# ON THE BLOCK GAMING

BUILDING YOUR VISION BLOCK BY BLOCK

# OTBG Plugins

## Audio System

The Audio system is designed using a Scriptable Object approach and has a few main parts to make it functional. We'll start with the main prefab, the AudioManager Prefab.

## Audio Manager

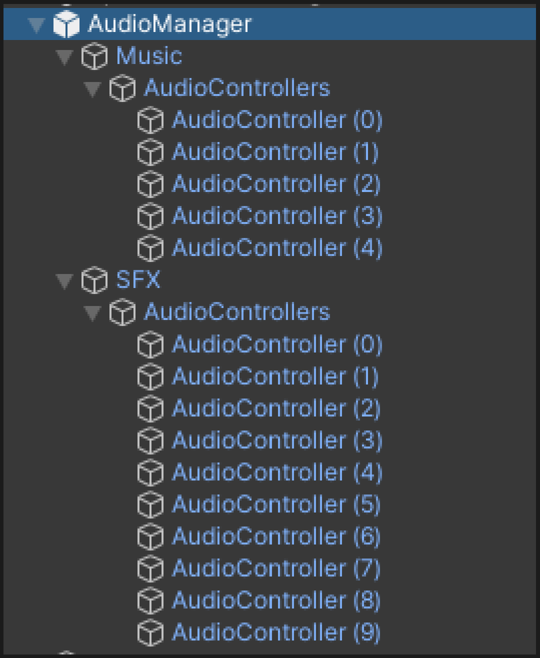
The AudioManager prefab holds all the logic for playing both music and sound effects. It's separated into two sections, SFX and BGM. Each one containing pre-setup AudioSources ready to be used when triggered.

The Audio Manager is a Singleton, which means it'll persist through all of your scenes.

### Setup

To setup the audio system. Start by dragging and dropping the AudioManager into the first scene of your game/application. This will allow it to persist through all scenes, don't make it a child of an object that isn't a singleton otherwise it won't be persistent.

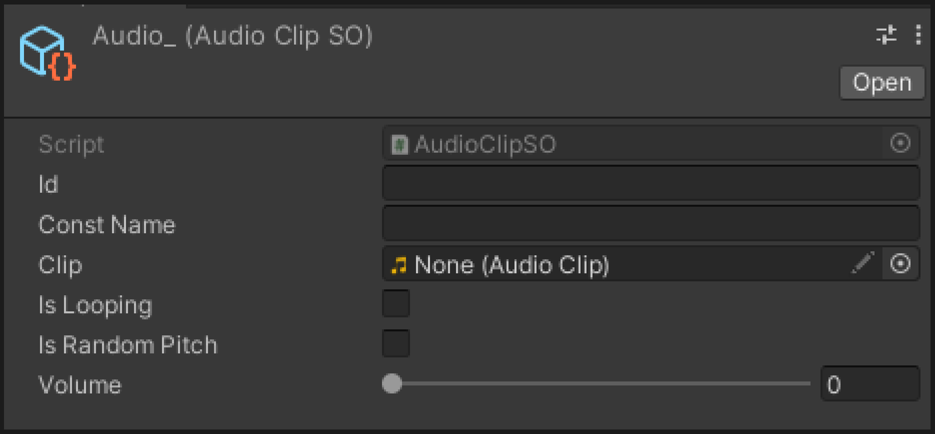
Once that is in, no more setup required for the Audio Manager. Next we'll move onto how to create new Music and SFX tracks.



```
AudioManager
  Music
    AudioControllers
      AudioController (0)
      AudioController (1)
      AudioController (2)
      AudioController (3)
      AudioController (4)
  SFX
    AudioControllers
      AudioController (0)
      AudioController (1)
      AudioController (2)
      AudioController (3)
      AudioController (4)
      AudioController (5)
      AudioController (6)
      AudioController (7)
      AudioController (8)
      AudioController (9)
```

# Audio Scriptable Objects

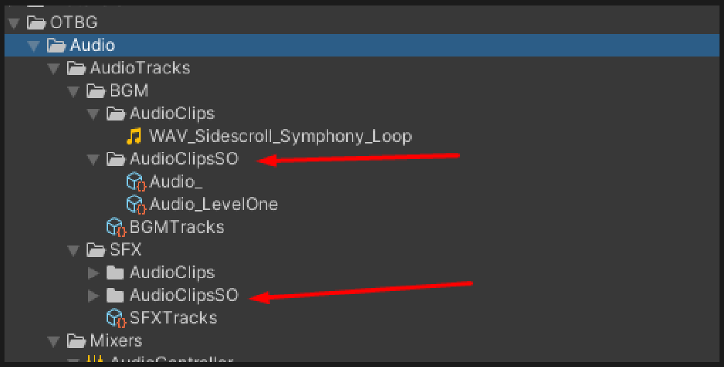
In order to play audio clips, you need to setup an AudioFile from Assets -> Create -> OTBG -> Audio -> Create Audio File
This will create a blank AudioFile Scriptable Object for you to use.



Audio_ (Audio Clip SO)

Script: AudioClipSO
Id:
Const Name:
Clip: None (Audio Clip)
Is Looping: □
Is Random Pitch: □
Volume: 0

Make sure to create it in the folder OTBG -> Audio -> AudioTracks -> BGM/SFX -> AudioClipsSO. This folder is setup with the system in order to generate Consts based on the Audio Files information.



```
▼ OTBG
  ▼ Audio
    ▼ AudioTracks
      ▼ BGM
        ▼ AudioClips
            WAV_Sidescroll_Symphony_Loop
        ▼ AudioClipsSO
            Audio_
            Audio_LevelOne
          BGMTracks
      ▼ SFX
        ▶ AudioClips
        ▶ AudioClipsSO
          SFXTracks
    ▼ Mixers
      ▼ AudioController
```

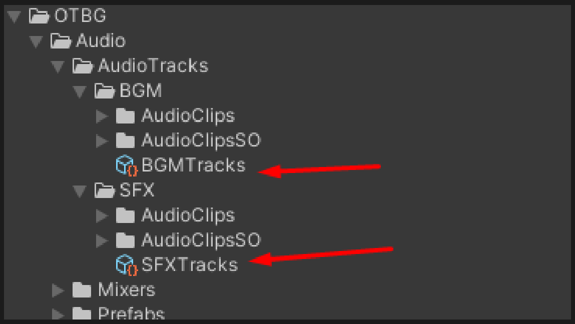
Once you have created an Audio File, you will have the ability to setup the different parameters.

- **ID:** This is the id of the audio clip. It's the key used to call this audio track from the AudioManager.
- **Const Name:** This is the Variable Name that the const generator will generate for this AudioFile
- **Clip:** This is the Audio clip that you will play when playing this AudioFile
- **IsLooping:** Will the audio track loop?
- **IsRandomPitch:** Is the pitch going to randomise, and once toggled, how much?
- **Volume:** What's the volume of this audio track?

Once these are all set, the next thing is to add it to the collection of SFX or BGM depending on what you created.

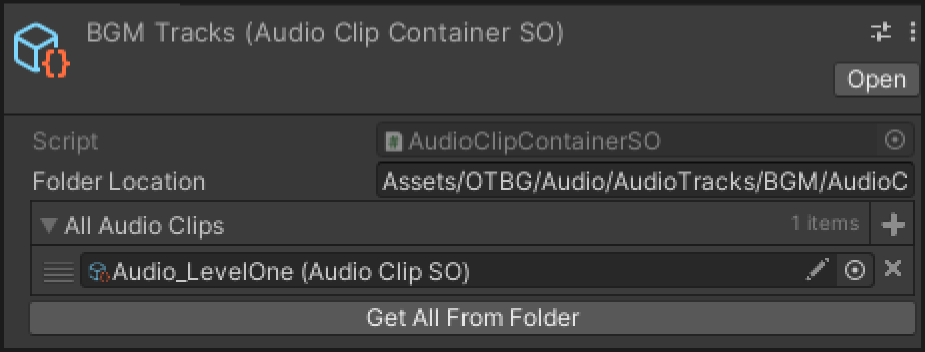
# The Audio Collections

Once we have created our AudioFile, we can move onto the AudioContainer. By default the two required are already in the folders SFX and BGM.



```
OTBG
  Audio
    AudioTracks
      BGM
        AudioClips
        AudioClipsSO
        BGMTracks
      SFX
        AudioClips
        AudioClipsSO
        SFXTracks
    Mixers
    Prefabs
```

If they don't exist. You can create them in the OTBG -> Audio tool bar.



BGM Tracks (Audio Clip Container SO)

Script: AudioClipContainerSO

Folder Location: Assets/OTBG/Audio/AudioTracks/BGM/AudioC

All Audio Clips — 1 items

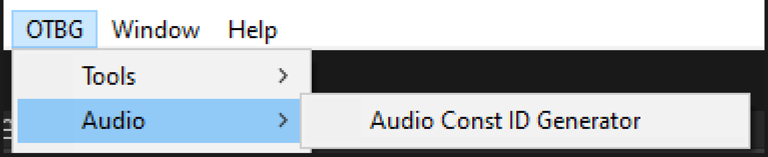
Audio_LevelOne (Audio Clip SO)

The purpose of the container is to track all the different audio clips as this will be relevant for generating all of the consts for the audio files.

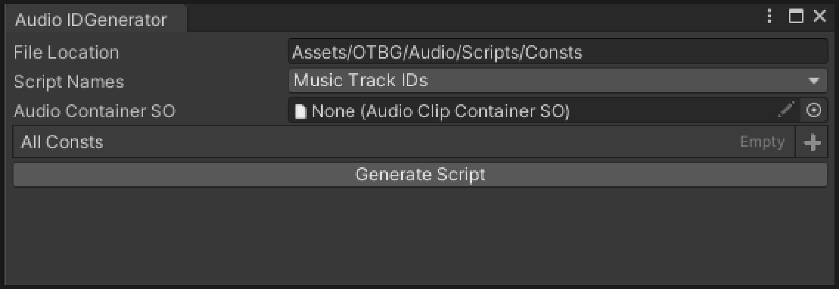
You can either manually drag them in or set the Folder Location and "**Get All From Folder**"

# Generating the Consts Files

So, once all of this is setup. You have your Audio Files, Audio Clip Containers and have assigned everything.
We can move onto generating the const files with the relevant information.
In the Unity Toolbar along the top, you'll see OTBG. Inside that you will have Audio then Audio Const ID Generator.



Once you open this, you'll be welcomed with a new window as follows.

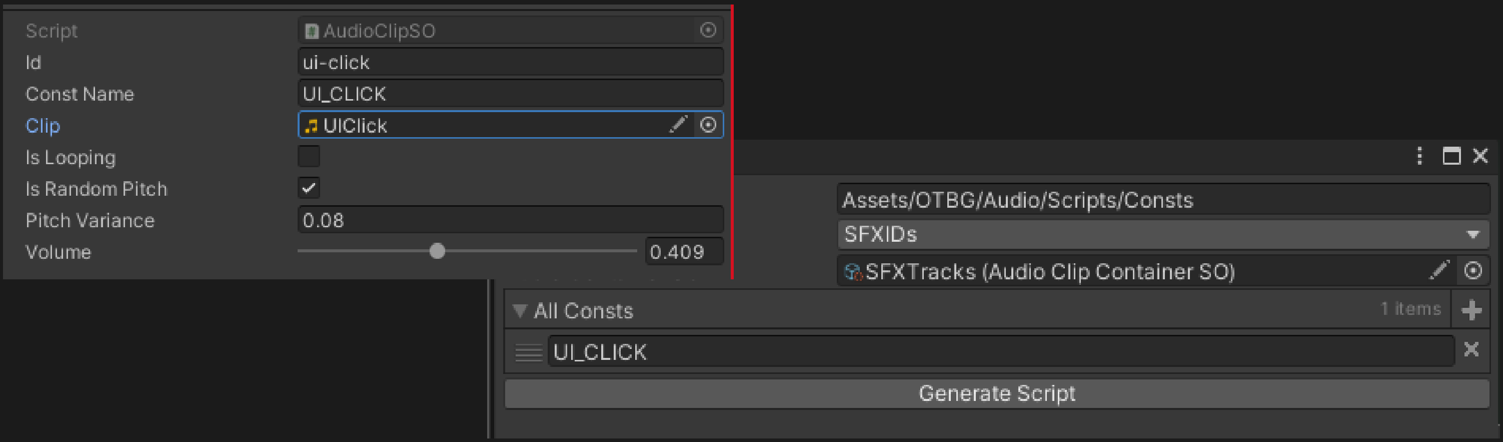


File Location: Assets/OTBG/Audio/Scripts/Consts
Script Names: Music Track IDs
Audio Container SO: None (Audio Clip Container SO)
All Consts: Empty

This window has a few elements to it.

- File Location: Where will the Const Files exist when you generate them
- Script Names: What's the name of the Const file you're trying to generate?
- Audio Container SO: Which Audio Container are you generating consts from and into what file?
- All Consts (List): This will visualise all of the consts that will be created when you input the Audio Container SO
- Generate Script: Create the const script in the folderpath provided.

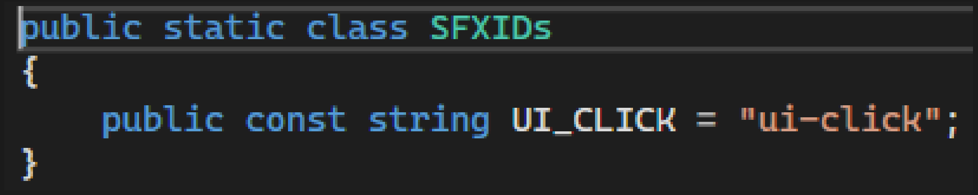
Once you've done this. You should find that the file has now got the Consts from your AudioFiles. In this example I will have an SFX AudioFile with the Const name "UI_CLICK" and it will have a key of "ui-click".



Script: AudioClipSO
Id: ui-click
Const Name: UI_CLICK
Clip: UIClick
Is Looping: ☐
Is Random Pitch: ☑
Pitch Variance: 0.08
Volume: 0.409

Assets/OTBG/Audio/Scripts/Consts
SFXIDs
SFXTracks (Audio Clip Container SO)
All Consts: 1 items
UI_CLICK

Now inside my window, it tells me the const. I’ve selected ScriptName to be SFXIDs and the Audio Container for my SFX. Once I click Generate. The const file would have updated.



```
public static class SFXIDs
{
    public const string UI_CLICK = "ui-click";
}
```

## Calling an Audio Track for your game.

Once everything is setup. You have your const file. It’s time to use it.
It’s fairly simple to play an Audio Track.
You need to use the AudioManager class. For example, if I want to play the sound effect “UI_CLICK” then I would do the following.

```
public void PlaySFX()
{
    AudioManager.Instance.PlaySFX(SFXIDs.UI_CLICK, true);
}
```

This would play the SFX from the AudioManager. The bool parameter is deciding whether or not it’s a oneshot clip or not. For instance, UI clicks, hit sounds, gun shot sounds. These would be one shot clips. But let’s say you had a sound effect for ambient background noise, then this could be played without being a oneshot.

## Useful Functions

```
//Plays a music track
AudioManager.Instance.PlayMusicTrack(string key)
//Stops a music track if playing
AudioManager.Instance.StopMusicTrack(string key)
//Stops all music tracks. Useful for moving between scenes.
AudioManager.Instance.StopAllMusicTracks
```