

# Simple Audio Manager



## Getting Started

The Simple Audio Manager is an easy way to set up audio in your Unity game project. It allows you to quickly create GameObjects that hold audio sources and can be called anywhere in your game project code.

The package containing the Simple Audio Manager has a folder with two subfolders, one folder contains the manager prefab (the prefab you will be adding to your scene in order to add audio sources). The other folder contains the scripts that make the audio manager function.

## AudioManager

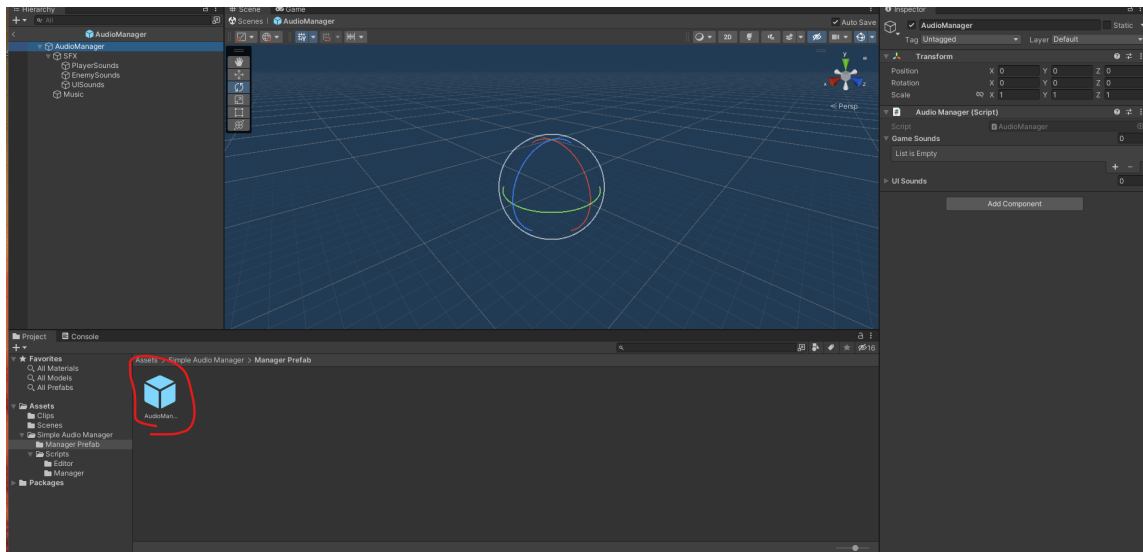
The AudioManager Object has a script on it called AudioManagerScript, which is outfitted with two lists, GameSounds and UISounds. Player and EnemySounds are added to the GameSounds list, and UISounds are added to the UISounds list when created.

## Creating A New Sound Effect

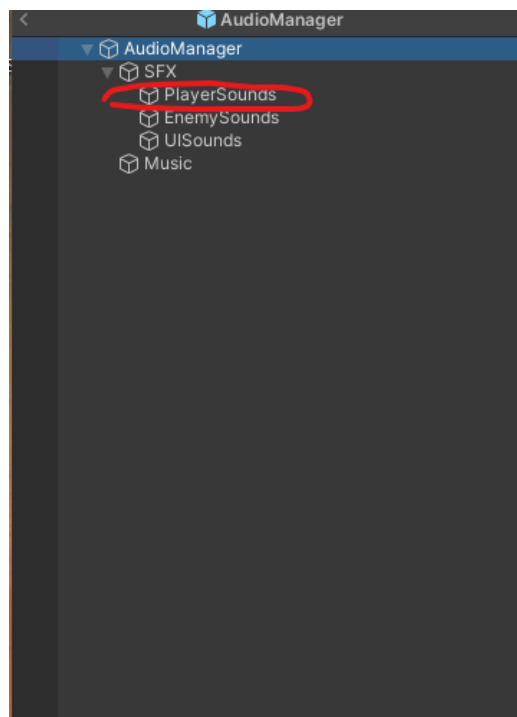
Creating a new sound is as simple as following these steps:

- Open the AudioManager prefab object

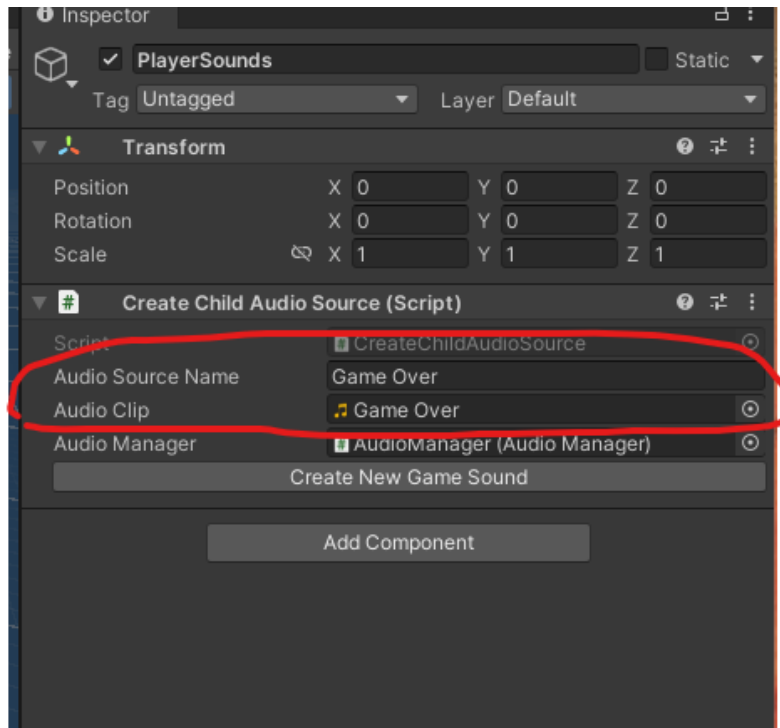
TKG Studios  
Simple Audio Manager - Documentation  
Version 1.0



- Navigate to the sound type that you wish to create. (For this example, we will create a *Player Sound* effect, which is a “Game Sound”)

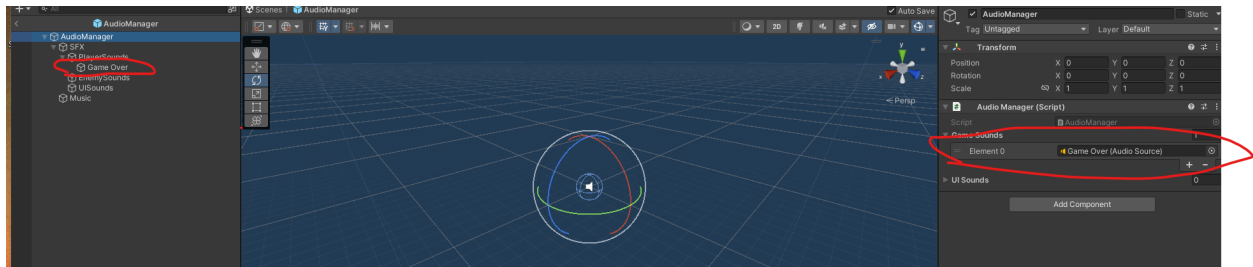


- Click the child object “PlayerSounds”
- In the Inspector, type in the Audio Source Name, and drag and drop the audio clip you wish to use



- Click “Create New Game Sound”

The Game Sound will be added as a child object of “Player Sounds”, and it will also be added to the list of Game Sounds in your Audio Manager



## Creating A New Music Track

The instructions for creating a music track are largely the same as creating a sound effect.

- Navigate to the Music child object in the AudioManager Prefab
- Supply a Track Name and an Audio Clip
- Click “Add New Track”
- The track will be added to the Track List on the Level Music script

## Calling Your Audio In Your Project

Calling sound effects and music in your project is very simple. The first step is to make sure there is a function that will play the clip present in either your Audio Manager script or your Level Music script. There are example functions available in both scripts to look at, so please open them and take a look at how to format your functions.

```
.....//Game Sound Functions Here

0 references
.....public void playGameSound0()
.....{
.....    gameSounds[0].Stop();
.....    gameSounds[0].Play();
.....}

.....//UI Sound Functions Here

0 references
.....public void playUISound0()
.....{
.....    UISounds[0].Stop();
.....    UISounds[0].Play();
.....}
```

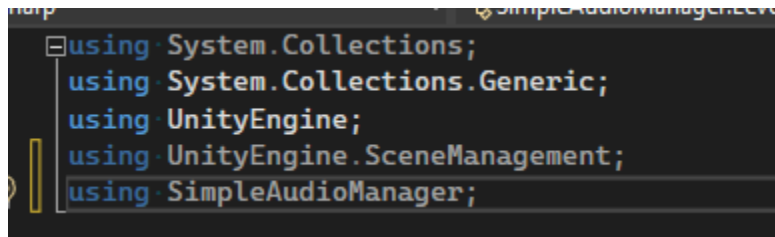
Example Functions for SFX

```
0 references
.....public void playTrack1()
.....{
.....    if (playMusic)
.....    {
.....        trackList[0].Stop();
.....        trackList[0].Play();
.....    }
.....}
```

Example Function for Music

- Ensure that the Audio Manager is present in the scene that you wish to use audio

- In the script that you want to call your audio function, make sure to add the `using SimpleAudioManager` directive to the top of your script

A screenshot of a code editor showing a C# script. The script includes the following 'using' directives: `using System.Collections;`, `using System.Collections.Generic;`, `using UnityEngine;`, `using UnityEngine.SceneManagement;`, and `using SimpleAudioManager;`. The code is highlighted in a dark-themed editor with yellow and blue syntax highlighting. A yellow cursor is visible on the line containing `using SimpleAudioManager;`.

```
using System.Collections;  
using System.Collections.Generic;  
using UnityEngine;  
using UnityEngine.SceneManagement;  
using SimpleAudioManager;
```

- From there, it's as simple as writing a function call wherever you need it.

As an example, to call the first gameSound in the gameSound list, this would be your call:  
`AudioManager.instance.playGameSound0()`

To call the first music track in your track list, this would be your call:  
`LevelManager.instance.playTrack1()`