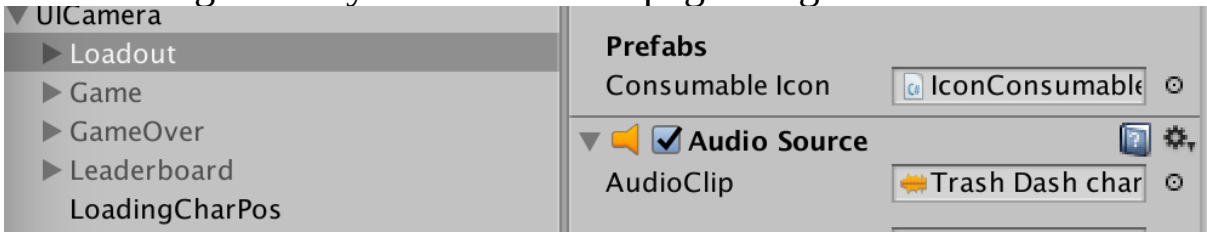
SCI-FI WORLDS MUSIC
Unity Setup

The tempo of the tracks are: 90bpm 4/4

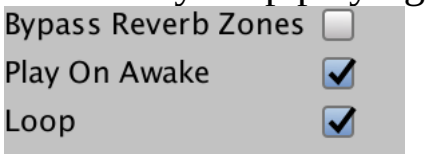
A music bed

1. Add an *audio source* on the camera (e.g. *UICamera* or *CameraRig*)
   This could be just on the camera's inspector panel, or on the subcategories if you have several pages. E.g.

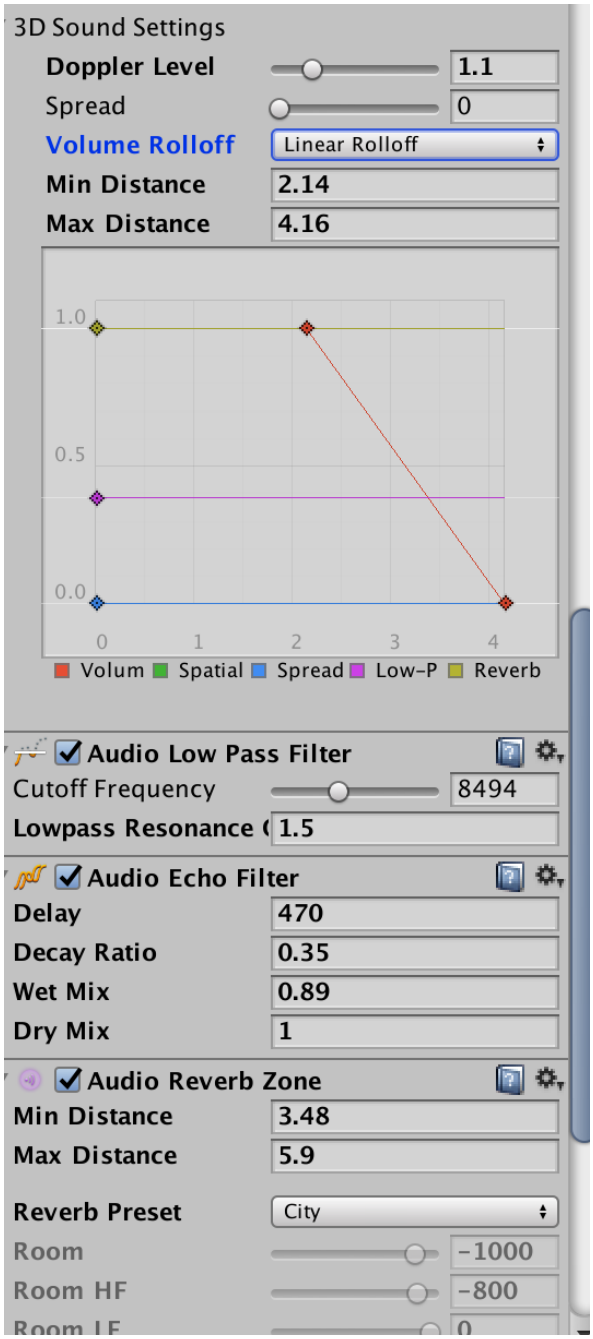


And don’t forget to add an *audio listener* on the camera in order to perceive the audio source.

2. If you want it to play as soon as you arrive to that scene make sure to tick “Play On Awake” and “Loop” for it to continually keep playing.

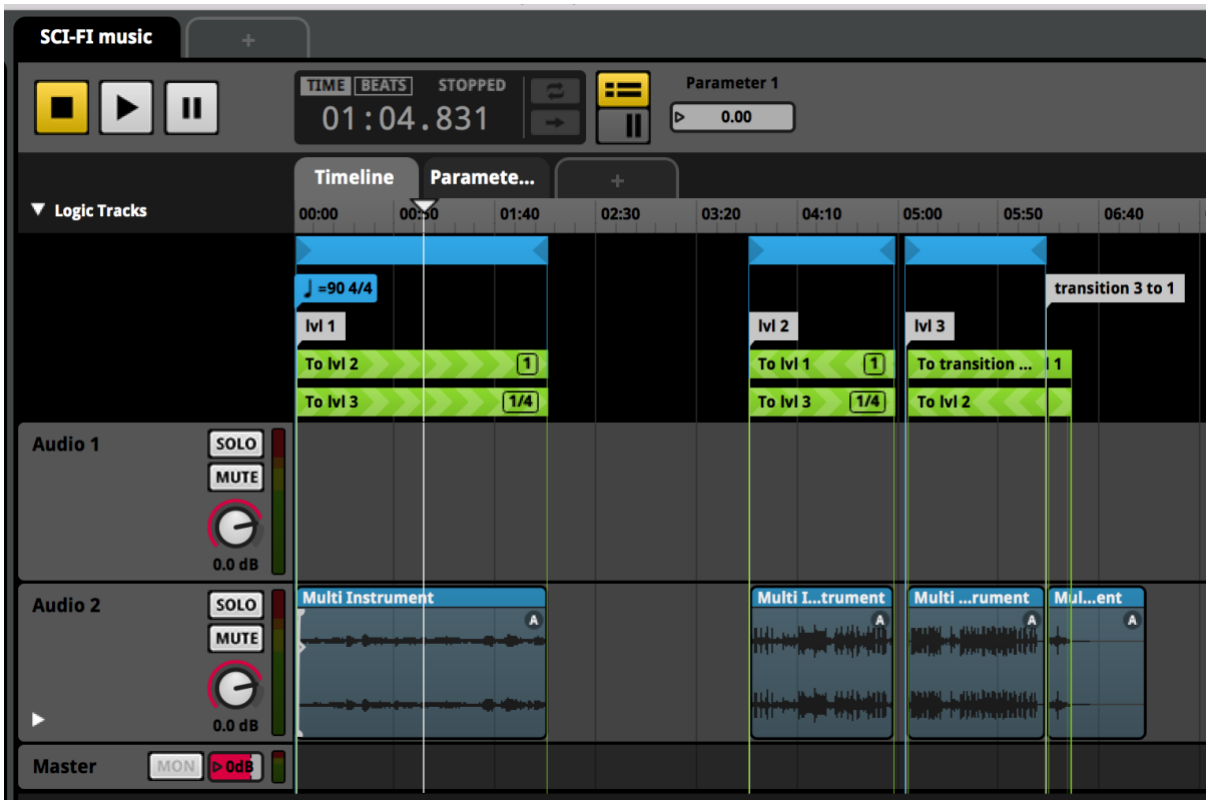


3. If you want to add the music into the 3D environment, add an *audio source* onto the desired object that will play the music. Set the *3D Sound Settings* as desired, and possibly play around with audio effects like Reverb and Echo, High or Low Filter to add the 3D effect. E.g:



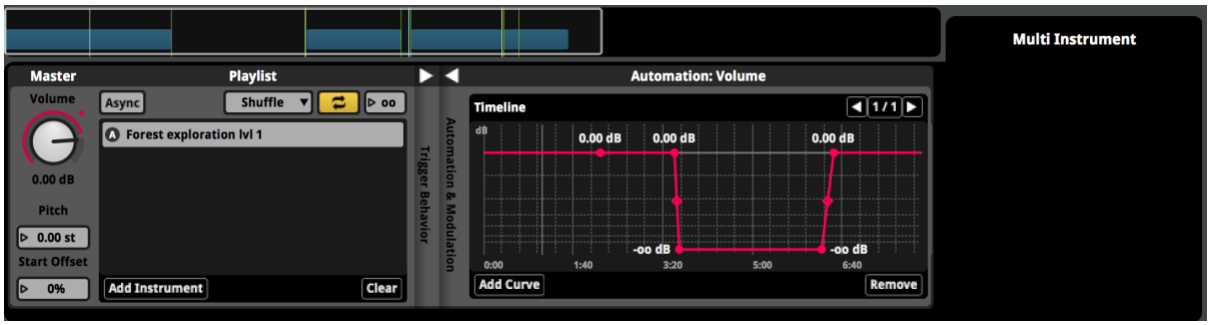
Several Musical layers

using Fmod



I have attached this example project in the folder for you to investigate by your own.
Please refer to the photo above for these explications

1. Create a *multi instrument* on the timeline and drop the music files corresponding to one level of intensity – if there are several variations for the same level of intensity, drag them into this one box. Do this for each level of intensity and transitions.



2. Create a loop region (blue bar) to repeat the music continually. Add a tempo marker

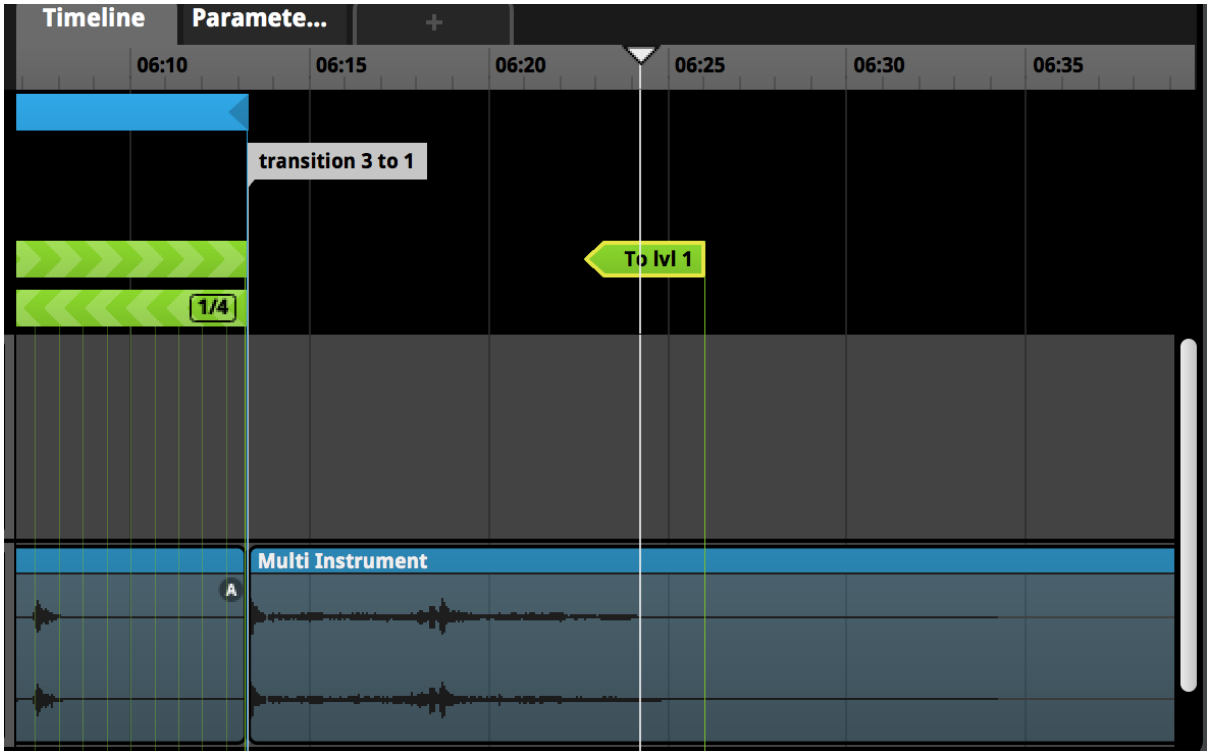
3. For each multi instrument – which represent each level of intensity (lvl 1 to 3) – create a *destination marker* (grey markers)
4. Create a *new parameter* by clicking on the + next to **Timeline**, in my case I just kept the boundaries from 0 to 1. If you are getting creative, you can assign the parameter to a distance or height level in the game in order to trigger changes in the music.

5. **In order to change the level of intensity**, you must create a *transition region* (green bars) for each level changes. For example, if you are in lvl 1, you need two transition regions to be able to pass from *lvl 1 to lvl 2*, and from *lvl 1 to lvl 3*. Repeat this for each lvl of intensity; *lvl 2 to lvl 3*, *lvl 2 to lvl 1*, *etc.*
   When selecting a transition region (green bar), determine a quantization that will divide the music in 1 bars or ¼ notes, for instance. These are time markers that trigger the change to a destination marker (after parameter modification).

6. In each *transition region*, add the parameter created earlier and divide the boundaries (0 to 1) in 3 sections for 3 levels

of intensity. *For example, 0 to 0.33 is for lvl 1, 0.33 to 0.65 is for lvl 2, 0.65 to 1 is for lvl 3.*

7. **To add transition segments** to a change in level intensity: create a *destination marker* where you placed the transition segment *multi instrument* on the timeline. Make sure to assign the parameter interval from the previous *transition region* of level intensity to the parameter interval usually assigned to the envisioned level of intensity. For example, if we are going from lvl 3 to lvl 1 but want to add a transition segment to connect both levels; we would assign *0 to 0.33* to the transition segment that will go to the *destination marker* for *transition segment.* Note that this interval (0 to 0.33) is usually assigned to the destination marker of lvl 1 directly.
   In this case, we will add a *transition* (which is a transition marker) to the end of the transition segment, and attach its destination to *destination marker lvl 1* (see below).
   To sum up, in lvl 3, there is a *transition region* destined to go to the *destination marker transition 3 to 1* (quantized at ¼ notes for high intenstity), that will transition when the parameter is in between 0 to 0.33 - usually assigned to lvl 1, because ultimately we want to go to lvl 1 but we added a transition segment to connect smoothly. At the *destination marker transition 3 to 1*, the transition segment will play and will go to lvl 1 when it arrives at the *transition marker* assigned to the *destination marker lvl 1*.



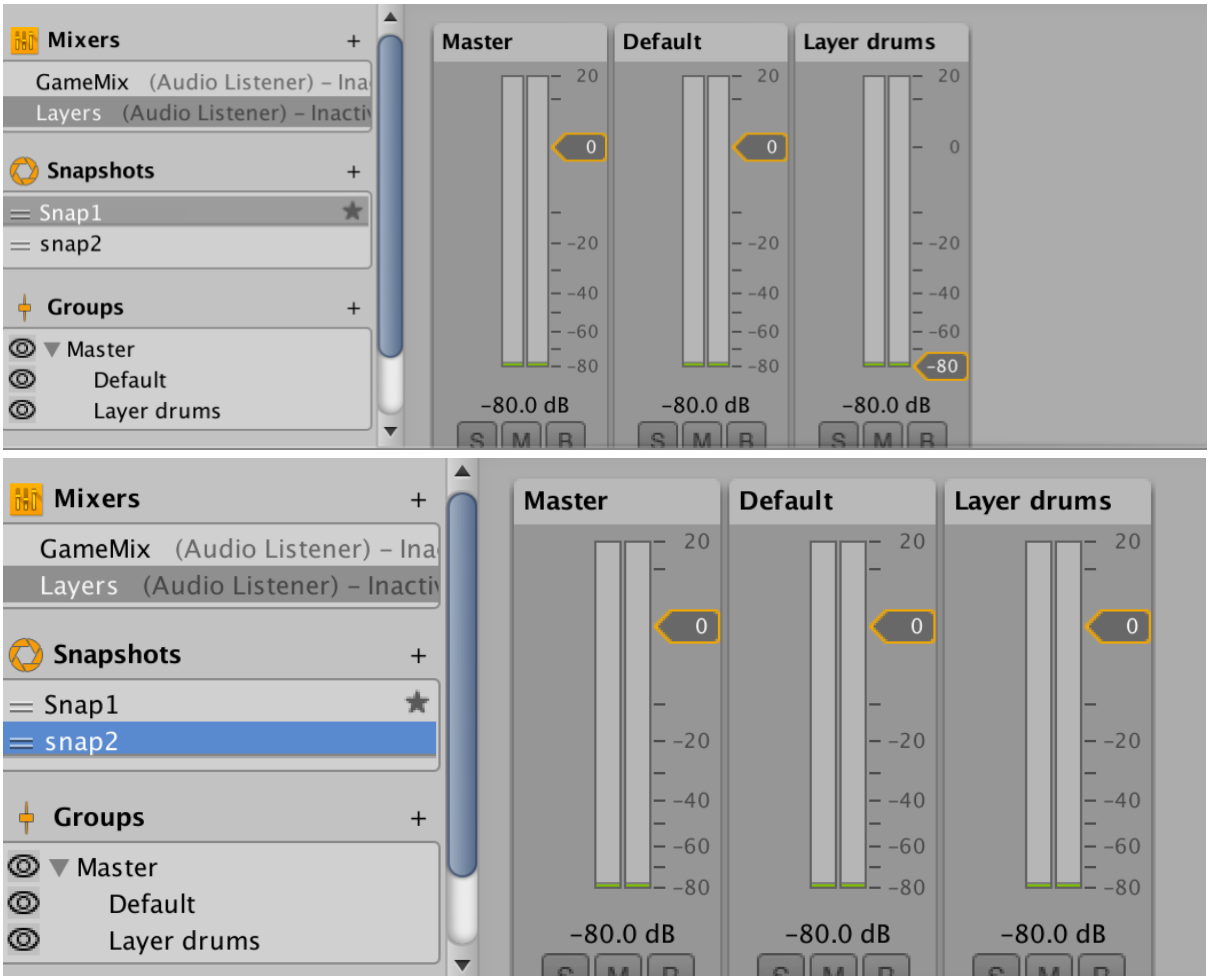
Or

Without using Fmod

1. Replace “chord” with the transition layer

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.Audio;
using TowerDefense.Level;

public class MusicControl : MonoBehaviour {

    public int waveNumber;
    public int waveThreshold;
    public AudioMixerSnapshot snap2;
    public AudioSource music;
    public AudioSource chord;
    public double startTime;
    public bool chordPlayed;

    // Use this for initialization
    void Start () {
        startTime = AudioSettings.dspTime - music.time;

    }

    // Update is called once per frame
    void Update () {
        waveNumber = LevelManager.instance.waveManager.waveNumber;
        if (waveNumber > waveThreshold) {
            snap2.TransitionTo (1);
        }
        if (waveNumber > 4 && !chordPlayed)
        {
            PlayChord ();
        }
    }
    void PlayChord()
    {
        double beat = 130 / 60;
        double time = AudioSettings.dspTime - startTime;
        double remainder = beat - (time % beat);
        chord.PlayScheduled (AudioSettings.dspTime + remainder);
        Debug.Log ("clip scheduled to play");
        chordPlayed = true;

    }

}

```

2. Code in order to play next level of intensity in music after



the transition by triggering a mixer snapshot. For example: