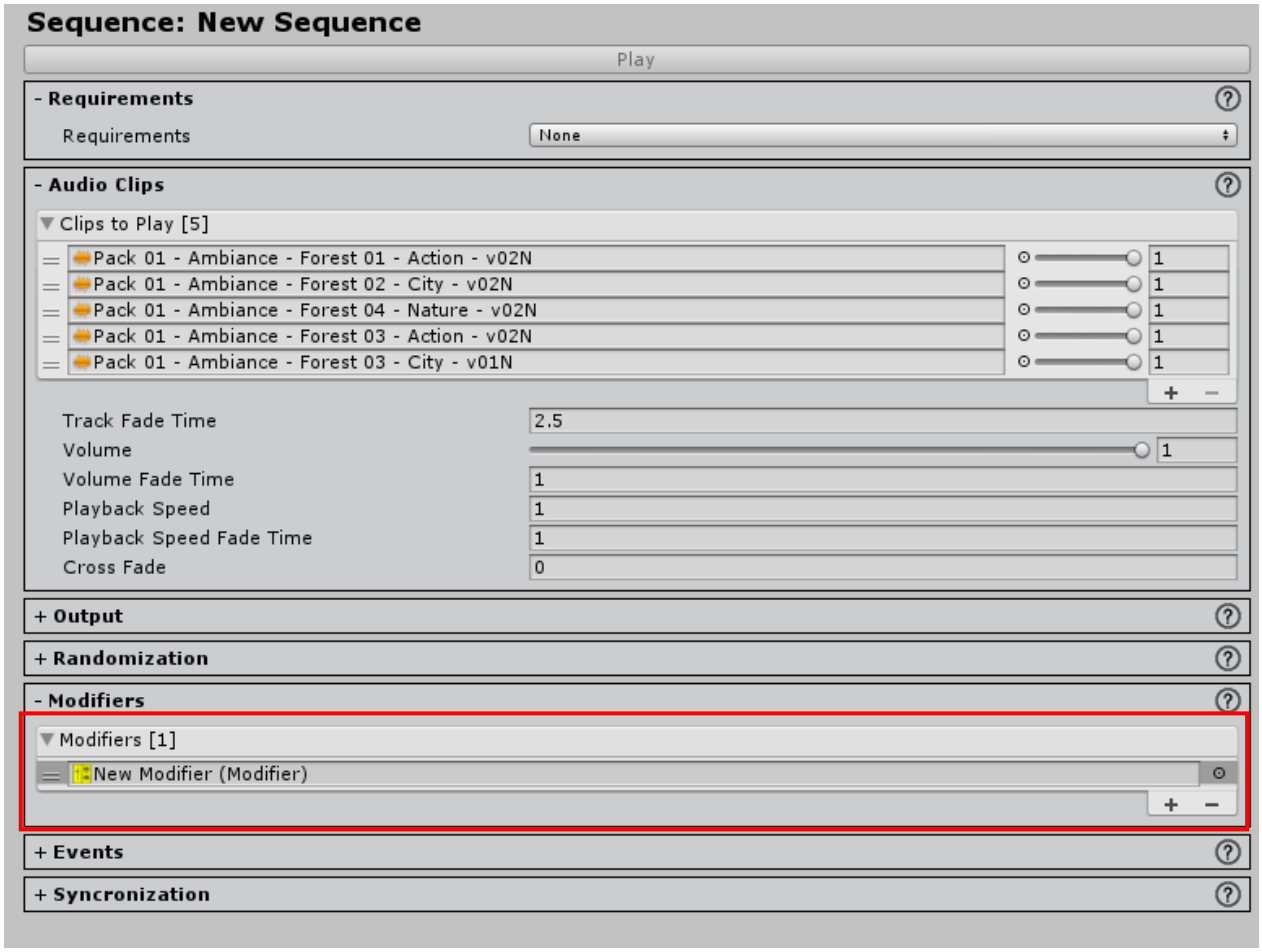
So far we have set up a couple of clips to play back at a certain location in the scene. Now let's try to add a modifier to change the playback settings of the scene. Stop your game and create a Modifier from the Asset Hierarchy Create Menu:



In the modifier set up a noticeable modification for our test Sequence. For example you can increase the playback speed to 3:



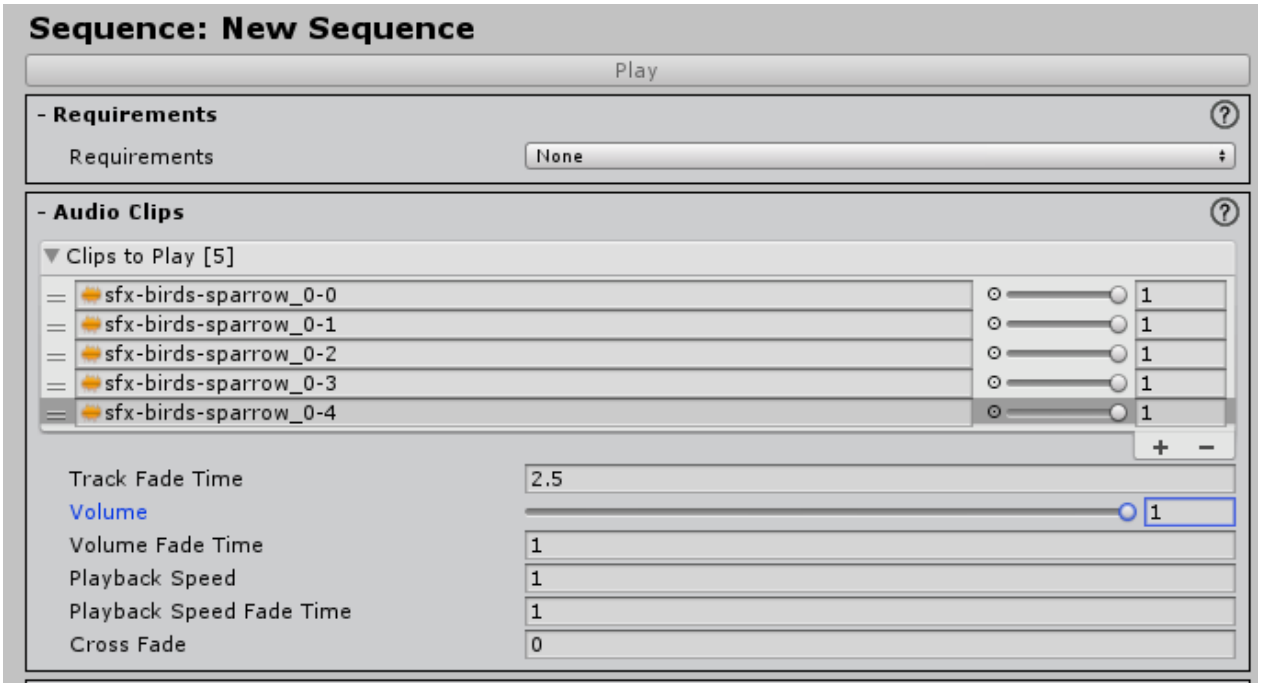
Now add the modifier to the Sequence you created earlier:



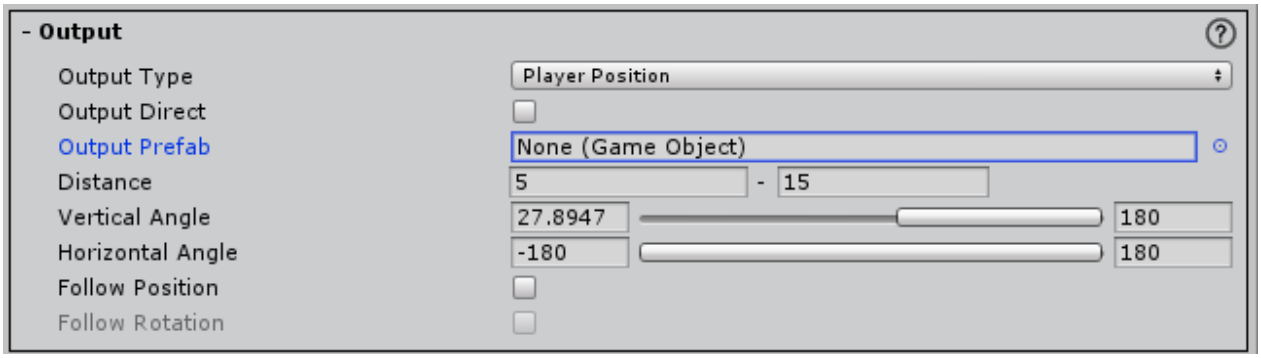
If you enter play mode now, you should notice that the audio playback speed in the positional Sequence is now much faster.

Playing random SFX around the player

If you followed this guide so far, all the output has been played back "straight" by Ambient Sounds, which means it was fed directly into the Audio Listener. While this is great for music and ambient background loops, you also want to be able to play back ambient sounds that play back with spatial positioning around the player. You can try this now by creating another Sequence with random noises. Create another Sequence in the Asset Hierarchy just as before from the "Create" menu in the Asset Hierarchy Window or from the Ambient Sounds Manager Window.
Add some random noises as Audio Clips. For this example you can use the bird noises that come with Ambient Sounds:



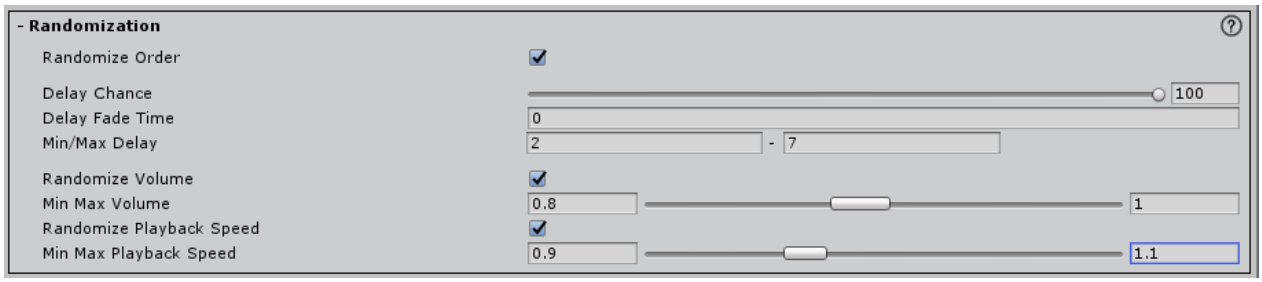
To get these Sounds played back in a random position around the player, we need to change the output type to "Player Position":



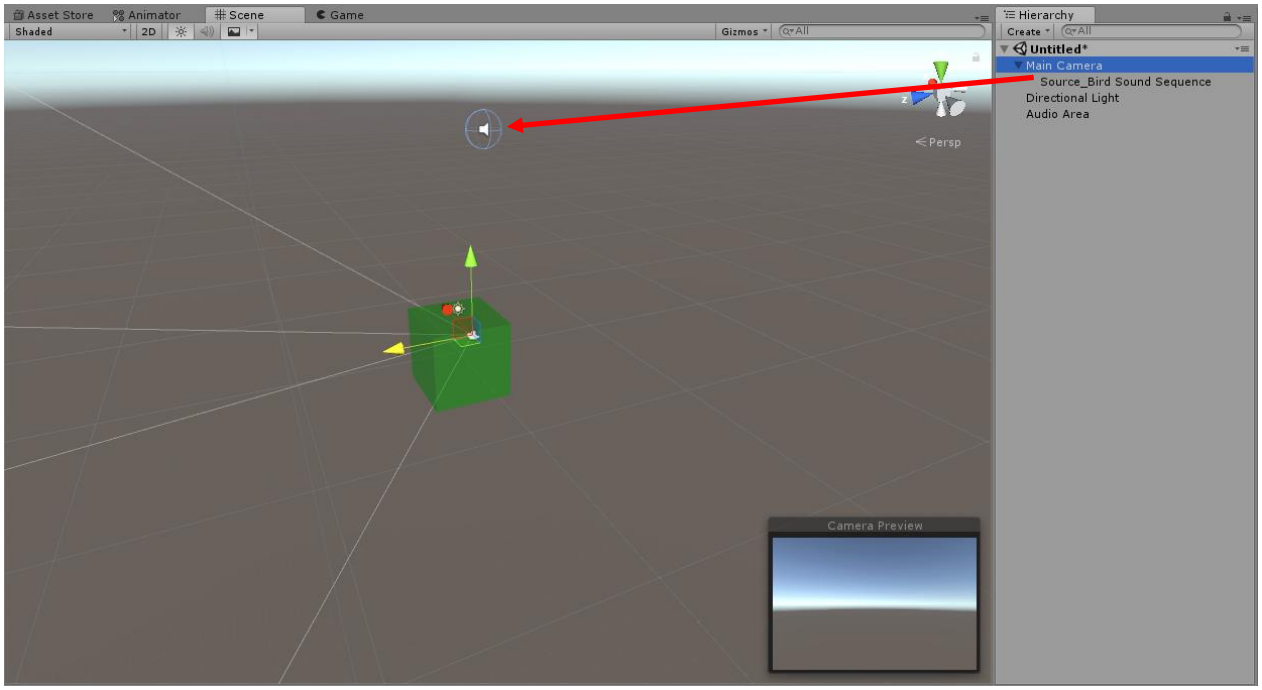
By setting Distance, Vertical & Horizontal Angle you can control in which range / which angle the audio clips should be played back around the player. The position will be determined anew for each clip that is played back. For the random bird sounds it makes sense to keep "Follow Position" and "Follow Rotation" unchecked,

we want the bird noises to be perceived as sounds that exist at their own location in the game world and don't follow the player's movements.

Of course you want the bird sounds to be random as well, so let's set up a random playback order, delay, volume and speed:

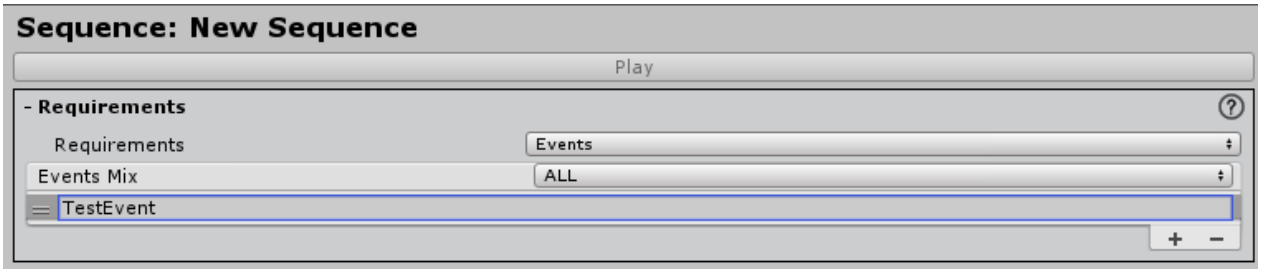


Add this new Sequence to the Audio Area you created before for playback. You should now notice that an Audio Source is created around the player that plays back the random bird noises, and that changes position within the parameters you set up before.

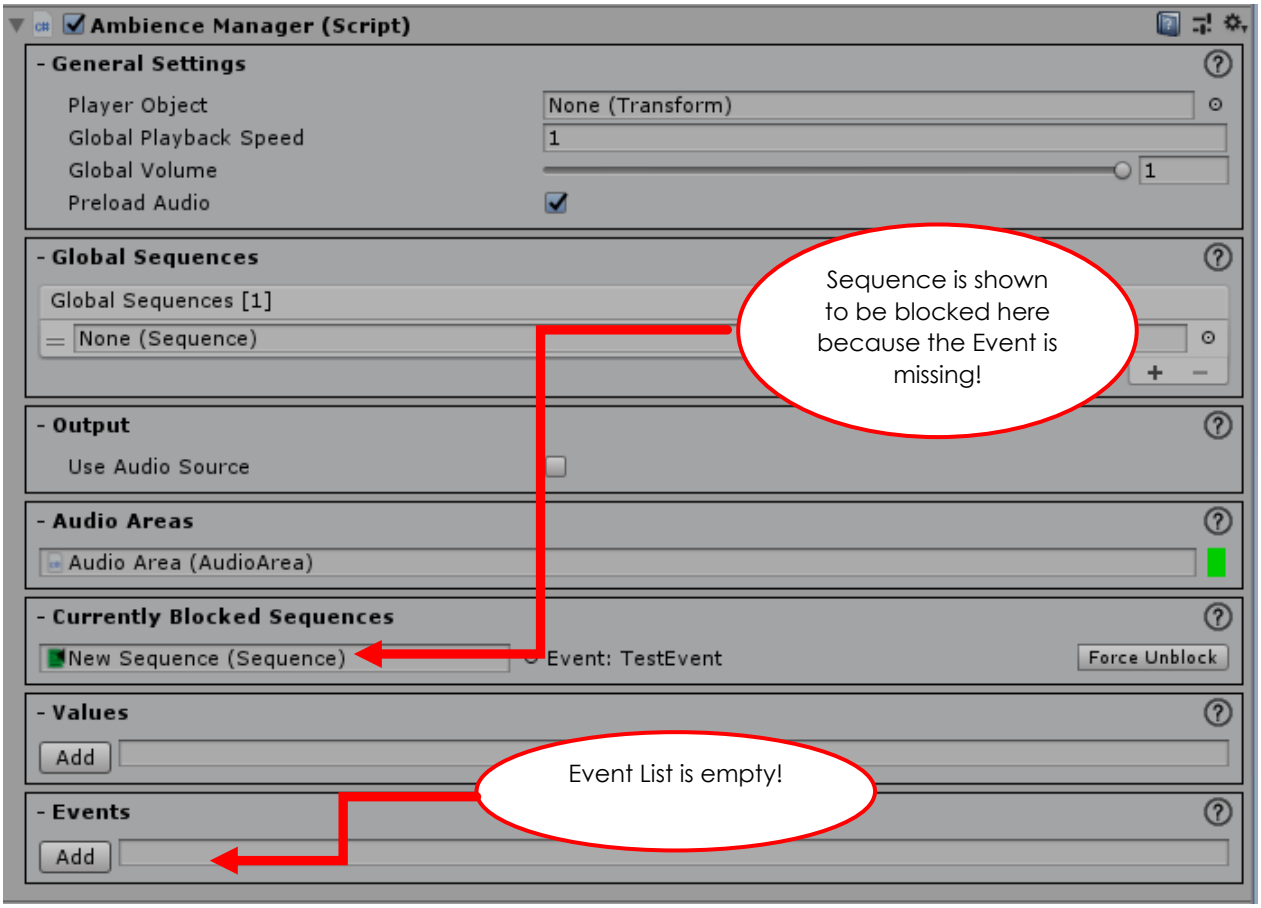


Setting up Requirements: Using Events

So far everything you set up has only been restricted by the player's position and the Audio Area. However often you want Sequences and modifiers to only become active when certain conditions are met. This is where Requirements come into play: Both Sequences and Modifiers can have Requirements that need to be fulfilled before they become active. Let's try an Event first: Select the Music Sequence you created earlier and add an Event named "TestEvent" to the Requirements (Click on the "plus" button at the bottom of the list for a new entry):



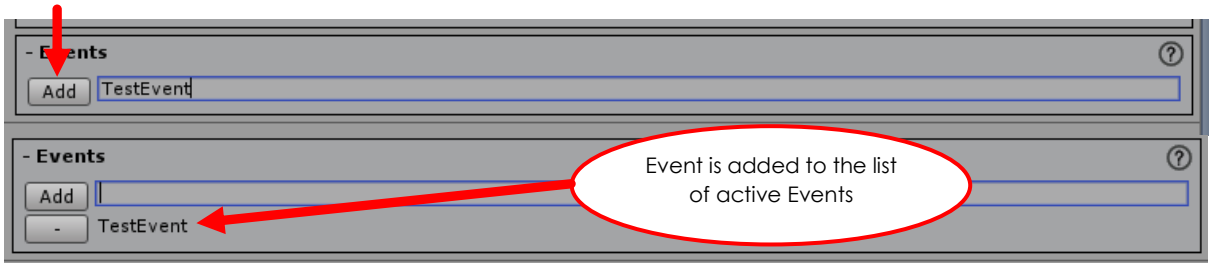
Start Play Mode again in the Unity Editor. There should be no sound being played back at the moment, because the Event in the Requirements is currently not active. You can check for active Events in the Ambience Manager:



Note that there is a section in the Ambience Manager that tells you that there is a Sequence that is not being played at the moment (and the reason for it) under "Currently Blocked Sequences". This is very useful info for debugging sequences. There is also a button to "Force Unblock" which will play the sequence anyways for testing purposes.

Normally you would add an Event from your gameplay logic to control the Ambient Sound playback according to what is going on in your scene, e.g. if enemies appear on the screen, you would create a "Combat" event in Ambient Sounds.

For testing purposes you can add the "TestEvent" manually now to the Ambience Manager by typing it in the text entry field and clicking the "Add" button:



The music should now start to play, if you deactivate the Event again by pressing the small "-" Button, playback should stop again.

Exercise: Stop playback and remove the Event Requirement from the Sequence, but add it to the Modifier you created earlier instead. When you run the scene now, you should be able to speed up / slow down the playback speed by adding and removing the TestEvent in the Ambience Manager.

As mentioned before, normally you would add and remove the Event from the gameplay logic in your scene. To do so, you only need a line of code each:

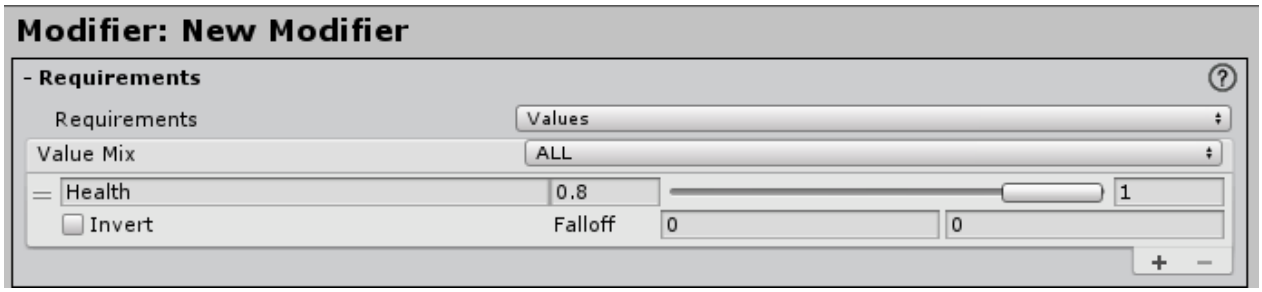
```
AmbientSounds.AmbienceManager.ActivateEvent("TestEvent");

AmbientSounds.AmbienceManager.DeactivateEvent("TestEvent");
```

Note that you can add multiple Events as Requirements as well.

Setting up Requirements: Using Values

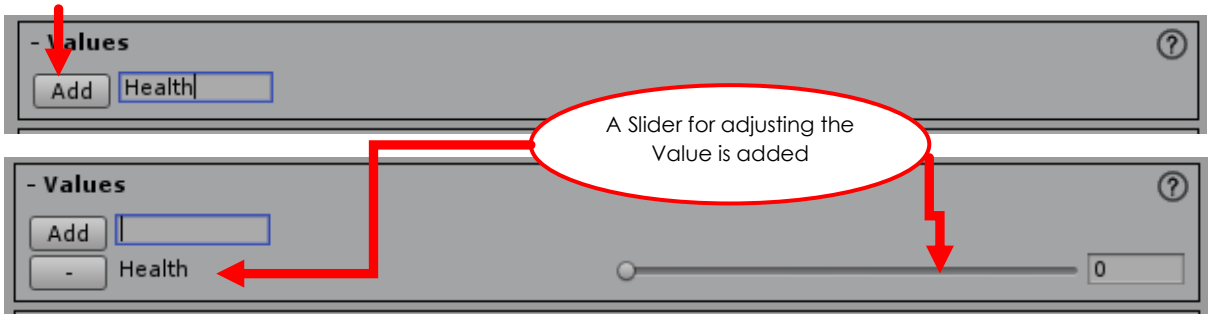
Another way to create a Requirement is Value Checks: A Value Requirement checks if a numerical value falls into a certain range to determine whether the Sequence / Modifier is active or not.

Let's assume you wanted to use the Modifier you created earlier to speed up the music when the player has full health. To do so, select the Modifier in the asset hierarchy and set up a Value Requirement. (Remove any other Requirements you may have set up before first). Final result should look like this:



A Value range is always normalized between 0 and 1, where 0 represents the minimum and 1 the maximum value of the Value variable. The setting as pictured above represents everything above 80% of health.
This does not mean that you need to represent all values in your game with a range between 0 and 1 as well, but you would need to normalize any other numerical range before using it in a Value Requirement Check. This is very easy thing to do; we will look at this in a bit.

For now enter play mode again. Select the Ambience Manager, and just as before with the Events you can add a Value on the fly for testing purposes:



You can play with the Value slider now, if you push it up towards 1 the Modifier should become active and the playback speed should increase, if you move the slider back down again the music will become normal again.

Exercise: Stop playback and remove the Value Requirement from the Modifier, but add it to the Sequence instead. When you run the scene now, you should be able to start and stop the audio playback altogether by adding and playing with the "Health" Value slider in the Ambience Manager.

As mentioned before, normally you would set the Values from the gameplay logic in your scene. To do so, you only need one line of code each time you want to update the Value:

```
AmbientSounds.AmbienceManager.SetValue(“Health”,0.5f);
```

It was also mentioned earlier that you don’t need to manage your numeric values in a range between 0 and 1 as well in your game to be able to use them to check against the Ambient Sounds Requirements.

Let’s assume the minimum health value in your game is 5, the maximum health is 238, and the player’s current health is 123. Now you want to check this current Health value against an Ambient Sounds Requirement. You can “translate” (or normalize) your value system to match it against the 0 to 1 Value Requirement in Ambient Sounds:

```
float normalizedValueForCheck = Mathf.InverseLerp(5, 238, 123);
AmbientSounds.AmbienceManager.SetValue(“Health”, normalizedValueForCheck);
```

This will interpolate the value 123 that lies between 5 and 238 into the corresponding 0 and 1 value of Ambient Sounds.

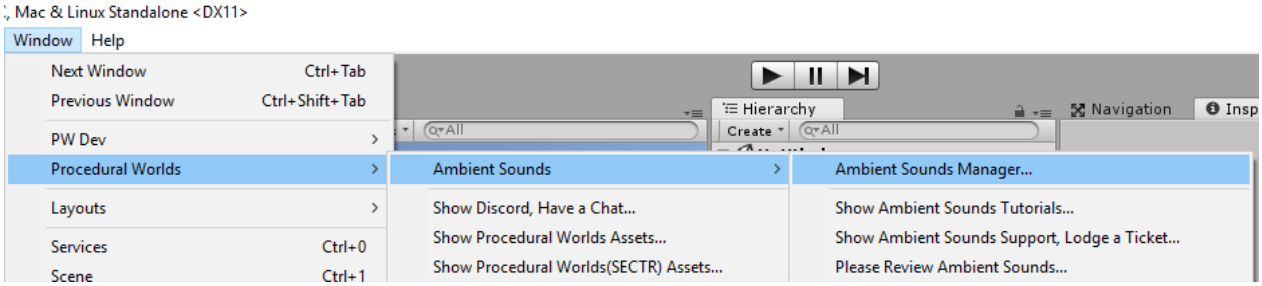
Note that you can add multiple Values for Requirement checks as well. It is also possible to combine Values and Events to create complex conditions to control your audio playback, especially if you can set up the Requirements for each Sequence and Modifier individually.

This concludes the Quick start / Workflow guide, if you followed along this far you should be in a good position to start configuring Ambient Sounds for use in your own project now. If you are stuck somewhere remember that you can enable help anywhere in the UI by clicking on the help button in the corresponding panel.

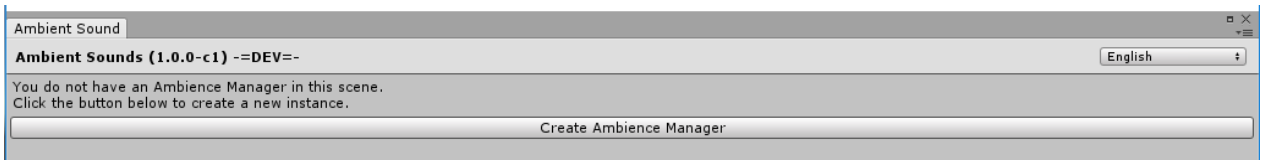
If you find yourself losing overview over the various Sequences and modifiers you created, the next chapter is for you as well: The Ambient Sounds Manager Window will help you to keep an overview for all the Ambient Sounds assets you created in your scene.

The Ambient Sound Manager Window

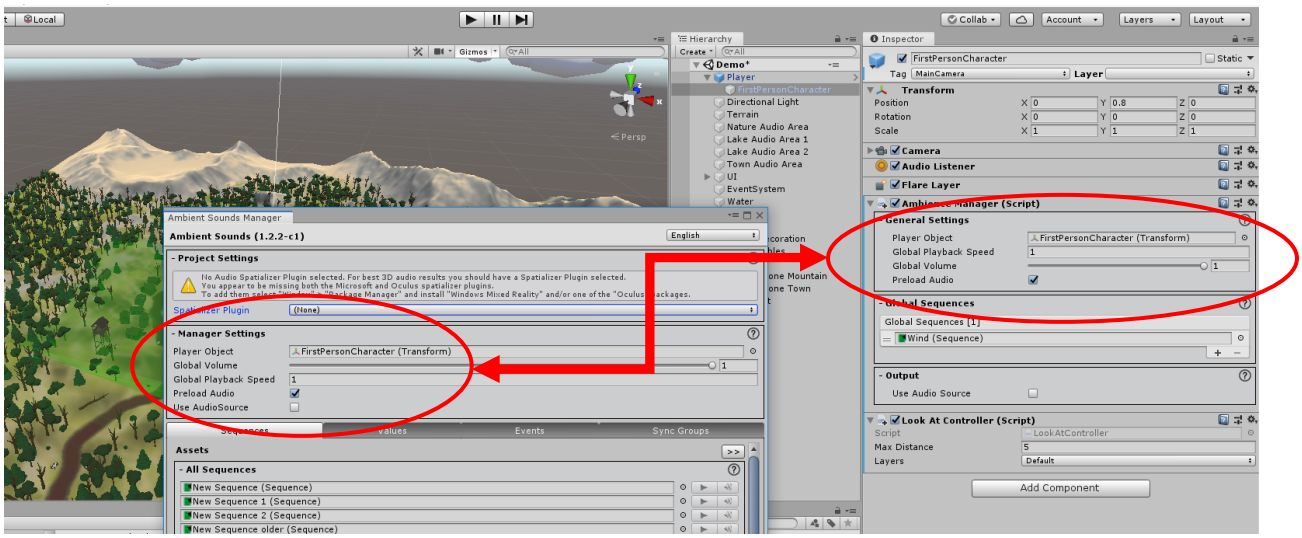
The Ambient Sound Manager Window gives an overview of all active Ambient Sounds Elements in a scene. You can open the Ambient Sounds Manager Window from the Unity Window menu:



In case you don't have any Ambience Manager in your scene you will be offered to have one created for you:



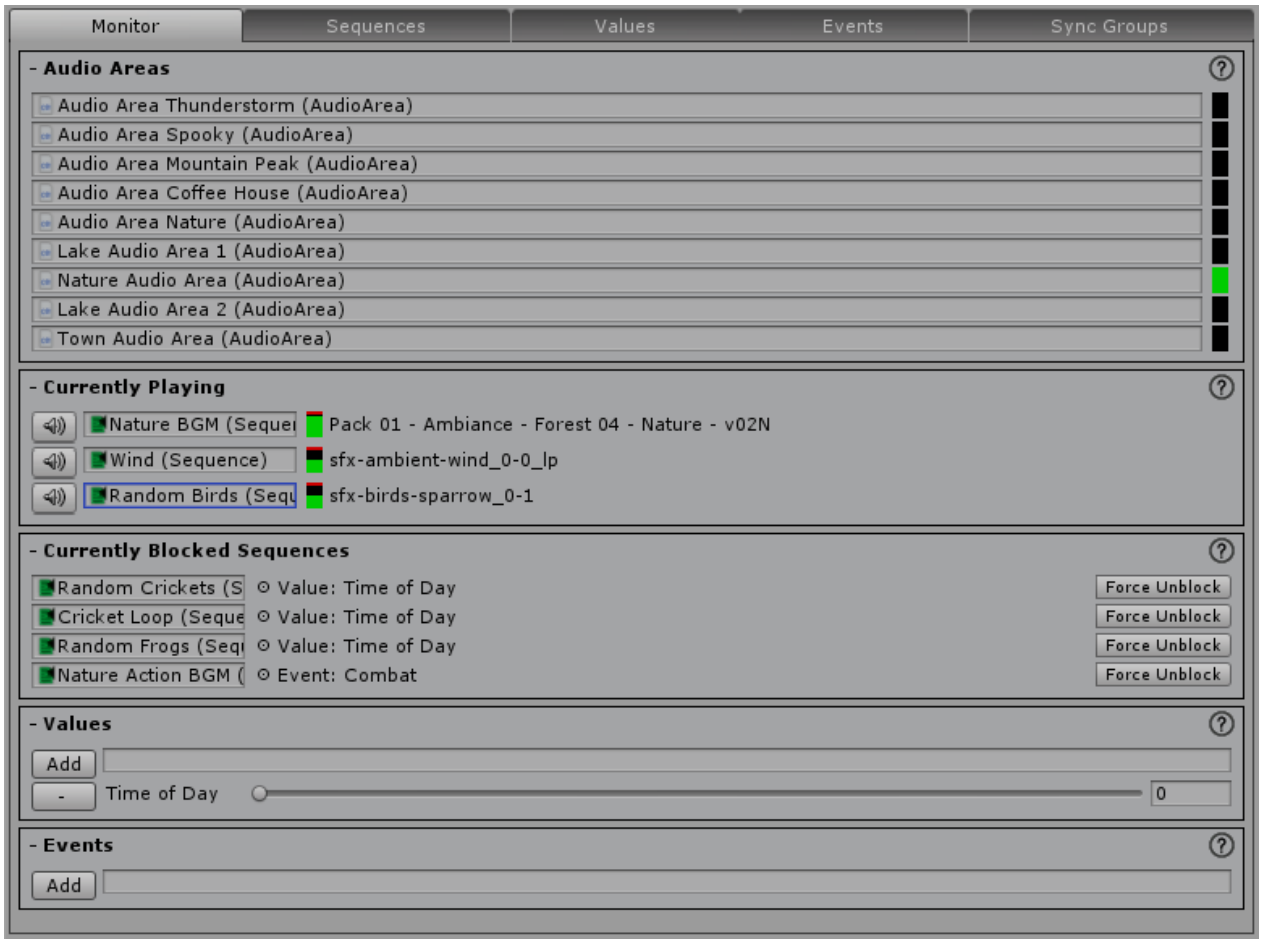
Once an Ambience Manager is present in the scene, its settings will be mirrored on top of the Ambient Sounds Manager Window for easy access:



The lower half of the Ambient Sounds Manager Window offers multiple Tabs with different functionalities:

Monitor Tab

The Monitor Tab only appears during runtime and gives you an overview over the playback state for Audio Areas and currently playing Sequences.



This tab offers the same functionality as the runtime information from the Ambience Manager so you can quickly cross-reference the current state of audio playback in the Manager Window.

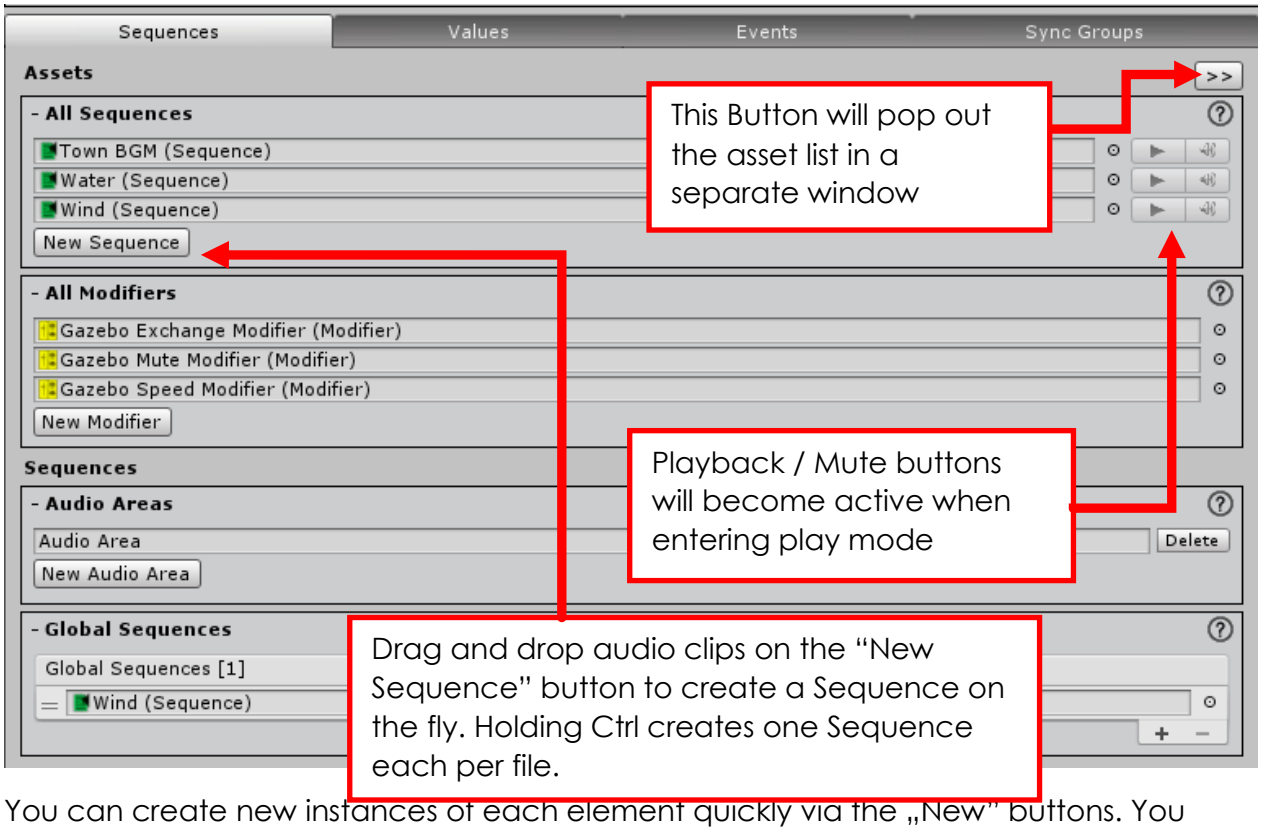
Double click in the listed elements to select them in your inspector.

If you want to know more about the single list items you can find more information in the Ambience Manager detail chapter.

NOTE: The Blocked Sequences section will show why a given sequence is not playing. The sequence will be unblocked when the relevant event or value has been provided.

Sequences Tab

In the Sequences Tab shows all Sequences, Modifiers, Audio Areas and Global Sequences in one comprehensive list.



You can create new instances of each element quickly via the „New" buttons. You can add audio clips to a sequence by dragging them from your asset hierarchy on the sequence entry in this list.

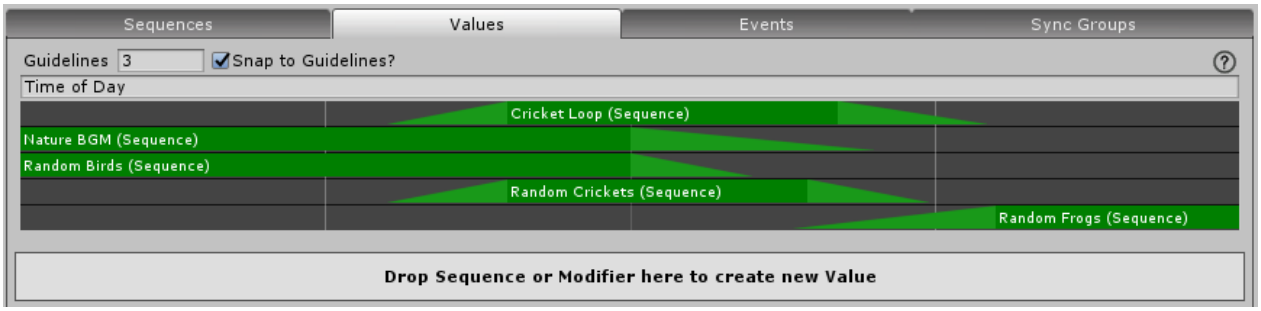
You can also drag and drop audio clips on the "New Sequence" button to create a Sequence on the fly. Holding Ctrl creates one Sequence each per file.

By double clicking any of the list entries the corresponding instance will be selected in the Unity Inspector for editing.

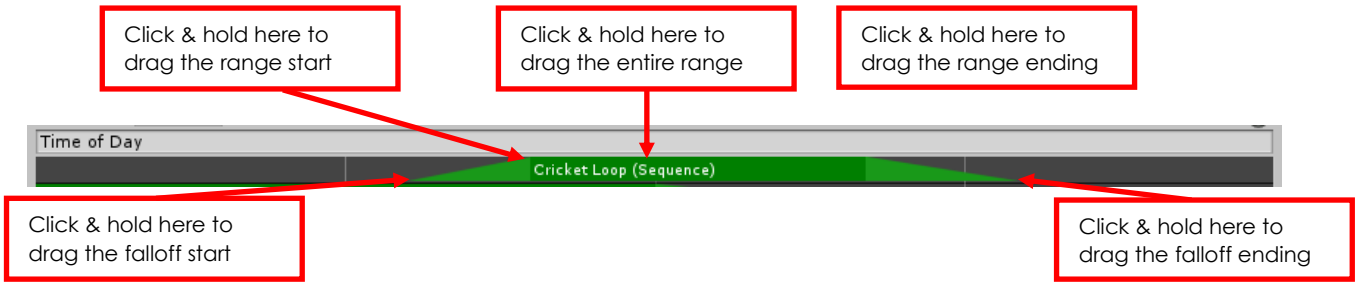
During runtime there is a quickplay button available for each sequence and a mute button to mute the playback for this sequence. Right-clicking the mute button will result in only the selected sequence being played back and all others being muted.

Values Tab

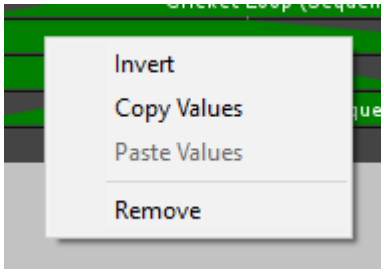
The Values tab displays all Value Requirement Checks that are used in your scene and the Sequences & Modifiers that make use of them. You can directly edit all Value Requirements from this list as well.



To edit a Value Requirement from this listing, you can hold down the mouse button on the center to shift the entire value range, or you can drag the start and end points to adjust the range itself.



You can also right-click on each Sequence or Modifier with the mouse to bring up a context menu with the following functions:



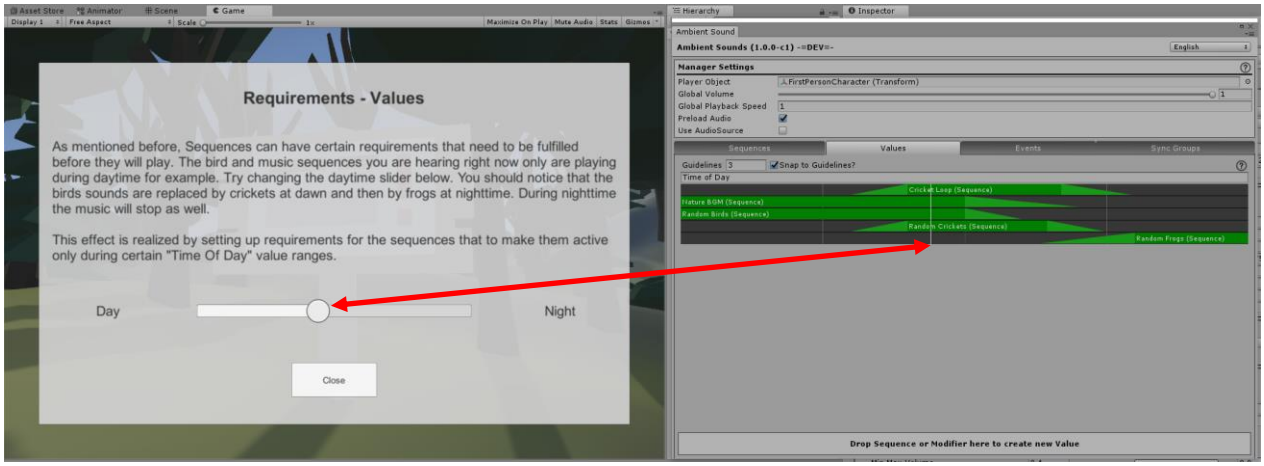
Invert – Inverts the Value Check
Copy Values – Copy the values to clipboard
Paste Values – Paste the values you copied earlier
Remove – Remove the Value Check for this Sequence / Modifier

At the top of the Values Tab you can find a configuration for the grey guidelines and an option to snap to them as well. This helps you organizing and aligning multiple sequences to the same values.

At the bottom of the window you can find an area that allows you to quickly create a new Value Requirement by dragging a Sequence or modifer onto it:

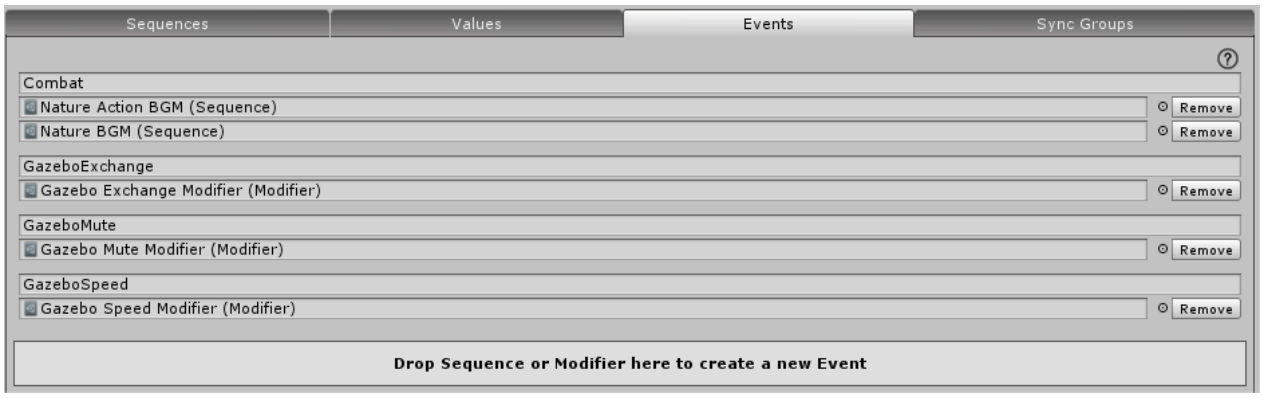
This list is not only useful for aligning your Value Requirements across your Sequences and Modifiers, but also for debugging. If you open this tab during runtime and a Value is set by your game, the current Value is visible in this tab as well. This can help to debug your Value Requirement Checks in real time.

Here you can see the Time of Day Value Requirement from the demo scene represented as white line in the Values Tab:



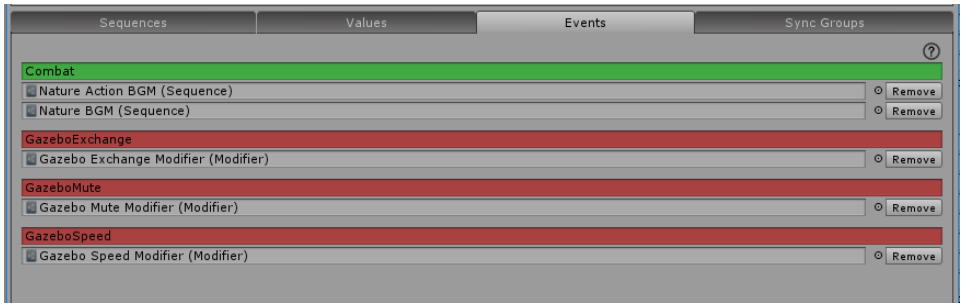
Events Tab

Similar to the Values Tab, the Events Tab lists all Events used in the scene and names the Sequences and modifiers that are using them.



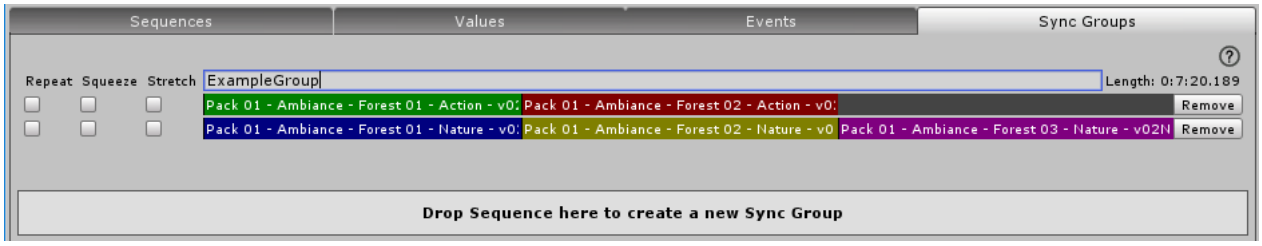
You can remove a Sequence from an Event quickly with the remove button, and create new Events as well by dragging a Sequence or Modifier in the area at the bottom:

During runtime the Events Tab displays which Event is active and which is not by displaying the Event Name boxes in green (=active) and red (=not active).



Sync Groups Tab

The Sync Groups Tab allows you to view and edit all Sync Groups in your scene. Editing Sync Groups is most convenient to do from this tab, since it will give you a visual representation of the effect of the synchronization settings.



You can create a new Sync Group by dragging and dropping a Sequence on the area at the bottom of the window.

You can edit existing Sync Groups the following way:

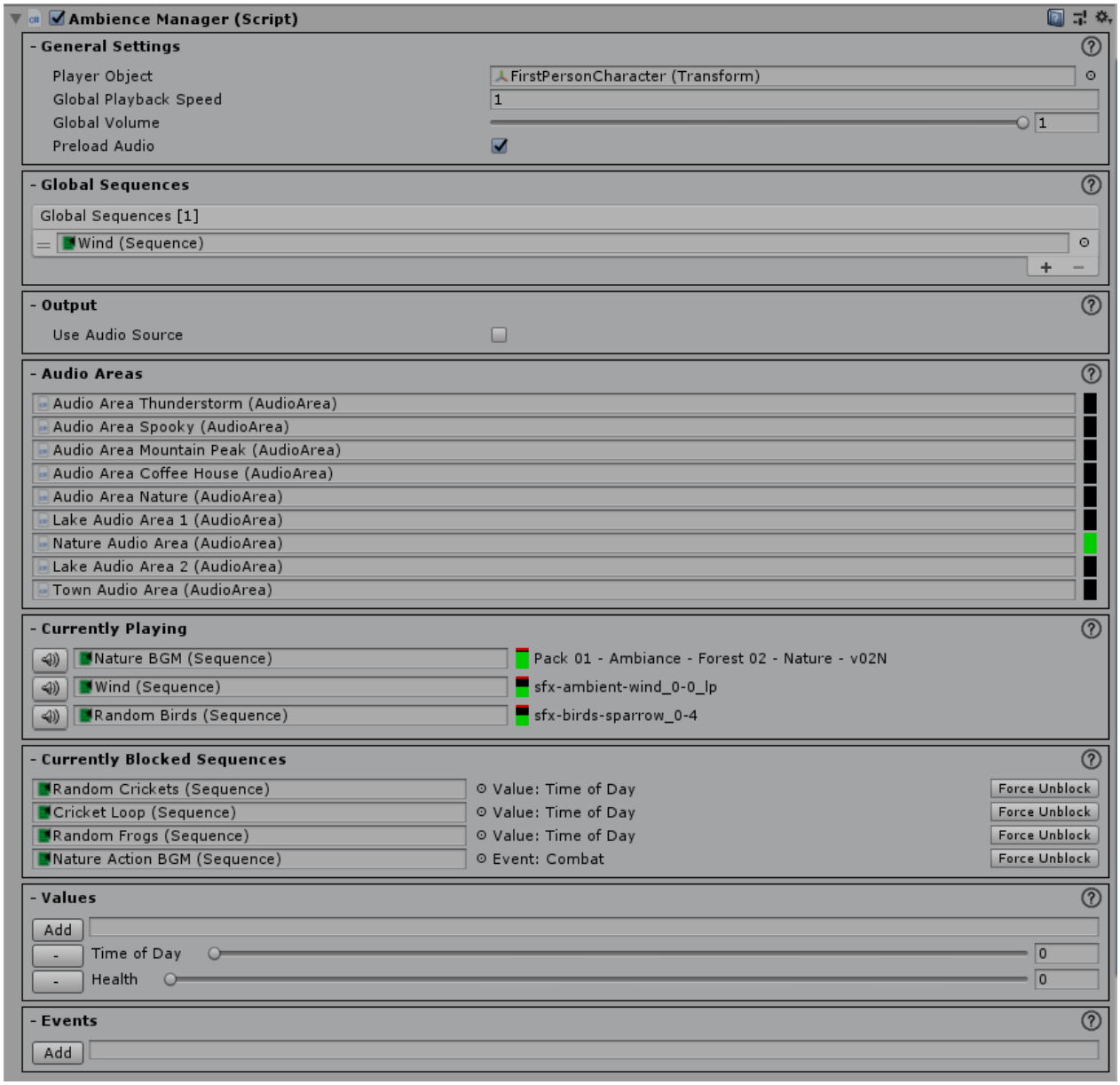
- Rename the Sync Group by entering a new name
- Click the checkboxes in front of each Sequence row to set the synchronization mode of that Sequence. Note that this will update the total duration and the visual representation of Sequences in the Sync Group according to your changes. This gives you visual confirmation of your changes to the group, which is far easier than editing these settings from the Sequence component instead.
- Drag and Drop a Sequence from the asset hierarchy on an existing Sync Group to add the Sequence to the group
- Click the Remove button to remove a Sequence from the group

Sync Groups and the effect of the individual synchronization settings will be covered in the next chapter in more detail.

The Ambient Sounds Components in Detail

This chapter lists the different Ambient Sounds components and describes all features and settings in detail. If you read through the manual from the start to this point, you should know enough to start working with Ambient Sounds already. You should only need to come back for this chapter as a reference or if you want to know more about a specific feature. Remember that you can use the help button on each component to get on screen help for the features as well.

Ambience Manager



The Ambience Manager is responsible for tracking your player and evaluating all requirements and conditions of Sequences and Modifiers to determine the current

Audio Output.

The Ambience Manager is a Unity component that can be attached to any GameObject in theory, but it will reattach itself to the GameObject containing the AudioListener for the scene during runtime.

The Ambience Manager can be configured with the following settings / features:

General Settings

Player Object	This is a reference to the player transform. This transform is used for multiple purposes, but mainly to check if the player has entered and Audio Area, so the Sequences of that Area can be played back (as long as the Requirements of that Sequence are met as well.) Generally, this should be set to the Camera or Player object so that the Sequences are activated as the player moves through the game world. This transform will also be used when you decide to play back audio relative to the player's position.
Global Playback Speed	This is the global playback speed applied to all sounds / Sequences played back by the Ambience Manager. If you want to increase / decrease the playback speed of all Sequences at the same time, this would be the value to change.
Global Volume	This is the global volume applied to all sounds / Sequences played back by the Ambience Manager. If you want to increase / decrease the Volume of all Sequences at the same time, this would be the value to change.
Preload Audio	Determines whether the Audio clips in the Sequences will be loaded in the Awake() Event of the Ambience Manager, or when they are used the first time. Usually you would want to keep this activated, unless you got that many Audio Clips in your project that it creates problems loading them on Startup.

Global Sequences – Sequences added in this list will be played back everywhere in the Scene regardless of the player's position (as long as the Requirements of that scene are met). You can also add a Sequence here quickly during runtime for testing purposes.

Output - Checking the "Use Audio Source" box here allows you to select a Unity Audio Source for playback or have one created for you during runtime. This allows you to further control Audio Playback on the Audio Source component directly such as routing it into a Unity mixer.

Once you check this box, further options appear for the setup of the Audio Source:

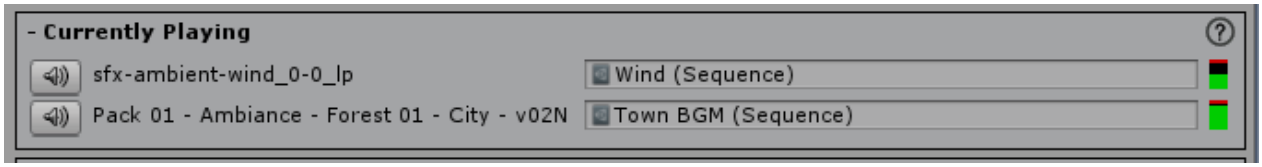
Audio Source Prefab	You can supply a prefab here with an Audio Source attached that comes with your own settings. It is also possible to have additional Components attached to that Prefab that further might influence Audio Playback, such as a Low Pass Filter for example. If you don't select any prefab here, Ambient Sounds will create one for you during runtime with default settings.
Audio Source Channels	Determines how many Audio Channels the created Audio Source will utilize.

During Runtime the Ambience Manager will display additional Entries as well:

Audio Areas – List of all Audio Areas in the scene with their current fade state. The green bar represents how far this area has been faded in according to the player's position. This list is useful to see which Audio Areas are currently playing. You can click on the Audio Area names in this list to select the corresponding Audio Area in the scene hierarchy.

Added Sequences – List of all Sequences that were added on the fly by pressing the "Play" Button on top of the Sequence Editor during runtime, or by an API Call.

Currently Playing – List of all Sequences that are currently playing. This list is useful to identify the Sequences that are contributing to the current Audio output. The bar graph on the left side displays both the fade state of the sequence as the current volume of the clip that is playing right now. Here is an example:



Both Wind Sequence and Town BGM sequence are fully faded in. The green bar indicates that the wind sound is played back at half volume, while the Town Music is nearly at full volume.

You can click on the Sequence names to open the corresponding Sequence in the asset hierarchy.
The small loudspeaker button allows you to mute / unmute the tracks so you can quickly analyze which Sequence is contributing to which audio output in the scene.

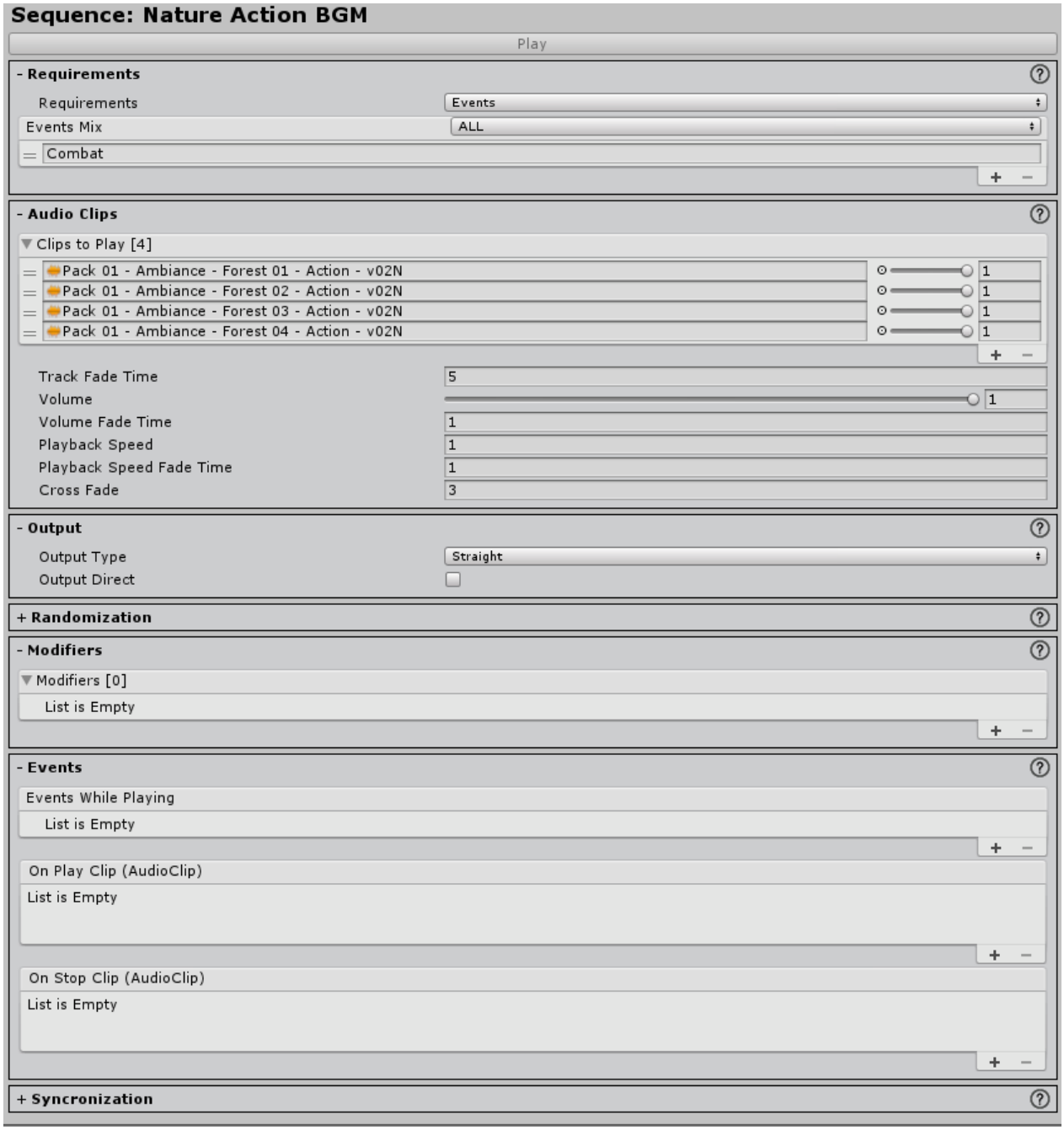
Currently Blocked Sequences – List of all Sequences which are currently blocked from being played back since the sequence does not meet the necessary requirements. You can see right next to the sequence name which value or event would be required to make the sequence play back, also there is a "Force Unblock" button which will create the necessary requirements to play back the sequence. Note that this might conflict with your gameplay logic to a certain degree, e.g. if you have a script running that constantly sets a certain event as inactive and you try to run a sequence via this button, your script might deactivate the event right away again and the sequence will not play.

Values – List of all currently active Requirement Value Checks. You can modify the Values directly here in this list to influence Sequences or Modifiers that check against these Values in their Requirements. Note that this list will only show Value Checks that have its current value set at least once during gameplay. If you – for example – set up a check against the player's health in a modifier but then never update the "PlayerHealth" value in Ambient Sounds during gameplay, the value will not display here yet.
The list allows you to quickly add a value that has not been set by your gameplay logic yet so you can just inject any value to test Requirements on the fly. To view a list of all Value checks regardless if their value has yet been set, you can use the Values Tab in the Ambient Sounds Manager Window.

Events – List of all currently active Events in the scene. You can add and remove Events directly in this list to influence Sequences or Modifiers that check against these Events in their Requirements.

Note that this list will only show **active** Events. There might be additional Events set up that your Sequences and Modifiers listen to that are currently not active. To view a list of all Events, active or not, you can use the Events Tab in the Ambient Sounds Manager Window.

Sequences



Sequences are a combination of clips that are played after one another in ordered or random fashion with common playback parameters.

You can create a new Sequence in your project from the Create menu in the Asset Hierarchy:

Procedural Worlds > Ambient Sounds > Sequence

Or from the "Sequences Tab" in the Ambient Sounds Manager Window.

A Sequence can be defined with the following settings / features:

Play Button – At the top of the Sequence you can find a play button that will play back the Sequence immediately in the Ambience Manager. **It is available during runtime / play mode only** since it requires a running Ambience Manager for playback.

When clicking on the Play Button, the Sequence will be added to the "Added Sequences" list in the Ambience Manager and plays back immediately. You can then stop playback from this button or from the Ambience Manager as well.
This is extremely valuable when you are working on a Sequence since it allows you to start and stop the sequence immediately while trying out different playback parameters.

Requirements – This is the Requirement setup that controls whether this Sequence will be active for playback or not. Requirements will be explained in detail in their own chapter below since they are a bit more complex and operate the same for both Sequences and Modifiers.

Audio Clips – The list of Audio Clips that will be played back in this Sequence. You can fill this list by increasing the list size with the plus button at the bottom of the list and drag and drop from the asset hierarchy, or you can use the file picker next to each list entry. You can set an individual volume for each clip, and you can also rearrange the order of the clips via drag and drop.
All Audio Clips will be played back with the following shared playback settings:

Track Fade Time	The time in seconds the clips take to fade in or out when start or stopping playing.
Volume	The volume level of this Sequence alone. Note that other Sequences might be played back at a different volume, if you want to change the overall volume of Ambient Sounds you need to change the global volume on the Ambience Manager instead. Please note further that each individual clip in the sequence has a volume level as well – this allows you to bring clips with different source volume into alignment quickly.
Volume Fade Time	The time in seconds this Sequence takes to react to volume changes during runtime. This is useful to make volume changes play out smoothly over time instead of instantly when changing the volume, e.g. by a Modifier that was activated.
Playback Speed	The playback speed for the clips in this Sequence. A value of 1 would be normal playback speed, 0.5 half the speed, 2 double the speed, and so on. Note that this will also change the pitch

	of the playback like speeding up a record player, so this is not a pure musical tempo change.
Playback Speed Fade Time	The time in seconds this Sequence takes to react to playback speed changes during runtime. This is useful to make playback speed changes play out smoothly over time instead of instantly when changing the playback speed, e.g. by a Modifier that was activated.
Cross Fade	The time in seconds to cross fade between Audio Clips played back without any delay.

Output

Each Sequence has its own output options that determine in which technical way the output is played back. You can choose between the following methods to output Audio:

- Straight - Audio is sent directly into the Audio Listener in the scene
- Local Position – Creates an Audio Source at the position of this Sequence
- Player Position – Creates an Audio Source at the position of the Player (= the tracked player transform in the Ambience Manager)

You can also select "Output direct" if you want to play back directly in the Audio Source without any extra processing in Ambient Sounds. Normally, the system outputs a stream to the audio listener doing cross-fades and mixing all the tracks into one audio stream.
If you turn on "Output direct" it will instead place audio sources and put the clips that need played on them, allowing Unity to do the playback in the normal audio system. Ambient Sounds will then just control playback speed and volume, with no cross-fades or other processing.

When choosing "Local" or "Player", the following options appear to control the Audio Source creation:

Output Prefab	Allows you to set up a prefab with an Audio Source for playback. In this way you can supply a prefab with a pre-configured Audio Source and extra components that will be used for playback. Use this when you want to route your playback into a certain Unity mixer group or if you want to add additional Audio Effects that require access to the Audio Source as a component.
Distance	A minimum and a maximum distance in which the Audio Source will be created. Set min and max to the same value to make the Audio Source appear in a fixed distance. With different min max values you can randomize the distance for each playback

	which is great for Ambient Sound Effects. Note that the distance will be relative towards the output position selected above (Local or Player).
Vertical Angle	A minimum and a maximum vertical angle (above or below the position) at which the Audio Source will be created. Set min and max to the same value to make the Audio Source appear in a fixed vertical angle. With different min max values you can randomize the angle for each playback which is great for Ambient Sound Effects. Note that the angle will be relative towards the output position selected above (Local or Player).
Horizontal Angle	A minimum and a maximum horizontal angle (before or behind the position) at which the Audio Source will be created. Set min and max to the same value to make the Audio Source appear in a fixed horizontal angle. With different min max values you can randomize the angle for each playback which is great for Ambient Sound Effects. Note that the angle will be relative towards the output position selected above (Local or Player).
Follow Position	Check this box if you want the Audio Source to follow its parent (Player or Audio Source) when it moves. It depends of the kind of sound you are playing back if you need to enable this setting. For Random Environment Sounds you normally want this off as the sound originates from the world itself and would not follow the player around as he moves, e.g. Bird Sounds. For Random Effects that are tied to the player, e.g. the sound effects of a magical spell that the player is weaving, this setting can be required.
Follow Rotation	Check this box if you want the Audio Source to follow its parents (Player or Audio Source) rotation. As with "Follow position" this highly depends on the sound you are playing back.

Note that Audio Areas have their own output options as well. When Output options between Sequences and Audio Areas are conflicting, Ambient Sounds will try to merge them the best way possible. E.g. if an Audio Area is set to "Straight" and has 5 Sequences, one of them with output type "Player Position" that one Sequence will be put out at player position, the other 4 will be played "Straight".

Randomization

The audio clips can be played back while randomizing the order, volume and delay between tracks. The single randomization options are as follows:

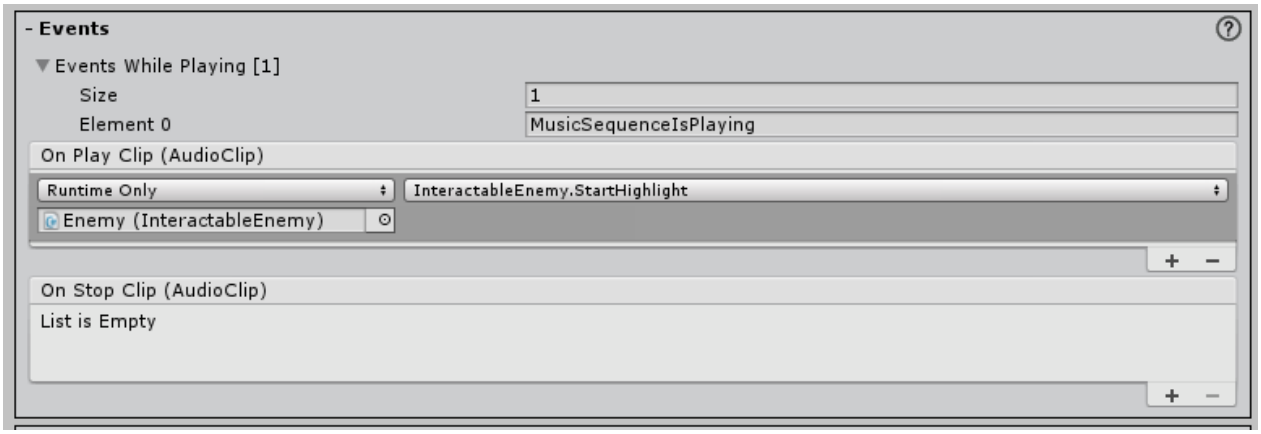
Randomize Order	Play back the clips in a randomized order. Leave unchecked if you want the clips to be played back in the same order as you listed them above.
Delay Chance	Chance in percent a delay will be applied in between playing back clips.
Delay Fade Time	Time in seconds the currently playing track will take to fade out when a delay is applied.
Min / Max Delay	Minimal and maximal time in seconds for the random delay (if applied). Add the same min-max value here if you want the exact same delay every time. (e.g. for a consistent delay between musical tracks) With a different min max value the delay time is randomized again between those values after each playback which is great for Ambient Sound FX.
Randomize Volume	Randomize the volume on each clip playback. Leave unchecked if you want your clips played back with the same volume each time.
Min / Max Volume	Minimum and maximum volume that will be used when "Randomize Volume" is checked. The random volume will be determined anew after each playback which is great for Ambient Sound FX.
Randomize Playback Speed	Randomize the playback speed on each clip playback. Leave unchecked if you want your clips played back with the same speed each time.
Min / Max Playback Speed	Minimum and maximum speed that will be used when "Randomize Playback Speed" is checked. The random speed will be determined anew after each playback which is great for Ambient Sound FX. Slightly modifying the playback speed every time makes it less apparent that the same sounds are played back every time.

Modifier

Modifier can be used to override some of the playback parameters for this Sequence. This allows you to reduce the volume for this Sequence when a certain Event happens in your game for example. Modifiers are applied to a Sequence in this list. You can fill this list by increasing the list size and drag and drop modifiers from the asset hierarchy into it, or you can use the file picker next to each list entry.

When multiple active Modifiers have conflicting overrides for the same setting, the settings of the later Modifier in the Sequence's Modifier List will be applied. Note that Modifiers have their own settings as well which will be described in the next chapter.

Events



You can automatically trigger Events when this Sequence is playing. This can be useful to create interdependencies between your Sequences without writing a single line of code. For example you could apply a Modifier to Sequence A that reduces its volume when Sequence B starts playing.

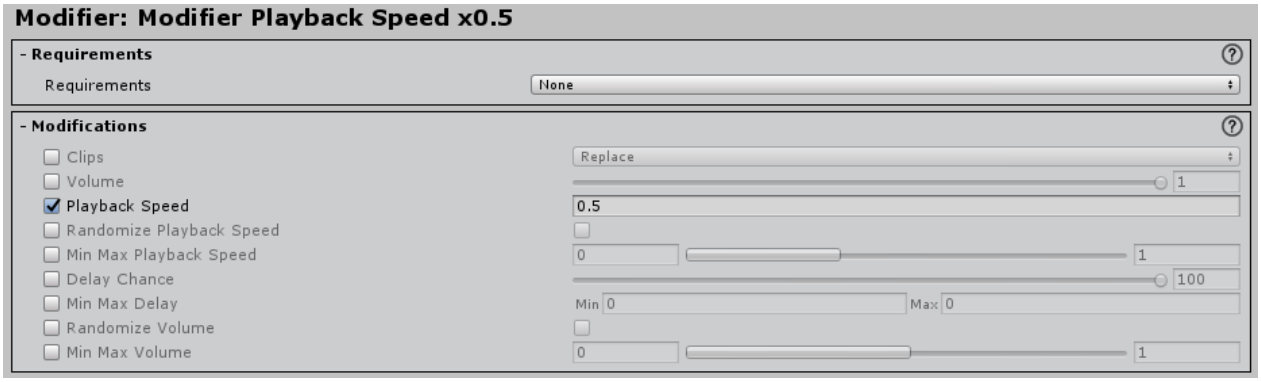
To Add Events you want to trigger simply increase the list size for the "Events While Playing" List and type in the Event Name.

This section also allows you to perform actions when a clip starts or stops playing. To configure this create a new entry via the small + button and select the object and the corresponding function you want to call when the clip starts or stops.
Note that the API offers you Events that you can attach to in programming as well. See the API section at the end of the manual for more information.

Sync Groups

You can assign Sequences to Sync Groups to synchronize them with each other. You can enter a Sync Group name here and select a combination of the different Sync Group modes to apply for this Sequence in this dropdown. The options of this dropdown correlate with the options available in the Sync Group Tab in the Ambient Sounds Manager Window. Note that Sync Groups will are described in detail in one of the following chapters.

Modifier



Modifiers allow you to override the clips and output parameters of a Sequence and tie the application of this modification to certain Requirements. When the Requirements are met, the Modifier will be active and the modifications will be applied to the Sequence. If the Requirements are not met anymore, the Modifier will be deactivated and the modifications will be undone on the Sequence.

You can create a new Modifier in your project from the Create menu in the Asset Hierarchy:

Procedural Worlds > Ambient Sounds > Modifier

Or from the "Sequences Tab" in the Ambient Sounds Manager Window.

A Modifier can be set up with the following features / settings:

Requirements – This is the Requirement setup that controls whether this Modifier will be active or not. Requirements will be explained in detail in their own chapter below since they are a bit more complex and operate the same for both Sequences and Modifiers.

Modifications – These are overrides for the corresponding settings of the Sequence this Modifier will be attached to. To use one of the overrides, activate the checkbox next to it. You can decide to use one, multiple or all modifiers at once.

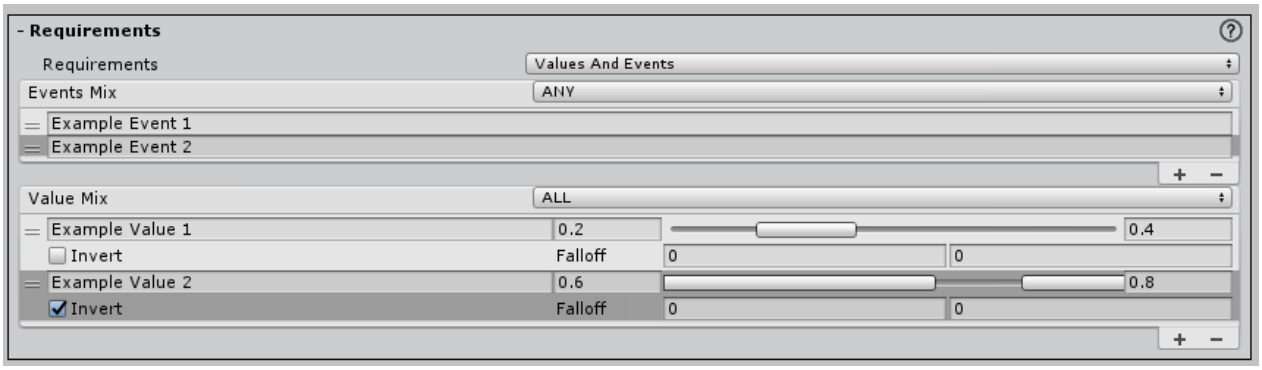
The overrides will influence the Sequence as follows:

Clips	Adds To, Removes From, or Replaces the List of Clips in the Sequence. Select one of the options from the dropdown and then supply a list of clips you want to add / remove / replace. The clips you supply here will be used to modify the original Clip list in the Sequence as follows: **"Add"** will add the clips from the Modifier at the end of the clip list in the Sequence

	"Remove" will look for the clips from the Modifier in the Sequence Clip list and remove them from the Sequence if found. **"Replace"** will replace the entire clip list in the Sequence with the one from the Modifier
Volume	Will override the playback volume for the Sequence.
Playback Speed	Will override the playback speed for the Sequence.
Randomize Playback Speed	Will override the "Randomize Playback Speed" setting in the Sequence
Min Max Playback Speed	Will override the minimum and maximum **randomized** Playback Speed in the Sequence.
Delay Chance	Will override the Delay Chance for the Sequence.
Min Max Delay	Will override the Minimum and Maximum delay for the Sequence.
Randomize Volume	Will override the "Randomize Volume" setting in the Sequence
Min Max Volume	Will override the minimum and maximum **randomized** Volume in the Sequence.

You can apply one or more Modifiers to a Sequence at the same time. When multiple active Modifiers have conflicting overrides for the same setting, the settings of the later Modifier in the Sequence's Modifier List will be applied.

Requirements: Values and Events

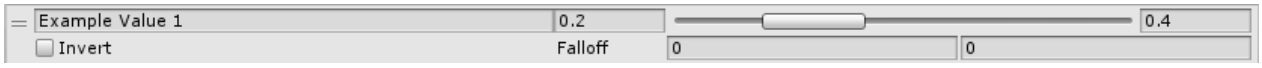


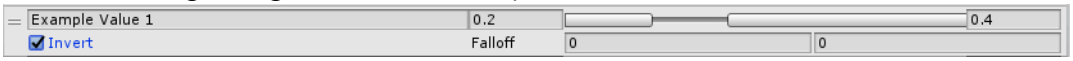
Even though Requirements are not really an individual component in Ambient Sounds, they play an important role in setting up the interactivity for your soundscapes. They work just the same in both Sequences and Modifiers which is why they are explained in detail here in their own chapter.

Requirements for both Sequences and Modifiers can consist of Value Checks and Events:

Value Checks

Value Checks will listen for a current value from your project like health, time, money etc. and check if that value falls into a predefined range. If that is the case, the Requirement is fulfilled. A Value Check can be set up with the following features / settings:



Name	A name to identify the Value Check. It is important to keep the name consistent across your project if you want to check against the same value in multiple requirements. If you enter "Health" and "health" for example, this will be treated as two separate values by Ambient Sounds. The name does not necessarily need to be the same as the variable name in your project, but you will need to know this name when updating Ambient Sounds with the current value from your gameplay logic.
Range	A Value Range to check against. This value range is always normalized to lie between 0 (minimum) and 1 (maximum). This does not mean all values in your game need to lie between 0 and 1 as well, but you will need to normalize them before updating Ambient Sounds with the current value. The Quick start Guide in this Manual has an example for this.
Invert	Will Invert the range check to make the outside areas positive and the selected range negative. The example screenshot from above Inverted: 
Falloff	You can enter one minimum and one maximum Falloff values here. Rather than activating / deactivating the Sequence or Modifier instantly it will be faded in across the Falloff range.

Once the requirement value check has been set up, you can set the actual value by sending an update from your game's code:

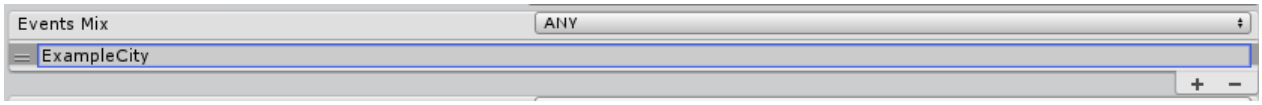
```
AmbientSounds.AmbienceManager.SetValue("ValueName",0.5f);
```

When and in which frequency you want to do this update is entirely up to you and depends on the value at hand as well. As a rule of thumb you should want to send an update every time the value changes in your game, unless that value changes multiple times per frame. Then it might be better to consolidate that change in a single update for Ambient Sounds.

The moment the Value is set with the call above, Ambient Sounds will re-evaluate all Sequences and Modifiers and change the resulting output if required.

Events

Events will check if there is an Event of the same name currently active in Ambient Sounds. If that is the case, the Requirement is fulfilled. This simple elegance results in a simple setup as well:



Just enter the name of the Event you want to listen to and that is the entire setup that is required. It is important to keep the name consistent across your project if you want to check against the same Event in multiple requirements. If you enter "Combat" and "combat" for example, this will be treated as two separate values by Ambient Sounds.
Note that it is entirely up to you and your imagination what an "Event" is or can be. It is just a thing that can be active or not, so you can use it for anything imaginable:

PlayerRidesOnHorse
PlayerDied
ConstructionComplete
ItemPickedUp
NewGameStarted

Etc.

The name does not need to be the same as those Events are named in your project. You might even not process those Events at all in your game if they are only relevant for Ambient Sounds. But you will need to know this name when you want to activate or deactivate that Event from your gameplay logic.

To activate an Event in Ambient Sounds you can call:

```
AmbientSounds.AmbienceManager.ActivateEvent("EventName");
```

To deactivate an Event in Ambient Sounds you can call:

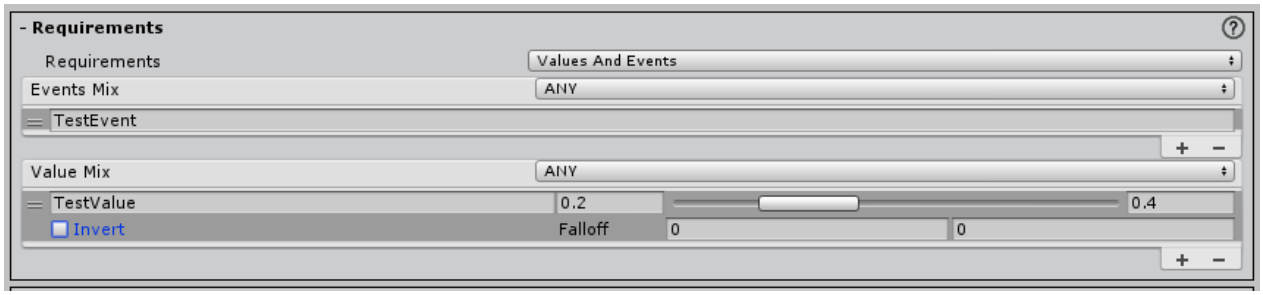
```
AmbientSounds.AmbienceManager.DeactivateEvent("EventName");
```

As with the Value Checks it is entirely up to you when and in which frequency you want to activate or deactivate these Events in Ambient Sounds.

Setting up Requirements

Value Checks and Events are combined in Requirements that need to be fulfilled before a Sequence or Modifier becomes active in Ambient Sounds. It is possible to set up multiple Value Checks and Events at once and even define if you want none, any or all of them to be positive so that the Requirement is fulfilled.

You will find the setup for Requirements at the very top in both Sequences and Modifiers and they work exactly the same for both.



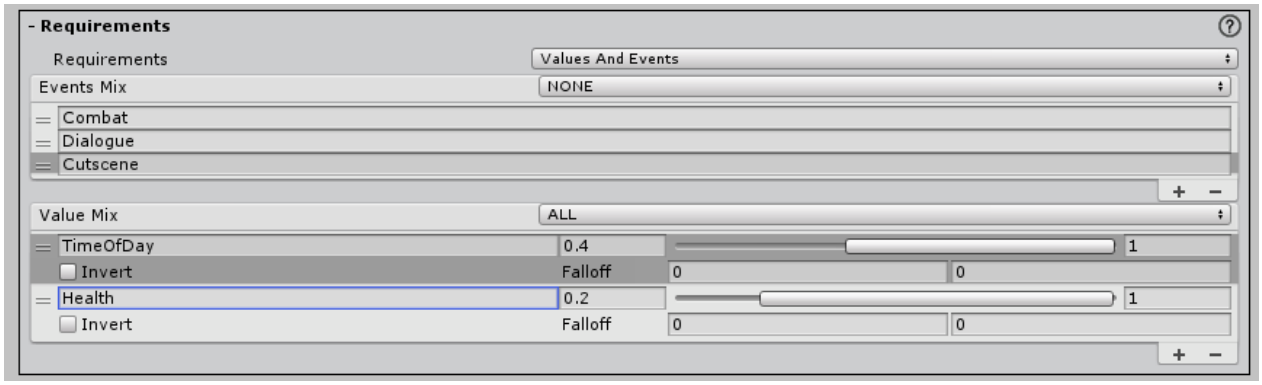
First you can select in the Requirements if you want to check for:

None	No Requirements are set up for this Sequence / Modifier (default)
Values	Only Values will be evaluated
Events	Only Events will be evaluated
Values Or Events	Both Values and Events will be evaluated, as soon as the Value Checks **OR** the Event Checks come out positive, the Requirements are fulfilled. The "OR" is an "inclusive or", which means the Requirement will also be fulfilled if both Values and Events are positive.
Values And Events	Both Values and Events will be evaluated, and both must come out positive for the Requirements to be fulfilled.

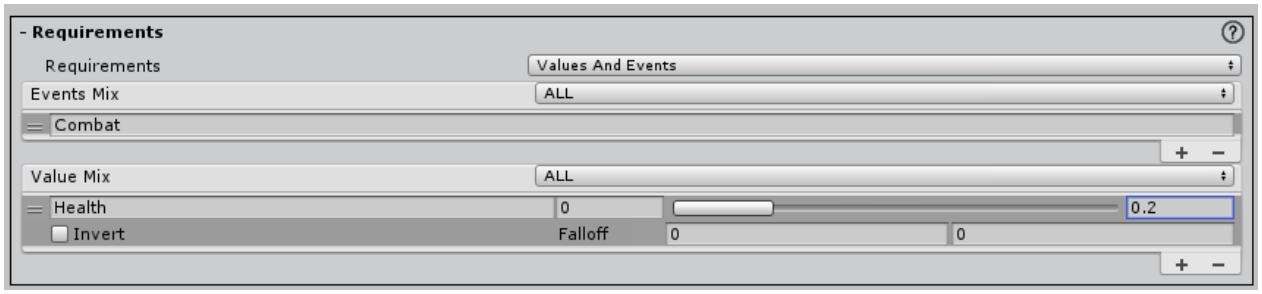
Depending on what you select here, you can select a "Mix" for both groups (Values & Events) how these should be evaluated:

ALL	All entries in the list need to be positive for the requirement to be fulfilled.
ANY	Only one or more entries of the list are enough for the requirement to be fulfilled.
NONE	No entry of the list may be positive for the requirement to be fulfilled.

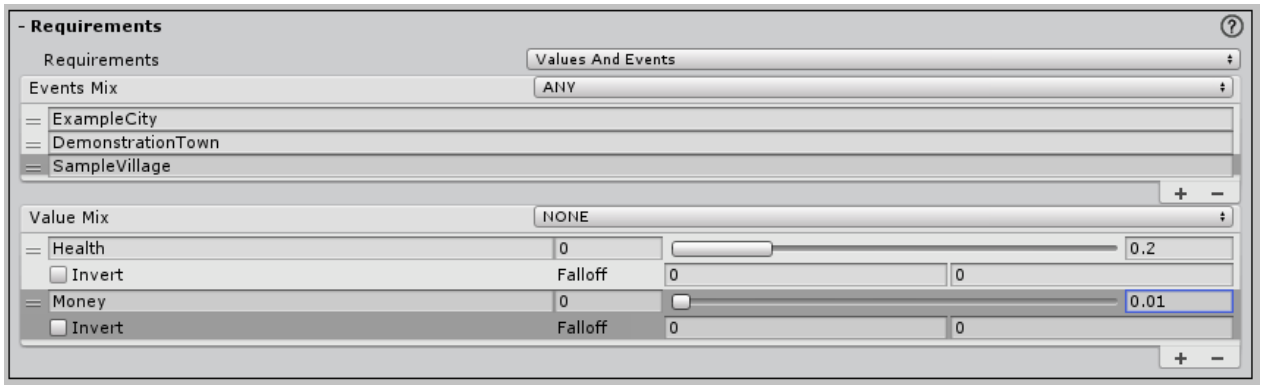
The logic dependencies between Values and Events and their mixes allow for complex condition checks without writing a single line of code. Here are some examples:



This Requirement will only be fulfilled if there is no Combat, no Dialogue and no Cutscene Event active, and if the time of day is in the second half and the player is somewhat healthy.



This Requirement would be fulfilled if the player is in combat **OR** low on health.



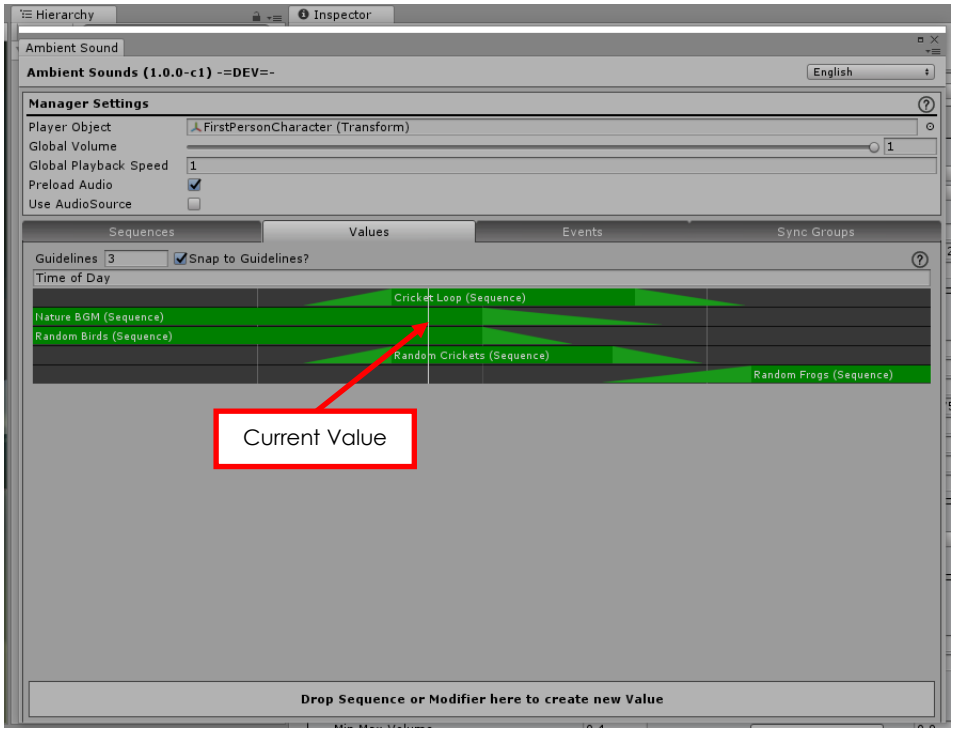
This Requirement would be fulfilled if the player is in one of the three locations ExampleCity, DemonstrationTown or SampleVillage, and is in good health and not poor. (Note Value Mix = NONE is enabled, which means none of the value checks must be positive!)

If these combinations are still not granular enough for your Ambient Sounds setup, you could make a duplicate of the Sequence / Modifier to gain an additional set of Requirements. In this way you can create a second set of Requirements that will lead to the same result when fulfilled.

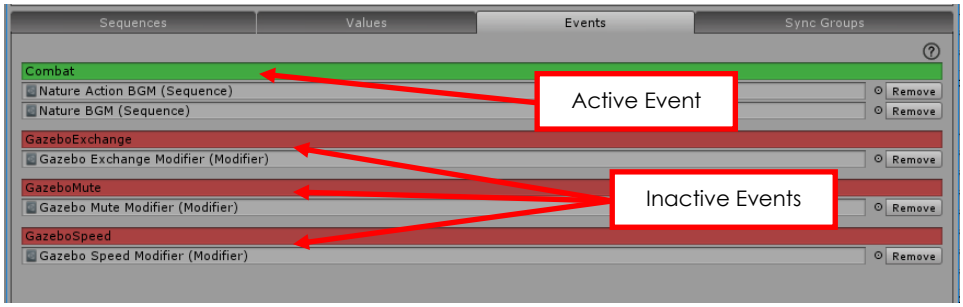
Debugging Requirements

If you need to debug Requirements keep in mind that there is an overview for both Values and Events in the Ambient Sound Manager window that shows the state of both Values and Events during runtime. There is also a List of "Currently Blocked Sequences" in the Ambience Manager and the Monitor Tab.
If you find yourself wondering why a Sequence / Modifier is not active, look at its Requirements first and then cross check the Values and Events tab during runtime. The "Currently Blocked Sequences" List should tell you as well why a certain Sequence is not playing right now.
The information in these views should help you get your issue sorted out quickly:

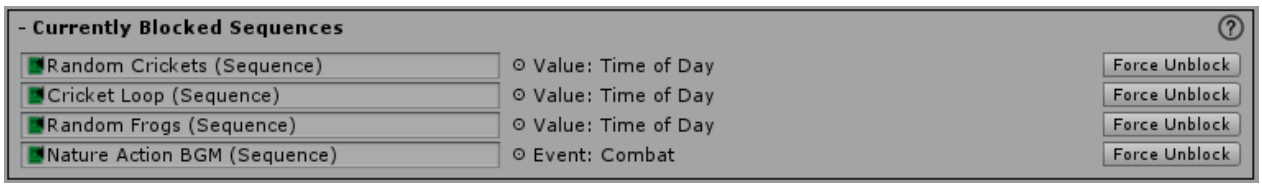
Values Tab during runtime:



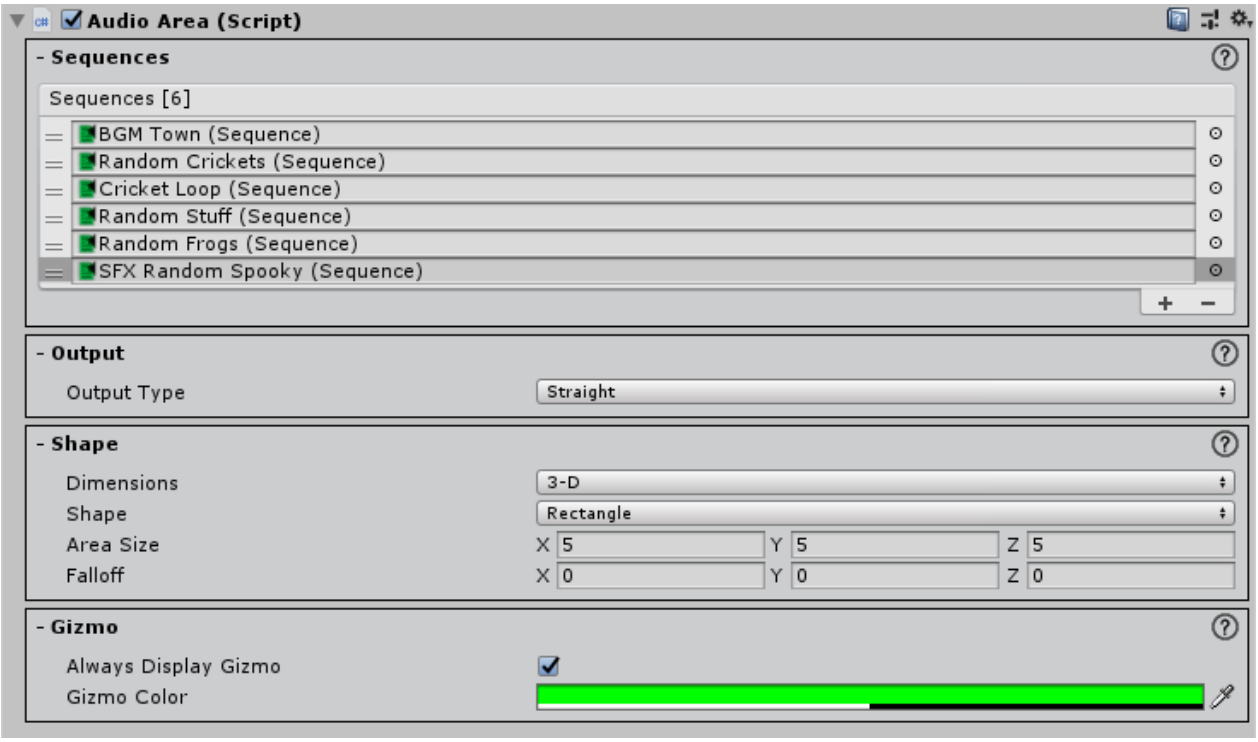
Events Tab during runtime:



List of „Currently Blocked Sequences“ in the Ambience Manager:



Audio Areas



Audio Areas play back one or more Sequences in a defined Area in the game world when the player enters them.

You can create a new Audio Area in your project from the Create menu in the Scene Hierarchy:

Procedural Worlds > Ambient Sounds > Audio Area

Or from the "Sequences Tab" in the Ambient Sounds Manager Window.

Once created, an Audio Area can be set up with the following Settings / Features:

Sequences

The List of Sequences that should be played back once the player enters the area. Sequences will only play back if they are active because all their Requirements have been met. Note that multiple Sequences will be played back in parallel. To play back multiple Audio Clips one after another you would need to create a single Sequence with multiple Clips instead. To add Sequences to this list, simply increase the list size with the small "plus" button at the bottom of the list and drag and drop from the asset hierarchy or use the file picker dialogue.

Output

Each Audio Area has its own output options that determine in which technical way the output is played back. You can choose between the following methods to output Audio:

- Straight - Audio is sent directly into the Audio Listener in the scene
- Local Position – Creates an Audio Source at the position of this Audio Area
- Player Position – Creates an Audio Source at the position of the Player (= the tracked player transform in the Ambience Manager)

When choosing "Local" or "Player", the following options appear to control the Audio Source creation:

Output Prefab	Allows you to set up a prefab with an Audio Source for playback. In this way you can supply a prefab with a pre-configured Audio Source and extra components that will be used for playback. Use this when you want to route your playback into a certain Unity mixer group or if you want to add additional Audio Effects that require access to the Audio Source as a component.
Distance	A minimum and a maximum distance in which the Audio Source will be created. Set min and max to the same value to make the Audio Source appear in a fixed distance. With different min max values you can randomize the distance for each playback which is great for Ambient Sound Effects. Note that the distance will be relative towards the output position selected above (Local or Player).
Vertical Angle	A minimum and a maximum vertical angle (above or below the position) at which the Audio Source will be created. Set min and max to the same value to make the Audio Source appear in a fixed vertical angle. With different min max values you can randomize the angle for each playback which is great for Ambient Sound Effects. Note that the angle will be relative towards the output position selected above (Local or Player).
Horizontal Angle	A minimum and a maximum horizontal angle (before or behind the position) at which the Audio Source will be created. Set min and max to the same value to make the Audio Source appear in a fixed horizontal angle. With different min max values you can randomize the angle for each playback which is great for Ambient Sound Effects. Note that the angle will be relative towards the output position selected above (Local or Player).

Follow Position	Check this box if you want the Audio Source to follow its parent (Player or Audio Source) when it moves. It depends of the kind of sound you are playing back if you need to enable this setting. For Random Environment Sounds you normally want this off as the sound originates from the world itself and would not follow the player around as he moves, e.g. Bird Sounds. For Random Effects that are tied to the player, e.g. the sound effects of a magical spell that the player is weaving, this setting can be required.
Follow Rotation	Check this box if you want the Audio Source to follow its parents (Player or Audio Source) rotation. As with "Follow position" this highly depends on the sound you are playing back.

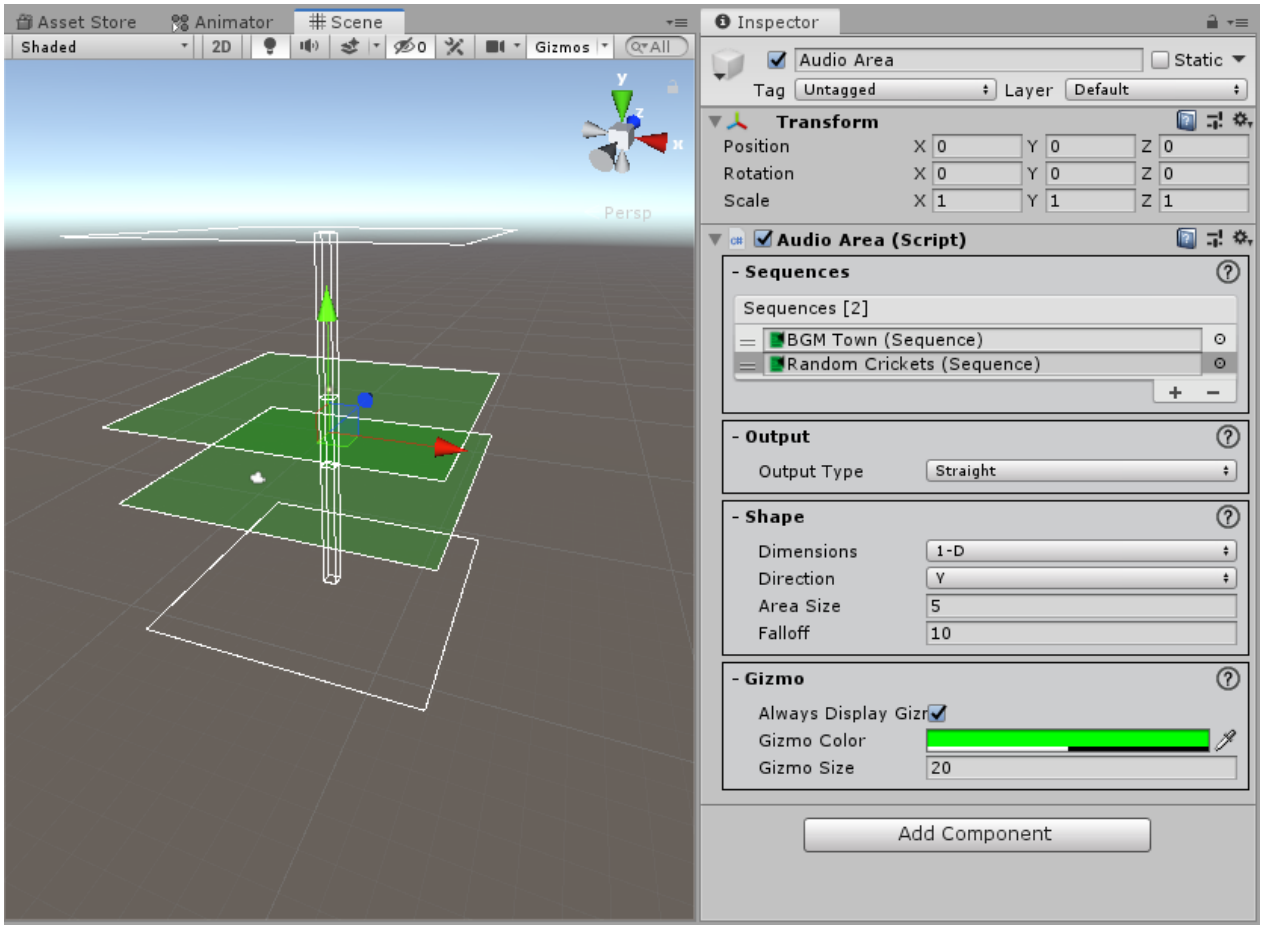
Note that each Sequence has their own output option as well. When Output options between Sequences and Audio Areas are conflicting, Ambient Sounds will try to merge them the best way possible. E.g. if an Audio Area is set to "Straight" and has 5 Sequences, one of them with output type "Player Position" that one Sequence will be put out at player position, the other 4 will be played "Straight".

Shape

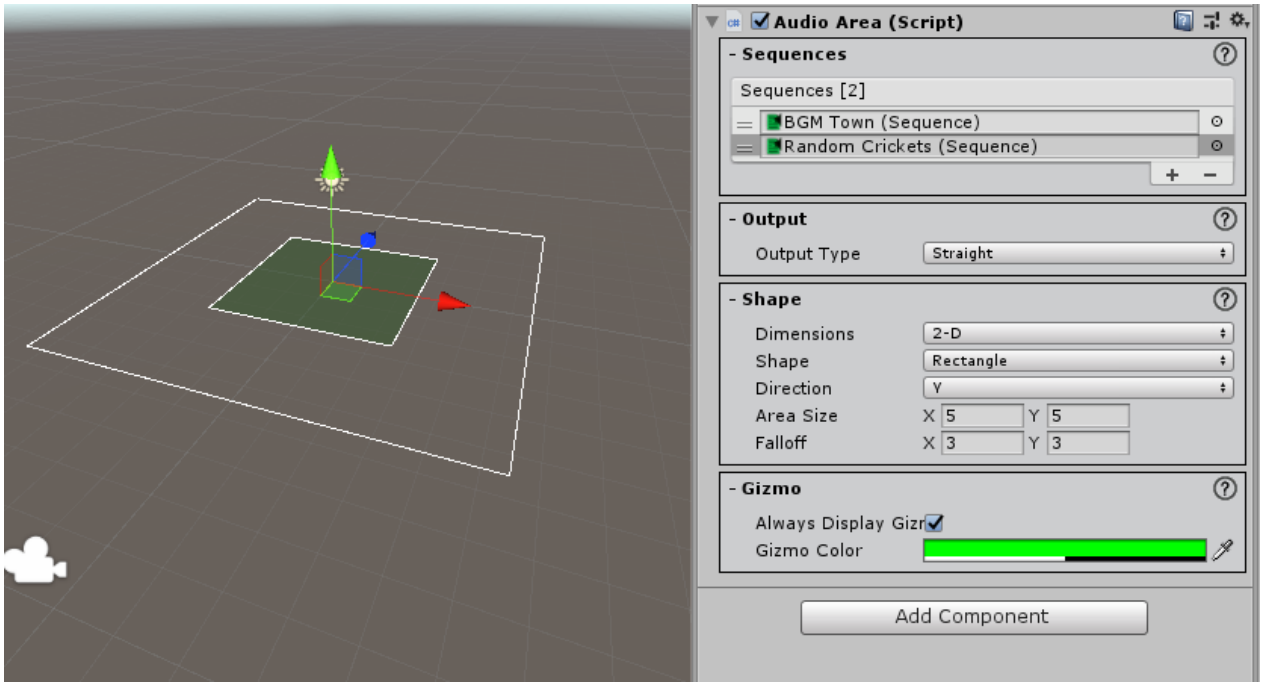
Each Audio Area has its own distinctive shape in the game world, and the associated Sequences will only be played back if the player is inside this area. Shapes can be set up with the following settings:

Dimensions	Determines how many dimensions / axes this shape should take into account when checking for the player's position. 1-D: Check will only take place in the direction of 1 axis. 2-D: Check will take place on 2 axes. 3-D: Check will take place on all 3 axes.
Direction	Sets the direction / orientation for 1D and 2D shapes.
Shape	Allows you to change between a sphere or a rectangle shape for 2D and 3D shapes.
Area Size	Allows you to define the size of the shape. Note that the size options change according to the number of dimensions selected.
Falloff	Size of the falloff area around the actual shape in which the playback will be faded out. This is good for having background music fading in gradually as the player enters a location. Note that the falloff options change according to the number of dimensions selected.

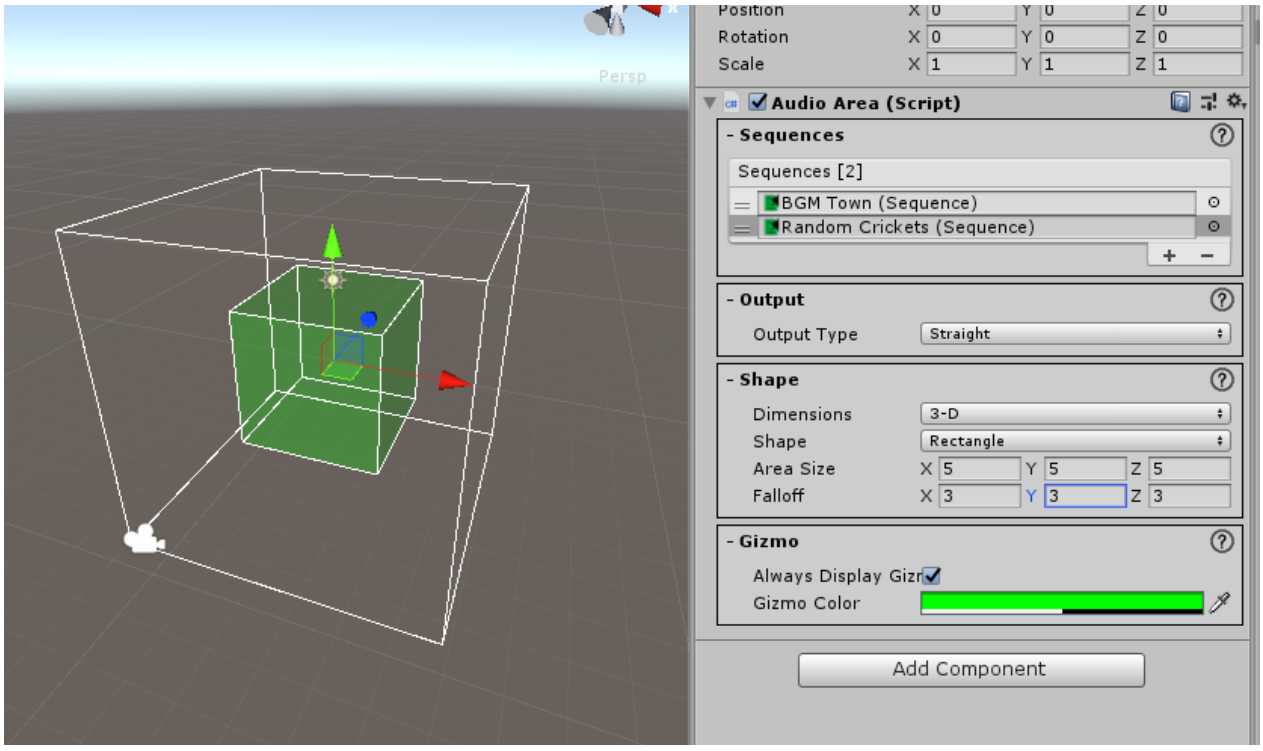
The shape settings might seem confusing at first, but should become clearer when you watch the shape gizmo in the scene while changing these options. Here are a few examples:



One-dimensional check: This Area would become active as long as the player's y-position is within 5 units of this area. The X and Z position would not matter at all. Note that the Gizmo for the one dimensional area is drawn as a exaggerated box shape with additional planes to symbolize the affected levels of the area and the falloff. This may be geometrically incorrect since the gizmo would need to be 1-dimensional as well in theory, but is better for usability than having a barely visible line as Gizmo only.



Two-dimensional check: This area would become active as long as the player would be above or below this 2-dimensional rectangle. The Y-position of the player would be irrelevant.



Three-dimensional check: This area would become active if the player is inside the 3D cube, all 3 axes X-Y-Z are taken into account here.

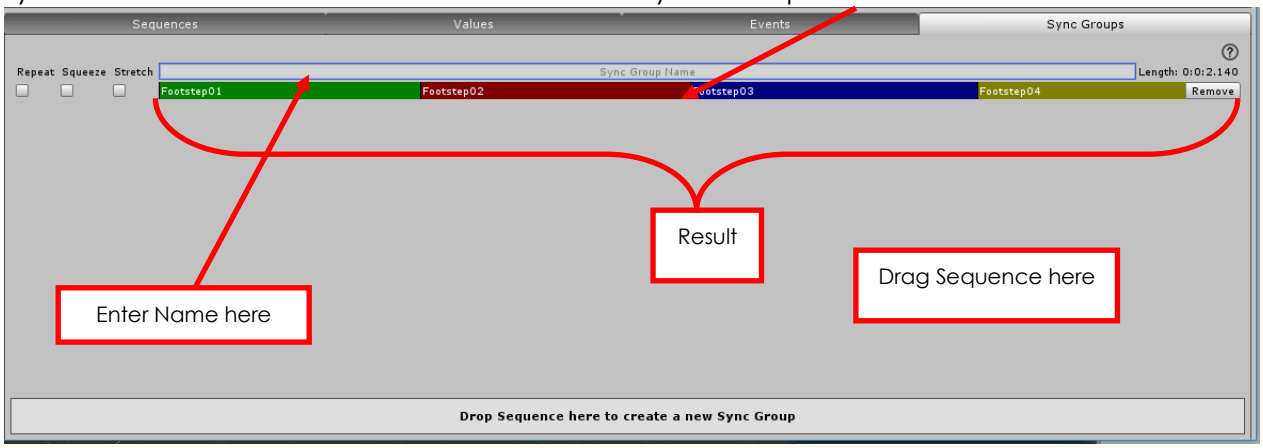
Gizmo

The Gizmo is the visual representation of your area in the scene view. You can decide whether it should always be displayed (or only if selected) and which color the Gizmo should use.
Note that you can also click on the Gizmo in the scene view to select the Audio Area, which can either be desirable or not, depending on what you are working on in your scene. It is therefore recommended to switch on "Always Display Gizmo" while working on multiple Audio Areas, and to switch that setting off otherwise.

Sync Groups

Sync Groups allow you to synchronize the playback of multiple Sequences so that all Sequences of a group can be played back with the same duration. Sync Groups are best understood & maintained from the Sync Group Tab in the Ambient Sound Manager Window as it gives you a visual representation of the Clips contained in the scene and how the different synchronization options will influence the result.

To set up a first Sync Group, drag and drop one of the Sequences you want to synchronize in the area at the bottom of the Sync Group Tab:

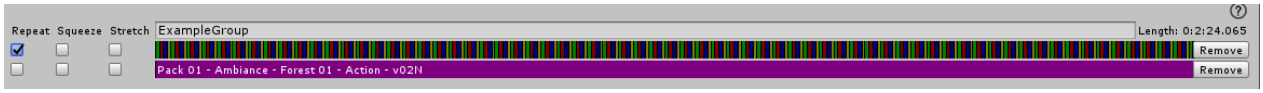


In the screenshot above, a Sequence containing four footstep clips was dragged into the window. Note that you can see the 4 clips being represented in the Sequence bar, also there is a total length for this Sequence shown on the right. You can give this new Sync Group a name in the text box above to identify it. You can drag another Sequence on this Sync Group to add it to that specific group.
This is what happens if we add another Sequence containing a longer musical piece to this Sync Group:

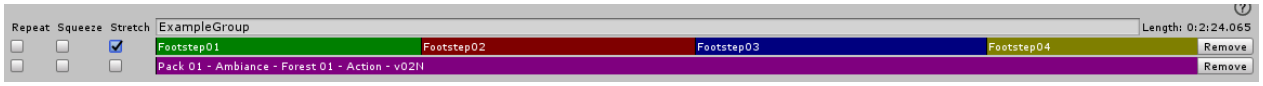
Note that due to the longer duration of the musical piece in the second Sequence, the total length of the Sync Group was increased. Also the display of the Footstep sounds has changed to represent the relative short duration compared to the other Sequence. When you play back the two Sequences at the same time now, e.g. by

adding them to an Audio Area, they will play back now with the synchronization options that you apply here.

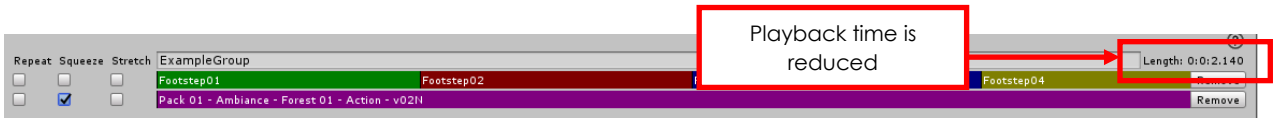
What would happen if you would play back this Sync Group now in this state? The four footstep sounds would be played back once and then never again until the end of the musical piece. This is exactly as it is represented in the preview in the Sync Group Tab.
By changing the synchronization options on the left side of each Sequence, you can change this behaviour to a more favorable synchronization:



Checking "Repeat" for the Footstep Sequence will repeat the footstep sounds over and over again to fill up the playback time for the Sync Group.

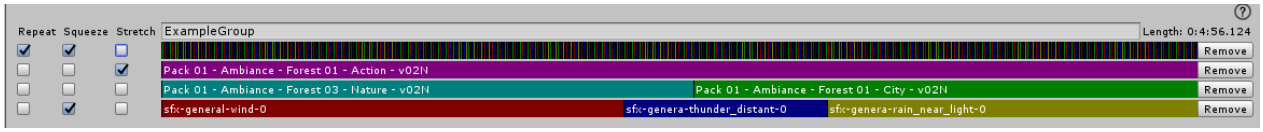


Checking "Stretch" for the Footstep Sequence will stretch the 4 footstep Clips over the duration of the musical piece by reducing the playback speed.



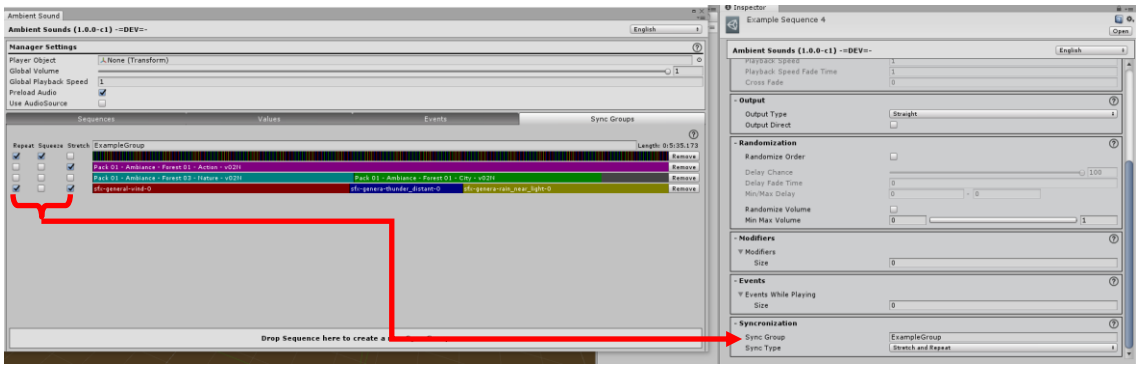
Checking „Squeeze" on the Musical Piece Sequence will squeeze the musical piece down to the duration of the Footstep Sequence. Note that the visual representation looks the same as with the "Stretch" option on the footsteps before, but the playback time has reduced. This indicates that it is now the musical piece that is being squeezed into the duration of the footsteps than the other way round.

By adjusting the checkboxes for each Sequence you can create the desired synchronization behaviour for the Sync Group. Note that you can use multiple options at the same time and different options on different Sequences. When options are conflicting, Ambient Sounds will find the best compromise for you.
If you are in doubt about how certain settings will play out, you can go by the visual representation of the Sequences and the total duration of the Sync Group.



Note that squeezing and stretching a Sequence will increase or decrease the pitch of that Sequence. This will barely be noticeable if you use stretch to compensate for a few seconds of difference in two 5-minute Sequences, but it will not be a feasible option for increasing the playtime of a Sequence by 2 minutes as the resulting playback will sound terrible due to the changed pitch.

You can assign Sequences to Sync Groups directly when editing a Sequence as well. At the bottom of the Sequence Editor you can enter the associated Sync Group name and select the desired synchronization options from a dropdown.



But this is more an option for experts as you don't have the same visual representation of the outcome of these settings as you would have in the Sync Groups Tab. Managing a complex Sync Group with many Sequences & different synchronization options this way is not recommended.

Ambient Sounds API

Ambient Sounds provides a very simple API to bring your gameplay in sync with the Ambient Sounds playback in no time. The AmbientSounds.**AmbienceManager** class provides multiple static functions that you can call anywhere from your project (Listed in alphabetical order):

- `void AmbientSounds.AmbienceManager.ActivateEvent(string EventName)`

  Activates an Event with the given name. These are the same Events you set up as a Requirement in Sequences / Modifiers.

- `void AmbientSounds.AmbienceManager.AddSequence(Sequence clipData)`

  Adds a Sequence to the "Global Sequences" list for Immediate Playback.

- `void AmbientSounds.AmbienceManager.ContinuePlayback()`

  Continues Playback (after PausePlayback() has been called).

- `void AmbientSounds.AmbienceManager.DeactivateEvent(string EventName)`

  Deactivates an Event with the given name. These are the same Events you set up as a Requirement in Sequences / Modifiers.

- `void AmbientSounds.AmbienceManager.Disable()`

  Disables all playback.

- `void AmbientSounds.AmbienceManager.Enable()`

  (Re-)enables all playback. (after using Disable())

- `string[] AmbientSounds.AmbienceManager.GetEvents()`

  Gets a string array of all active Event Names from Ambient Sounds.

- `float AmbientSounds.AmbienceManager.GetValue(string Name, bool AddIfMissing)`

  Gets the current Value in Ambient Sounds for this Name. These are the same Values you set up as a Requirement Value Checks in Sequences / Modifiers. If AddIsMissing is set to true(default), this value will be visible in Ambient Sounds

after the query.

- `Dictionary<string, float> AmbientSounds.AmbienceManager.GetValues()`

  Gets a Dictionary with all registered Values in Ambient Sound. These are the same Values you set up as a Requirement Value Checks in Sequences / Modifiers. Note that this list will only contain Values that have received at least one update before.

- `bool AmbientSounds.AmbienceManager.IsPlaying(AudioClip clip)`

  Returns true if the Audio Clip is played by the Play() function.

- `bool AmbientSounds.AmbienceManager.IsSequenceActive(Sequence seq)`

  Returns true if the Sequence is currently active and will be played if its Requirements are met.

- `void AmbientSounds.AmbienceManager.PausePlayback()`

  Pauses all Audio playback (can be resumed with ContinuePlayback()).

- `void AmbientSounds.AmbienceManager.Play(AudioClip clip, float volume)`

  Plays back the Audio clip immediately with the given volume.

- `void AmbientSounds.AmbienceManager.PreloadAudioClip(AudioClip clip)`

  Preloads an Audio Clip for playback. Not required for using other playback functions, but can prevent loading hiccups, hitches etc. from loading the clip on the fly during gameplay.

- `void AmbientSounds.AmbienceManager.PreloadSequence(Sequence seq)`

  Preloads an entire Sequence for playback. Not required for using other playback functions, but can prevent loading hiccups, hitches etc. from loading the clip on the fly during gameplay.

- `void AmbientSounds.AmbienceManager.RemoveSequence(Sequence clipData)`

  Removes a Sequence from the "Global Sequence" playback list.

- `void AmbientSounds.AmbienceManager.RemoveValue(string Name)`

  Removes any Values in Ambient Sounds for this Name. These are the same

Values you set up as a Requirement Value Check in Sequences / Modifiers. Note that this will only clear the current Value for this name, but will leave the Value checks in the Requirements intact.

- `void AmbientSounds.AmbienceManager.SetValue(string Name, float Value)`

  Sets a Value in Ambient Sound for this Name. These are the same Values you set up as a Requirement Value Check in Sequences / Modifiers.

- `bool AmbientSounds.AmbienceManager.WasSequenceAdded(Sequence seq)`

  Returns true when Sequence seq was added by AddSequence() and can be removed with RemoveSequence().

If you are an inventive programmer you might notice that the Ambience Manager class offers even more functions and Events for you to use in building interactions with your project and Ambient Sounds. However usually only the functions above should be required to integrate Ambient Sounds, use anything else at your own risk.

The Ambient Sounds **Sequence** class has events that allow you to detect when clips are started and stopped. Have a look at SequenceControlledLight.cs for an example on how to us them.

- `PlayEvent AmbientSounds.Sequence.m_OnPlayClip`

  Will be called when playback is started on a clip within the sequence.

- `PlayEvent AmbientSounds.Sequence.m_OnStopClip`

  Will be called when playback is stopped on a clip within the sequence.

All Ambient Sounds components and assets can be scripted like regular Unity assets, e.g. it should be no problem to automate the creation of Sequences, Audio Areas, etc. if you require it for your project.