

# Implementation of Audio Module in Unity with FMOD

[FMOD Version 2.02.04 (Unity Verified)]

- **Type of sounds Implemented**

- **One-shot sounds**
- **Looping sounds**
- **Event based sounds**

- **Implementation:**

The main audio play logic is written in **AudioManagerSO.cs**. Every MonoBehaviour will shoot event through RaiseEvent() function and definition of that event is done in AudioManagerSO.cs. RaiseEvent() will pass FMOD event and gameobject from which event is shoot.

There is another class called AudioHelper.cs, which will help custom monobehaviors to implement or access the functionality of FMOD API. Below is the examples on how you can implement sounds in different scenarios.

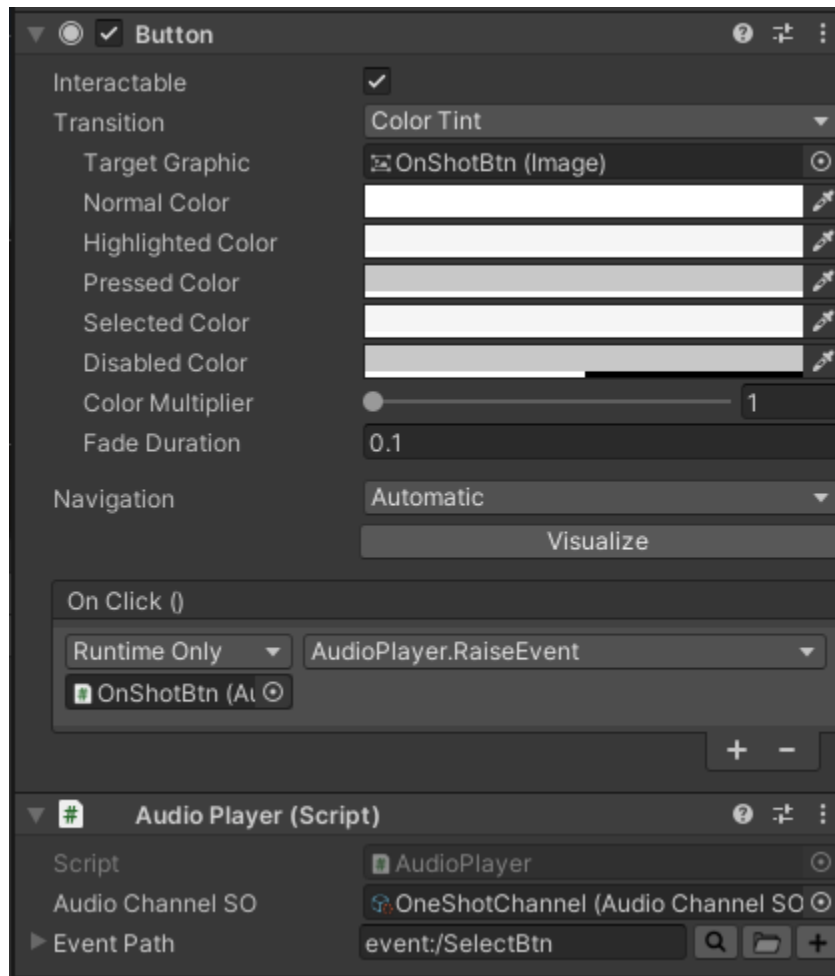
- **OneShot Audio Implementation:**

\* AudioPlayer is called from object which has particular sound.

```

1 reference
void PlayAudio() {
    gameObject.GetComponent<AudioPlayer>().RaiseEvent();
}

```



\* In this image AudioPlayer.cs script is called from component Button on gameobject.

As you can clearly see that AudioPlayer.cs has two properties, one is channel type and other is event path of required audio.

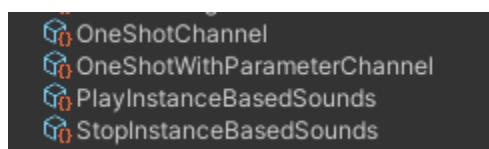
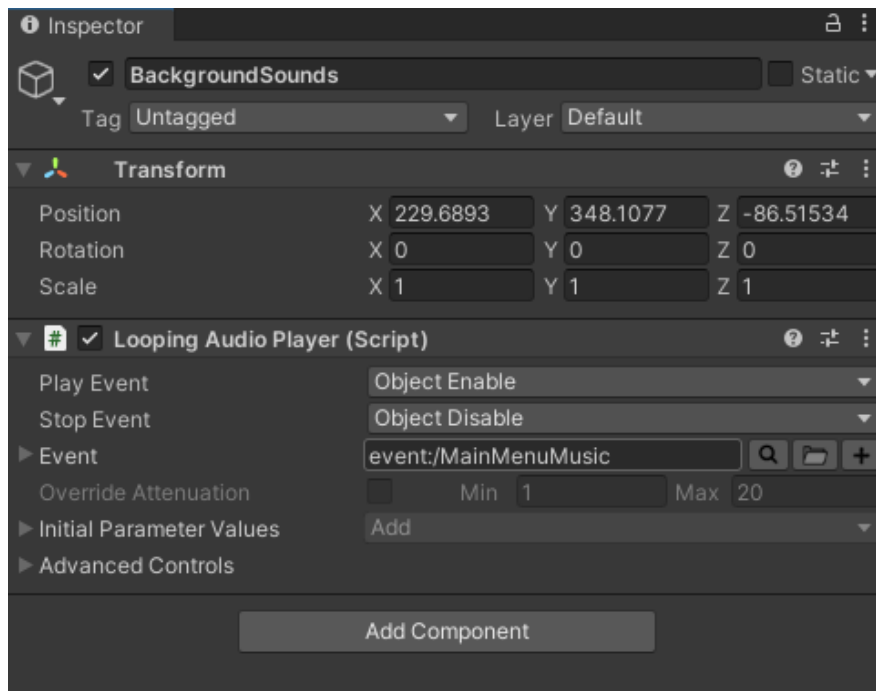


Image above shows the type of sounds we are playing from `AudioPlayer.cs`, i.e One-shot sound with or without parameter and instance based sounds.

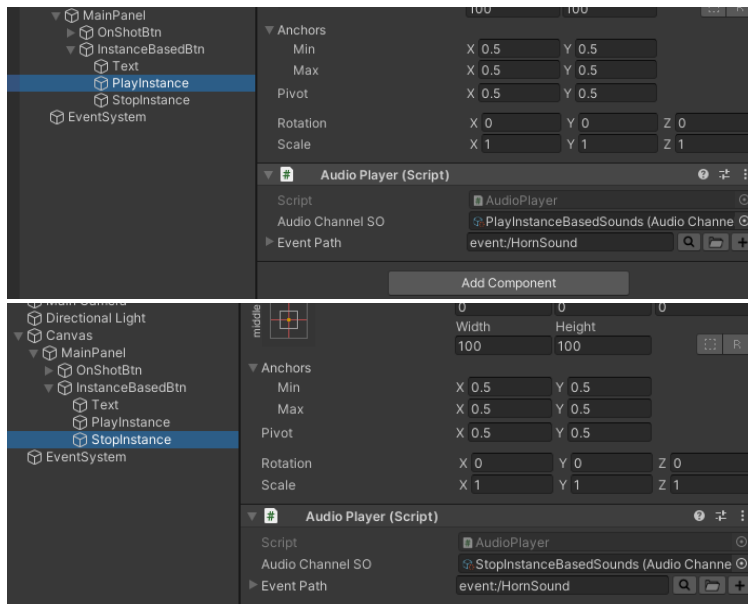
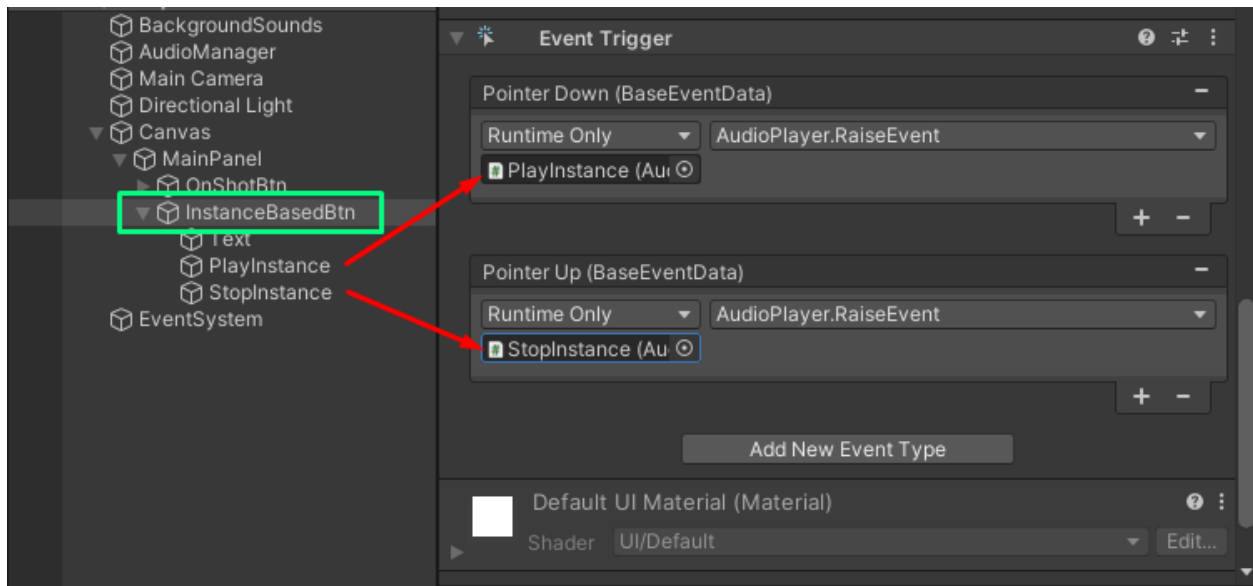
- **Looping Audio Implementation:**

This raises a question then how we will going to implement our looping sounds, so in FMOD we do have a couple of ways for doing so. We can use `FMODStudioEventEmitter.cs` class for playing these type of sounds or you can use our custom implemented class of `LoopingAudioPlayer.cs`. `LoopingAudioPlayer.cs` has edge as it implements parameter change aswell.



- **Instance based Audio Implementation:**

Here is example of playing instance-based sound from Inspector on some special events.



Now we will see how we will implement any specific instance-based sound from script.

```
FMOD.Studio.EventInstance footstepEvent = FMODUnity.RuntimeManager.CreateInstance(eventPath);

// Set position of event with optional offset
footstepEvent.set3DAttributes(FMODUnity.RuntimeUtils.To3DAttributes(transform.position + footstepOffset));
// Optionally you could set parameters here instead of using the supplied FMOD parameter script
// See the helper function below named "SetParameter" by Liam de Koster-Kjaer
SetParameter(footstepEvent, "concrete", concrete);
SetParameter(footstepEvent, "wood", wood);

// Play and release one-shot instance
footstepEvent.start();

footstepEvent.stop();
footstepEvent.release();
```

So this is the basic documentation for the implementation of all three types of sounds code in this plugin.

**That's All!**

**Thank You** 😊