TKG Studios
Easy Audio Manager - Documentation
Version 1.0

Easy Audio Manager

Getting Started

The Easy Audio Manager is an easy way to set up audio in your Unity game project with minimal effort. All you need to do to use it is to drag and drop your audio clips into the audio manager prefab, and create simple calls to it that can be accessed anywhere in your code! Read further to get started.

Demo Scene

The demo scene (***TKG Studios > Easy Audio Manager > Scenes > Demo***) contains a few buttons that play different sounds, from music to SFX to voice overs. You can click each of these buttons to test out the sound types.

Working with the Audio Manager

In order to start using the Audio Manager, all you need to do is grab the prefab and put it into your scene.

The Audio Manager Prefab (***TKG Studios > Easy Audio Manager > Prefabs > Audio Manager)*** is where the magic happens. You can easily just drag and drop the prefab into any of

your scene (so long as there is an Audio Listener in your scene — typically on the main camera) and start using it.

Audio Manager Script

In the Audio Managers inspector, you will find two components, the Audio Manager script and an audio source component. The Audio Manager script essentially is a collection of ***AudioAsset*** lists (***TKG Studios > Easy Audio Manager > Scripts > AudioAsset***). The lists are all assigned to their out Mixer Groups, which you can use to manipulate the volume level of each of the ***AudioAsset*** types.

The exposed properties of the AudioAsset lists are a string property (called name) , an AudioClip array property (called Clip) and an Audio Mixer Group property (called Mixer Group). You'll only need to manipulate specific properties on each audio list type in order to get this going, and a few of them will already be filled out for you when you drag in the prefab.

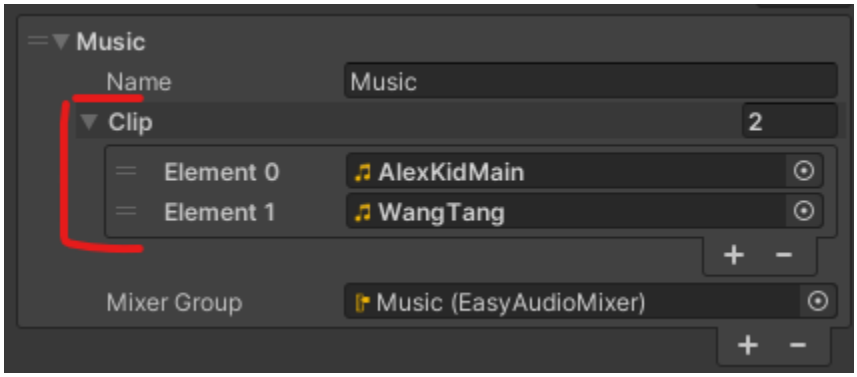
Music and Voice Over

The music list section will have the name of "Music", and the Voice Over list section will have the name "Voice Over" which should not be changed.

If it is changed, it will break the way that music is played in the project and will no longer work. Unless you have a good understanding of how to program, I suggest you leave these names as they are.

For the both lists, you only need to add new audio clips to the clip array property in their respective lists in order to play the music over voice overs in game. Simply click the "+" icon below the clip array and drag your music or VO tracks into the element list.

This screenshot uses the music list as an example.



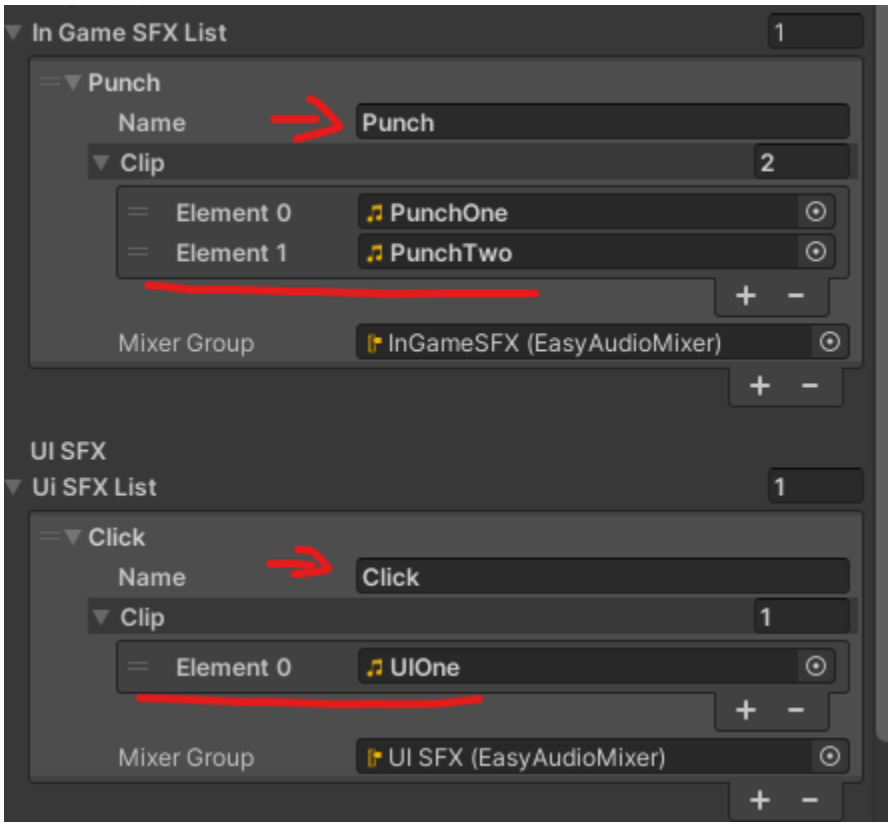
.

SFX

The SFX lists have a bit different set up.

For the In Game SFX list, you will have to create new items in the In Game SFX list, give them proper names, and add as many clips as you want to represent that sound into the clips array. The example in the demo uses a “Punch” sound effect, with two possible clips that can be played for that sound. This setup is great for games where you want slightly different random sound effects to be played during certain actions for variety, such as punching an enemy.

For the UI SFX, you essentially can set this up the same way, but you should only put one clip in each Clip array, as the UI SFX typically should be consistent within the game.



Controlling Audio In Game

In order to play audio in game, you only need to call the respective function on the ***UniversalAudioPlayer.cs*** (***TKG Studios > Easy Audio Manager > Scripts > UniversalAudioPlayer***).

First, make sure the script is using the EasyAudioManager namespace. Add this line to the top of your script declarations in any script that calls the simple audio manager.

using EasyAudioManager;

Music

To play music, you just need to call this function anywhere in your own scripts where you want the music to play.

UniversalAudioPlayer.PlayMusic(audioTrackID);

audioTrackID here represents the ID of the AudioTrack, which corresponds to the track element number in the Music List clip array.

For example, the ID for the first track that is in the list is 0, the second track is 1, and so on.

So, if you want to play track 0, you need only call ***UniversalAudioPlayer.PlayMusic(0);***

If you want to stop music, you need only call ***UniversalAudioPlayer.StopMusic();***

Voice Over

To play music, you just need to call this function anywhere in your own scripts where you want the music to play.

UniversalAudioPlayer.PlayVO(VOID);

audioTrackID here represents the ID of the AudioTrack, which corresponds to the track element number in the VO List clip array.

For example, the ID for the first track that is in the list is 0, the second track is 1, and so on.

So, if you want to play VO track 0, you need only call ***UniversalAudioPlayer.PlayVO(0);***

If you want to stop VO, you need only call ***UniversalAudioPlayer.StopVO();***

In Game SFX

To play In Game SFX, you need only call ***UniversalAudioPlayer.PlayInGameSFX(SFXName);***

SFXName here represents the name that was assigned to the Audio Asset element in the In Game SFX list.
For example, if you want to play a sound effect from the “Punch” list, you need only call ***UniversalAudioPlayer.PlayInGameSFX(“Punch”);***

UI SFX

To play UI SFX, you need only call ***UniversalAudioPlayer.PlayUISFXSFX(SFXName);***

SFXName here represents the name that was assigned to the Audio Asset element in the UI SFX list.
For example, if you want to play the “Click” sound effect, you need only call ***UniversalAudioPlayer.PlayUISFXSFX(“Click”);***

Audio Mixer

The Audio Mixer is found in (***TKG Studios > Easy Audio Manager > AudioMixer).***

The main use for the Audio Mixer is to control what sound level each of the items in the audio list is played. You can tuned these however you want, or just control them all at once by modifying the “Master” volume knob.

Simply slide the “Volume” slider wherever you’d like on each mixer group to change the volume level for the sounds in each group.