Write a short tutorial on how to use sounds in Unity

* Fire and Forget Sounds
* Looping Sounds (Start & Stop)
* Mixer
  + Music (Separate Volumes)
  + FX

Unity Sound Tutorial

# Intro

Sounds give a player a sense of atmosphere and adds realism to your game. A shotgun firing, bullets ricocheting, background music, enemies speaking, wind blowing through the trees… without sound in a game, it would be bland.

This tutorial is intended to teach you about using Sounds in Unity. We will be exploring three sound-based topics, Audio Sources, Fire and Forget sounds, Looping sounds, and the Audio Mixer.

# Audio Sources

## What are Audio Sources

Audio Sources are components added onto a Unity game object. We need an object to hold our Audio Source so create a cube by clicking on the Create tab on the top of your Unity window. Then select 3D Objects, and then cube. Click on the cube if it is not already selected, and then click “Add Component” in the inspector window, you should be able to find the Audio Source component.

Audio Sources are components that let you play sounds in a 3D environment from a game object. Audio Sources need an Audio Listener in order to be heard, however the Audio Listener is a component that needs to be attached to your main camera.

Audio Sources are usually only able to play one sound, unless you use the PlayOneShot method associated with one.

# Fire and Forget

## What are Fire and Forget Sounds

Fire and Forget sounds are sounds that are played once and then unloaded. If you were to play a sound from a machine gun, you’d want the sound to play once and then stop for each time the gun is fired. However, audio sources normally only allow one sound to be played at a time… This is where the PlayOneShot method comes into play.

## PlayOneShot

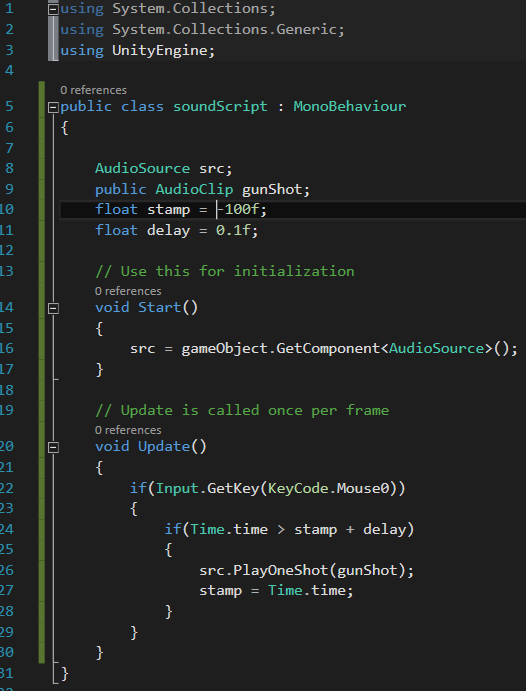
With PlayOneShot you are able to play any audio clip that you have a reference to **ONCE.** However, you are able use PlayOneShot as many times as you’d like.

Going back to the machine gun example from earlier; A machine gun should play a gunshot sound every time a bullet is fired, however these bullets are fired rapidly so you would need to find a way to play the sound multiple times. Using PlayOneShot will make this simple, you’d simply use the PlayOneShot method every time a bullet is fired and you will hear the gunshot sound play for every bullet.

The syntax for the PlayOneShot method is as follows (square brackets indicate optional fields):

**AudioSource**.PlayOneShot(**AudioClip** clip [, **float** volumeScale])

In the screenshot below you will see me use this method in a simple machine gun script.



Every time the player holds down the left mouse button (Mouse0), the gunshot sound will be played every 0.1 seconds. The sound for each gunShot will be played over the previous gunShot sounds, because each sound is made to play once and then destroy itself **once the sound ends.** So if the sound hasn’t ended yet, it will continue until it has.

# Sound Looping

## What is a Sound Loop

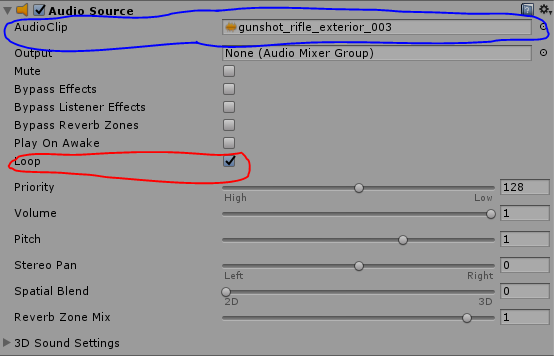
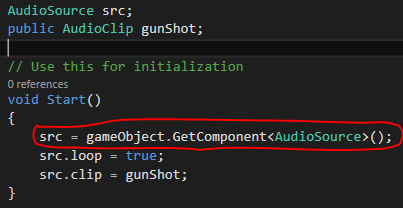
A sound loop (or looping a sound) is when a sound **stops** when it ends, and then **replays** again. Sound looping is only available for audio clips that are inside of an AudioSource, therefore it’s not possible to play multiple looping sounds through one AudioSource component. It is also not possible to loop a sound using PlayOneShot since the sound is unloaded after it has finished playing.

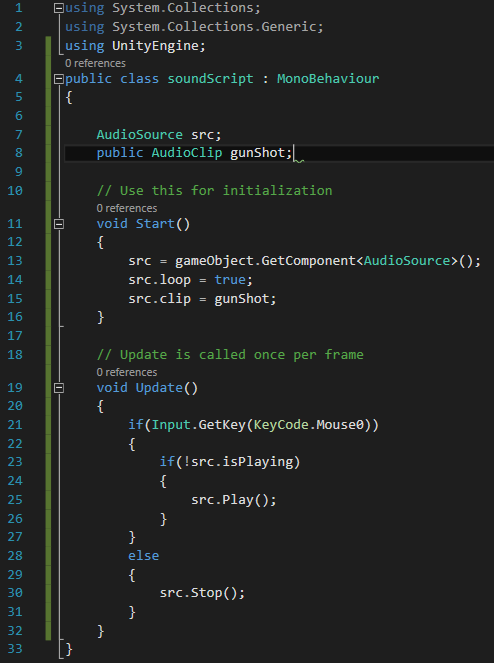
## How to Loop Audio

Looping is done with the loop property of an AudioSource.

You can access this property via scripting by getting a reference to your AudioSource component (named **src** in this example) via GetComponent<AudioSource>() and then using this reference to access **AudioSource.loop.**

You can also access this property in the inspector window of unity by selecting our cube and checking  
the “Loop” checkbox on our Audio Source component.

See the red outline below for examples of both.

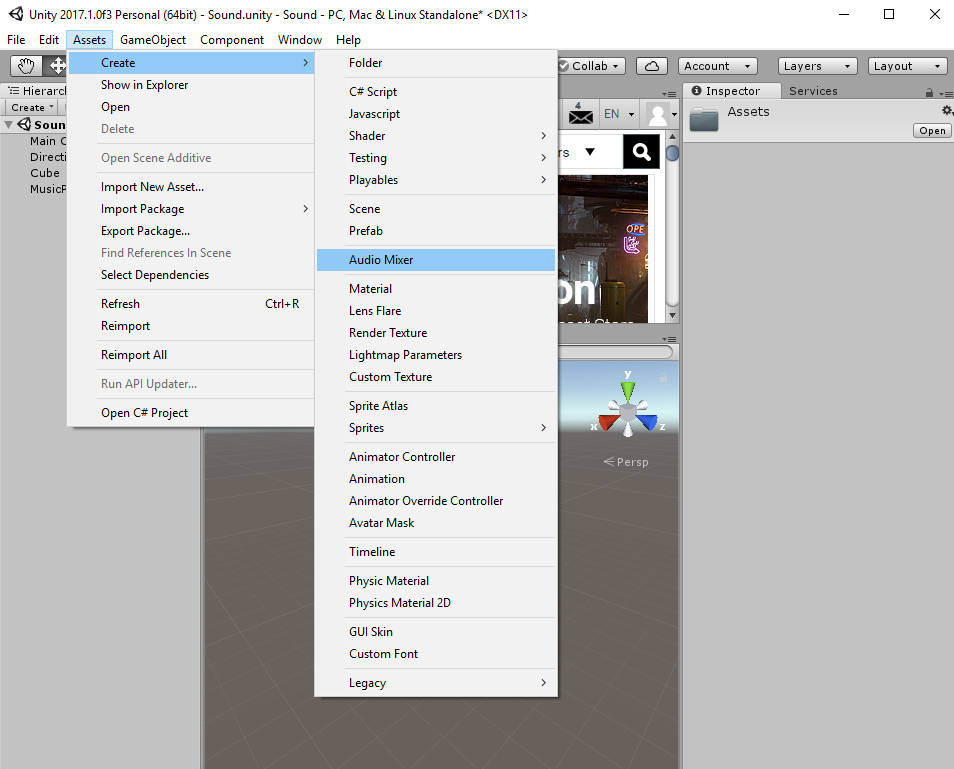
However, in order for this looping to take effect we need to also know how to play the sound. Playing the sound is as simple as ensuring that you have your sound loaded into your audio source (Blue Outline) and calling the Play method of your audio source (**src**).

In our machine gun example instead of using a timer to play the gunshot sound every 0.1 seconds, we could let the sound loop itself. However, the problem with this is that the sound does not loop as quickly as your machine gun is firing, but this is no problem for slower weaponry as long as you match your sound with your firing.

See the code to the right for an example on how we would make our machine gun play the gunshot sound in a loop so that it restarts itself after it has finished playing.

# Audio Mixer

## What is an Audio Mixer

An Audio Mixer is an asset that is created to control and mix different Audio Sources together, and to apply effects to them. This is excellent for when you have a game with both background music, sound effects, and dialogue. Depending on your preference you may want to have the dialogue louder than the music, and the sound effects louder than the music but quieter than the dialogue. An Audio Mixer will accomplish this for you.

## How to use an Audio Mixer

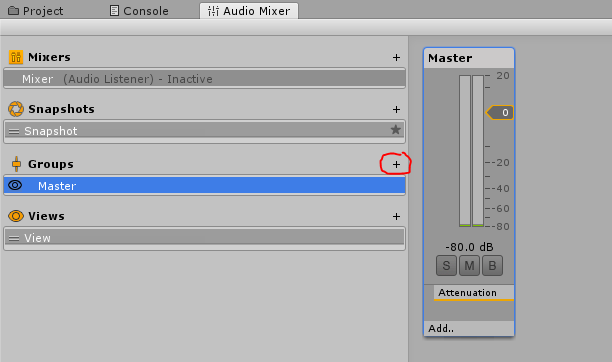
First we need to create an Audio Mixer asset. You can create one by going into the assets menu on the top of your Unity window, and then going into the Create dropdown menu, and selecting Audio Mixer. You can also create an Audio Mixer by right clicking in your project files area, and then once again going into the Create dropdown menu, and then selecting Audio Mixer.

Now that you have an Audio Mixer, double click on it in your project files area to open it. This will open your Mixer window and give you plenty of options associated with a mixer!

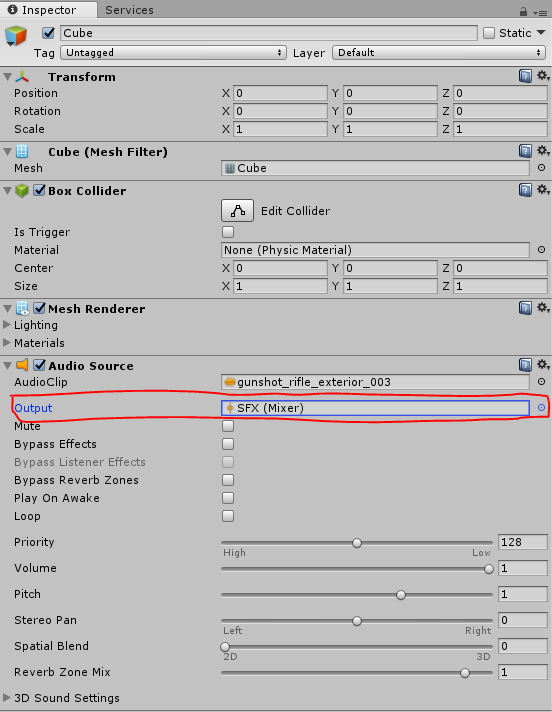
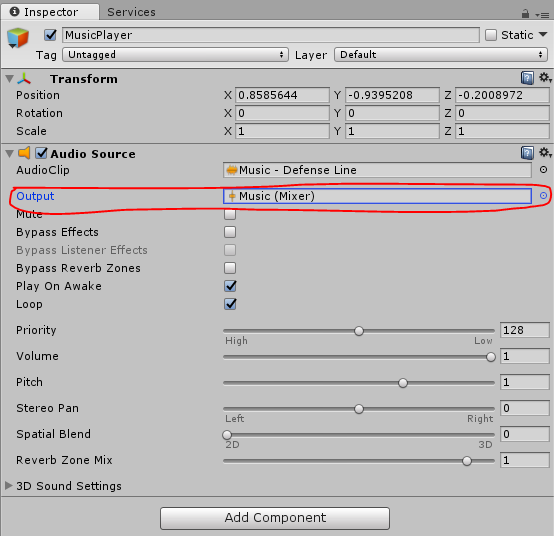
### Mixer Groups

Our mixer has a **Mixer Group** called Master, this group generally controls the audio for every single sound in our game. This is normal for all master groups, however this makes it so we are unable to add effects or control specific audio sources. We can change this by adding sub-groups to our mixer.

Adding a second Mixer Group as a child of the Master group will allow us to control that group individually, while still letting the Master Group control all of the audio sources in our game. To add a Mixer Group as a child of the master group, highlight the master group and click on the + icon to the top right of the group section. Name this group Music. Repeat the process a second time and name that group SFX. See the result on the right-most screenshot.



These two groups are now made, but we’re missing something… we don’t have a game object in our scene that has any music attached to it yet! Find any song you’d like and import the file into your Unity by dragging it into your project files area. Next, create a game object in your scene the same way you created the 3D cube earlier. This can be either a visible game object like another cube or an empty game object. Once the object is created, add a Audio Source component to it just like we did earlier, and assign the music file as the audio clip for your Audio Source.

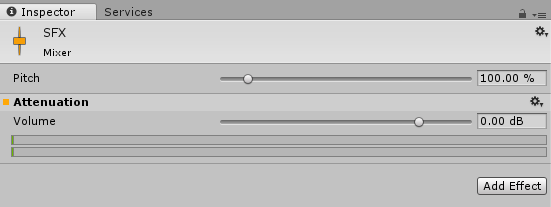
Below the field where you put your music file into your Audio Source, you’ll see a Output field. This is where you assign the Audio Source to a Mixer Group. Go ahead and click on it to assign this Audio Source to the Music group that we created earlier. Next, assign the Audio Source for our machine gun cube to the SFX group that we created earlier.

Now if you play your scene while your mixer window is open, you’ll see that both the Master and Music groups are playing audio. If you hold left click, the code we wrote earlier will run the machine gun sounds and you’ll see that the SFX group we created starts to light up. This proves that both of our Audio Sources are now independent. Now comes the fun part.

## Sound Effects

Each group we’ve created has an effect attached to it called “Attentuation”. This effect is used to control the volume of the entire group. (Red Outline) We can add extra effects to each group by clicking on their respective Add… buttons. (Blue Outline) Each effect you add to a group can be seen by selecting the group and looking at the inspector window. You can also change the way these effects affect your sound from the inspector. Have fun playing around with the different effects to get a feel for them.





While your scene is playing, you will see an option in your Audio Mixer area pop up with a red recording symbol next to it called “Edit in Play Mode”. (Red Outline) Selecting this option lets you modify everything about the Mixer and it’s audio while hearing the results live from your game. Unlike most things in Unity, stopping the scene after making changes will **save your changes** to the mixer.

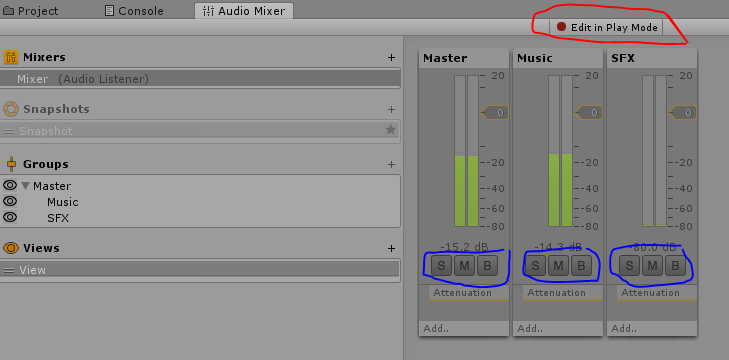
### Temporary Modifications

You are also able to modify what sounds you hear **for creation purposes only**. These next three buttons should never be set active by a script or left on during a build of your scene. These three buttons are labelled as S, M, and B.

The S button stands for “Solo”. Selecting this button will make it so that this group will be playing audio while the rest that are not on Solo mode will not be playing any audio.

The M button stands for “Mute”. Selecting this button will make it so that this group will **NOT** be playing audio while the rest that are not on Mute mode **WILL** be playing audio. It is essentially the opposite of Solo mode.

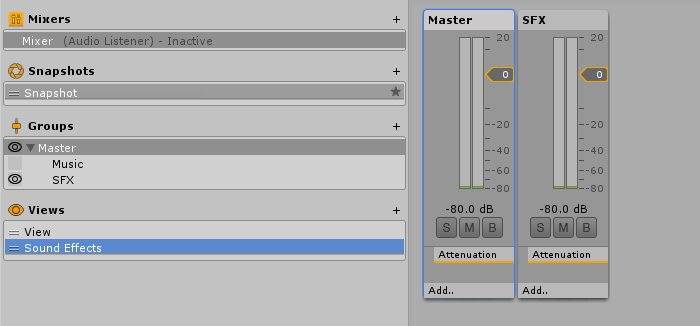
The B button stands for “Bypass”. Selecting this button will make it so that all the effects you have on that group will be “bypassed” (ignored). This lets you compare the raw version of the sounds to their modified versions.



### Views

The views area of the audio mixer is used to save states where we hide or unhide some of our Mixer Groups. Each Mixer Group in the “Groups” section of the mixer has an eye next to it, if you click on one of those eyes, the group associated with it will be hidden. This does not affect the children of the group you hide.

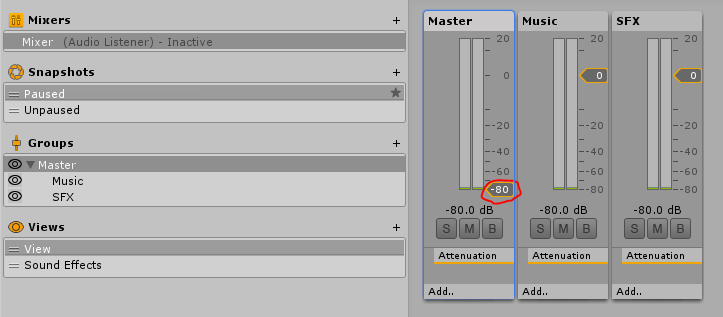
Make sure that all the eyes are active and click on the plus icon in the “Views” section of our mixer. This creates a copy of our default view, “View”. Rename it to “Sound Effects”. With our new Sound Effects view selected, you are now free to hide any of the Mixer Groups and it will be saved to that specific view. Since our view is named “Sound Effects”, we’re going to hide the Music group by clicking on the eye next to it. Now when you select the default view “View” from our Views list, it will show all of our groups, and when you select the view “Sound Effects” it will hide our Music Mixer Group.



### Snapshots

Lastly we’re going to look at Snapshots. Snapshots allow you to save the state that your audio mixer is in so that you can go back to it anytime you want without having to play with all the settings again. This is commonly used to mute all sounds while your game is paused. Let’s create a “Pause” and “Unpaused” snapshot of our Mixer.

In the Snapshots section of the mixer, click on the + on the top right and you will create a copy of the default snapshot. Rename this to “Unpaused”, also rename the original snapshot to “Paused”. Select the Paused snapshot we just created and lower the volume of our **Master** to -80 by using the grey slider handle to the right of our master group. (Red Outline)



Now that our master volume is set to -80, none of the sounds in this state will be heard even if they are still going to be played. Don’t worry though, we haven’t modified our “Unpaused” snapshot so we haven’t lost the original setting of our volume. Don’t forget that when we created our Unpaused snapshot our volume was already at it’s original setting, and that Unpaused is just a copy of it. If you click on our Unpaused snapshot on the left hand side you will see that the Master volume goes back to it’s original value. This is very valuable, because it is now possible for a script to “Pause” and “Unpause” our sounds just by switching to a different snapshot.

While a snapshot may not seem like a big deal right now, that’s simply because we’ve only played with the volume of our master. If we decided to add a reverb effect to our SFX group we could have different reverb levels for each snapshot… although in this case that would be pointless since we wouldn’t be able to hear the reverb in our Paused snapshot.

# Summary

As a summary, we took a good look at how sound can be used in the Unity game engine.

We now know how valuable an Audio Source component is, and that a scene **needs** and Audio Listener for our Audio Sources to be heard. We also now know that we can play sounds once using the PlayOneShot method, or on a loop using the loop property of an audio source. Mixers are also a great way to add interesting effects to our sounds, and to separate our sounds into groups and different states (snapshots).

Overall, sounds are very important for game development because they help keep the player interested by stimulating one extra sense, hearing.