

Modul 12 – Memanipulasi Sound

A. TUJUAN

- Mahasiswa dapat mengolah sound yang dapat memutar audio pada suatu object
- Mahasiswa dapat mengolah sound dengan menambahkan kontrol volume.
- Mahasiswa dapat mengolah sound dengan membuat soundtrack dinamis.
- Mahasiswa dapat mengatur sound dengan menyeimbangkan audio dalam game.

B. PETUNJUK

1. Awali setiap kegiatan praktikum dengan berdoa
2. Baca dan pahami tujuan, dasar teori, dan latihan-latihan praktikum dengan baik
3. Kerjakan tugas-tugas praktikum dengan baik, sabar dan jujur
4. Tanyakan kepada dosen apabila ada hal-hal yang kurang jelas

C. ALOKASI WAKTU: 6 jam pelajaran

D. DASAR TEORI

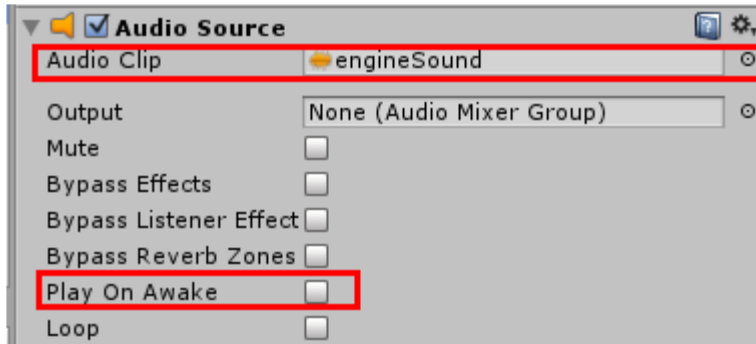
Modul 12 adalah modul lanjutan yang membahas cara untuk memainkan dan memanipulasi sound, sehingga game yang dibuat lebih menarik dan lebih nyata.

E. LATIHAN PRAKTIKUM

1. Waiting for audio to finish playing before auto-destructing an object

Untuk melakukan praktikum ini, kita membutuhkan *engineSound* audio yang terdapat dalam folder 1362_09_04.

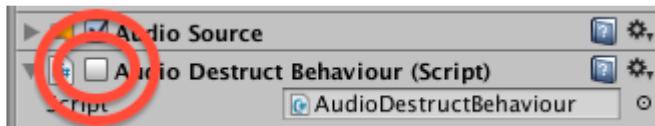
1. Buatlah sebuah GameObject dan beri nama **AudioObject**. Kemudian tambahkan komponen **Audio Source** (menu **Component | Audio | Audio Source**).
2. Import *engineSound* audio clip dan drag dari **Project** view menuju **Audio Clip** parameter yang ada di komponen **Audio Source** dari **AudioObject**, dan hapus centang komponen **Play On Awake**:



3. Tambahkan **C# script** untuk **AudioObject**:

```
using UnityEngine;
using System.Collections;
public class AudioDestructBehaviour : MonoBehaviour
{
    private AudioSource audioSource;
    void Start()
    {
        audioSource = GetComponent<AudioSource>();
    }
    private void Update()
    {
        if (!audioSource.isPlaying)
            Destroy(gameObject);
    }
}
```

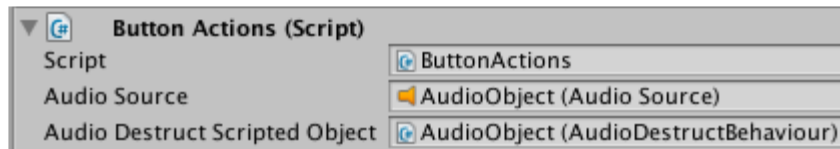
4. Pada **Inspector** view, (un-check) komponen *AudioDestructBehaviour* dari **AudioObject**:



5. Buatlah **C# Script** dan beri nama **ButtonActions**:

```
using UnityEngine;
using System.Collections;
public class ButtonActions : MonoBehaviour
{
    public AudioSource audioSource;
    public AudioDestructBehaviour audioDestructScriptedObject;
    public void PlaySound()
    {
        if (!audioSource.isPlaying)
            audioSource.Play();
    }
    public void DestroyAfterSoundStops()
    {
        audioDestructScriptedObject.enabled = true;
    }
}
```

6. Buatlah **UI button** ganti nama dengan **PlaySoundButton** dan beri nama *Play Sound* pada button, dan lampirkan **ButtonActions** script menuju button tersebut.
7. Pilih **PlaySoundButton** pada **Hierarchy**, buatlah new on-click event handler, drag **PlaySoundButton** kedalam slot **Object**, dan pilih **PlaySound()** function.
8. Pilih **PlaySoundButton** pada **Hierarchy** panel, drag **AudioObject** menuju **Audio Source**. Kemudian drag **AudioObject** script menuju **AudioDestructScriptedObject**, seperti pada screenshot dibawah ini:



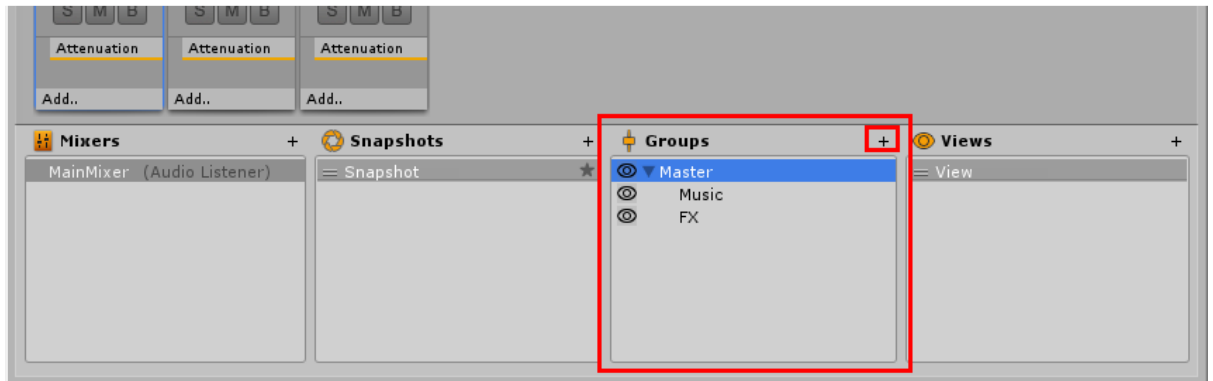
9. Buatlah UI button kedua dengan nama **DestoryWhenSoundFinishedButton**, dan beri nama *Destroy When Sound Finished* pada button, dan lampirkan **ButtonActions** script pada button tersebut.
10. Pilih **DestoryWhenSoundFinishedButton** pada **Hierarchy**, buatlah new on-click event handler, drag **PlaySoundButton** kedalam slot **GO**, kemudian pilih **DestroyAfterSoundStops()** function.
11. Lakukan seperti pada button lainnya, pilih **DestoryWhenSoundFinishedButton** pada **Hierarchy** panel, Kemudian drag **AudioObject** script menuju **MyAudioDestructObect**.

2. Adding volume control with Audio Mixers

Untuk melakukan praktikum ini, kita membutuhkan Unity package dengan nama *Volume.unitypackage*. File ini terdapat di dalam folder 1362_09_05.

1. Import *Volume.unitypackage* kedalam project.
2. Buka **Volume** scene (terdapat di folder **Assets | Volume**). Jalankan scene dan jalan menuju the tembok hijau semitransparent pada tunnel, gunakan W A S D keys (tekan Shift key untuk lari). Anda akan bisa mendengarkan:
 - Musik soundtrack yang berulang
 - lonceng berdering
 - Suara robot ketika karakter bertabrakan dengan tembok

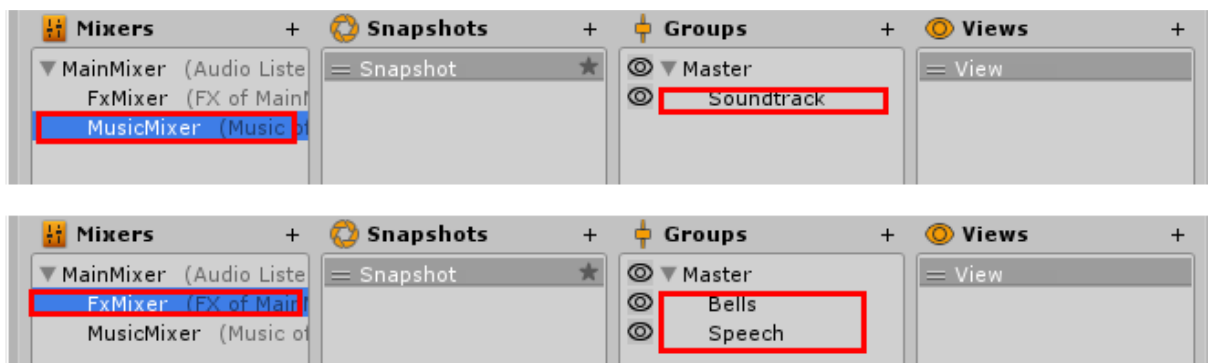
3. Dari **Project** view, gunakan menu **Create** untuk menambahkan **Audio Mixer** kedalam project. Ganti nama dengan **MainMixer** kemudian double-click untuk membuka **Audio Mixer** window.
4. Dari **Groups** view, Pilih **Master** kemudian klik tanda “+” untuk menambahkan percabangan pada grup **Master**. Beri nama **Music**, kemudian lakukan lagi hal tersebut dan beri nama **FX**, seperti pada screenshot dibawah ini:



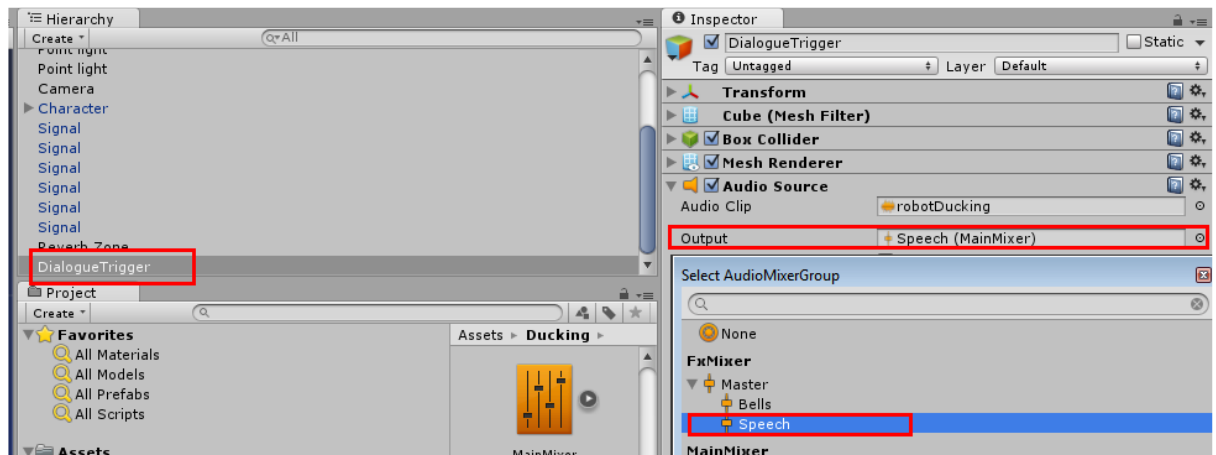
5. Dari **Mixers** view, Pilih **MainMixer** kemudian klik tanda “+” untuk menambahkan **Mixer** baru pada project. Beri nama **MusicMixer**, kemudian drag menuju **MainMixer** dan pilih **Music** sebagai **Output**. Ulangi operasi tersebut untuk menambahkan mixer dengan nama **FxMixer** pada project dan pilih **FX** sebagai **Output**:



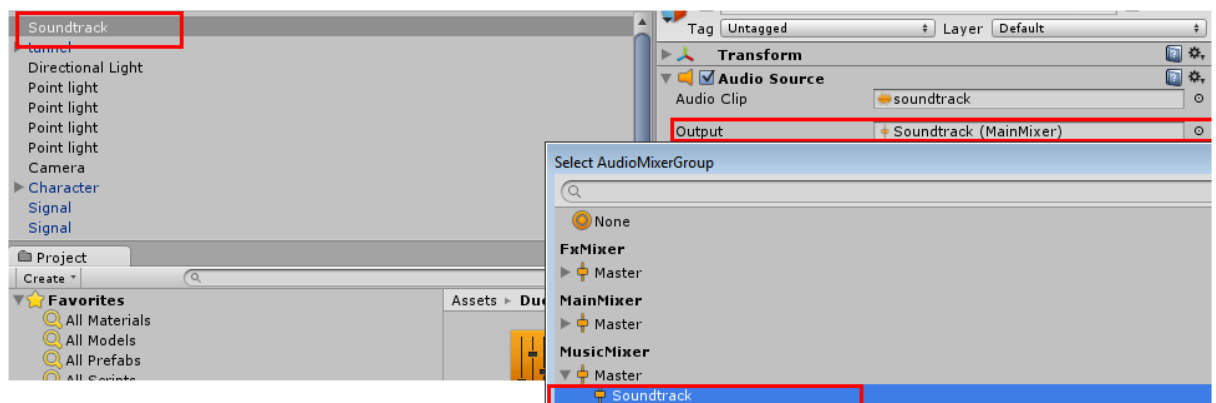
6. Pilih **MusicMixer** kemudian pilih **Master** di dalam **Groups** dan buatlah percabangan dengan nama **Soundtrack**. Lalu pilih **FxMixer** dan tambahkan dua percabangan pada **Master** di dalam **Groups** dengan nama **Speech** dan **Bells**, seperti dibawah ini:



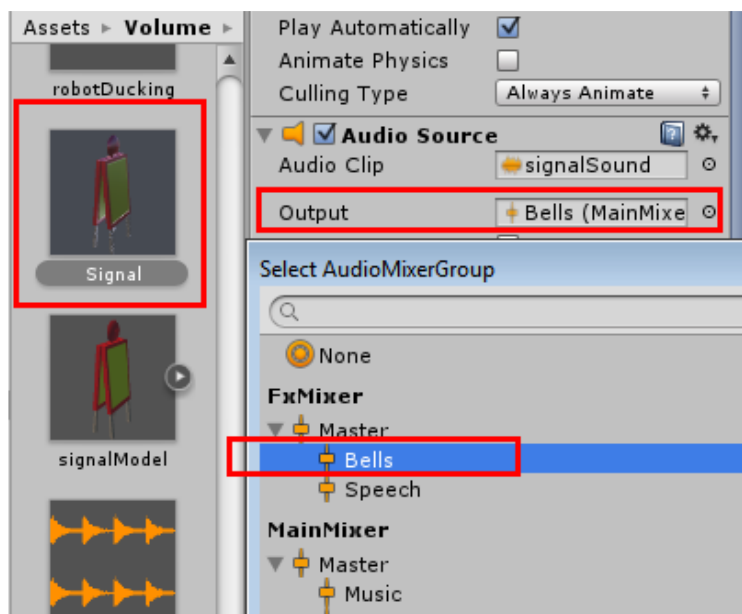
7. Dari **Hierarchy** view, pilih **DialogueTrigger** object. Kemudian pada **Inspector** view, rubah **Output** menjadi **FxMixer | Speech** pada komponen **Audio Source**:



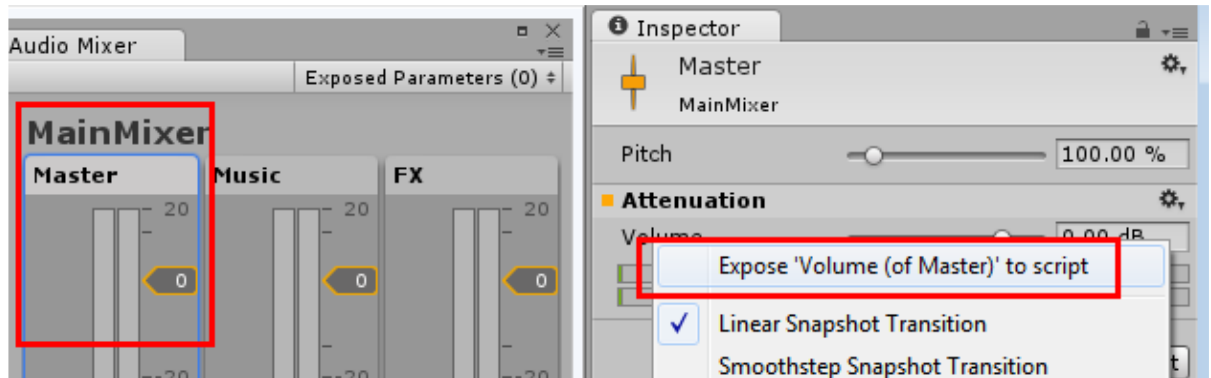
8. Pilih **Soundtrack** GameObject. Dari **Inspector** view, cari komponen **Audio Source** dan rubah **Output** menjadi **MusicMixer | Soundtrack**:



9. Kemudian dari folder **Assets** pada **Project** view, pilih **Signal** prefab. Dari **Inspector** view, akses komponen **Audio Source** dan rubah **Output** menjadi **FxMixer | Bells**:



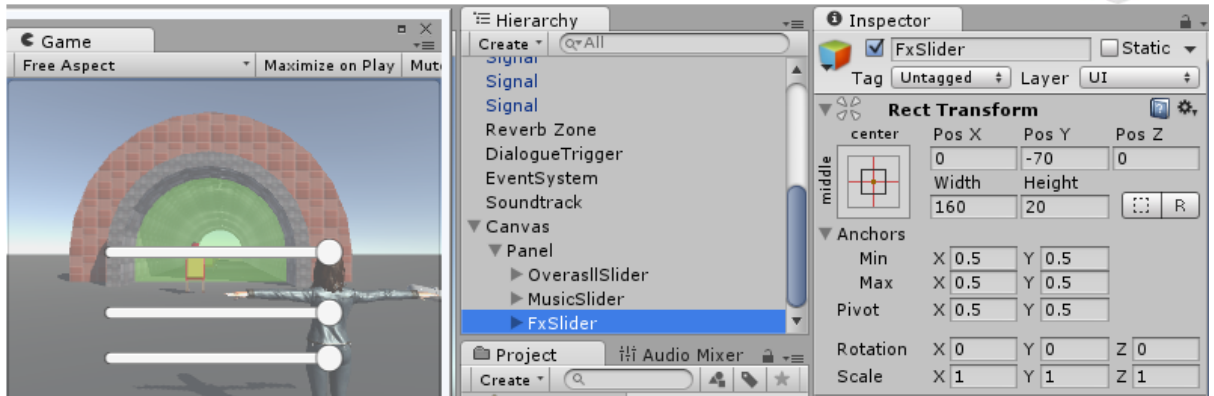
10. Dari **Audio Mixer** window, pada **MainMixer** pilih **Master**. Kemudian dari **Inspector** view, right-click **Volume** pada komponen **Attenuation**. Pilih **Expose 'Volume (of Master) to script** seperti screenshot dibawah ini. Lakukan hal yang sama pada **Music** and **FX** tracks:



11. Pada **Audio Mixer** dengan memilih **MainMixer**, akses menu **Exposed Parameters**. Kemudian, right-click pada **MyExposedParam** dan ganti nama menjadi *OverallVolume*. Kemudian ganti nama **MyExposedParam1** menjadi *MusicVolume* and **MyExposedParam2** menjadi *FxVolume*.
12. Dari **Project** view, buatlah **C# Script** dan beri nama *VolumeControl*:

```
using UnityEngine;
using UnityEngine.Audio;
using System.Collections;
public class VolumeControl : MonoBehaviour
{
    public AudioManager myMixer;
    private GameObject panel;
    private bool isPaused = false;
    void Start()
    {
        panel = GameObject.Find("Panel");
        panel.SetActive(false);
    }
    void Update()
    {
        if (Input.GetKeyUp(KeyCode.Escape))
        {
            panel.SetActive(!panel.activeInHierarchy);
            if (isPaused)
                Time.timeScale = 1.0f;
            else
                Time.timeScale = 0.0f;
            isPaused = !isPaused;
        }
    }
    public void ChangeMusicVol(float vol)
    {
        myMixer.SetFloat("MusicVolume", Mathf.Log10(vol) * 20f);
    }
    public void ChangeFxVol(float vol)
    {
        myMixer.SetFloat("FxVolume", Mathf.Log10(vol) * 20f);
    }
    public void ChangeOverallVol(float vol)
    {
        myMixer.SetFloat("OverallVolume", Mathf.Log10(vol) * 20f);
    }
}
```

13. Dari **Hierarchy** view, gunakan menu **Create** untuk menambah **Panel** ke dalam scene (**Create | UI | Panel**). Maka akan muncul panel dengan nama **Canvas** pada scene.
14. Dari **Hierarchy** view, gunakan menu **Create** untuk menambah **Slider** ke dalam scene (**Create | UI | Slider**). Buatlah hal tersebut pada objek **Panel** sebagai percabangannya.
15. Ganti nama slider tersebut menjadi **OverallSlider**. Duplikat slider tersebut dan ganti nama menjadi **MusicSlider**. Kemudian pada **Inspector** view, **Rect Transform** rubah parameter **Pos Y** menjadi -40.
16. Duplikat slider **MusicSlider** dan ganti nama menjadi **FxSlider**. Kemudian rubah parameter **Pos Y** menjadi -70:

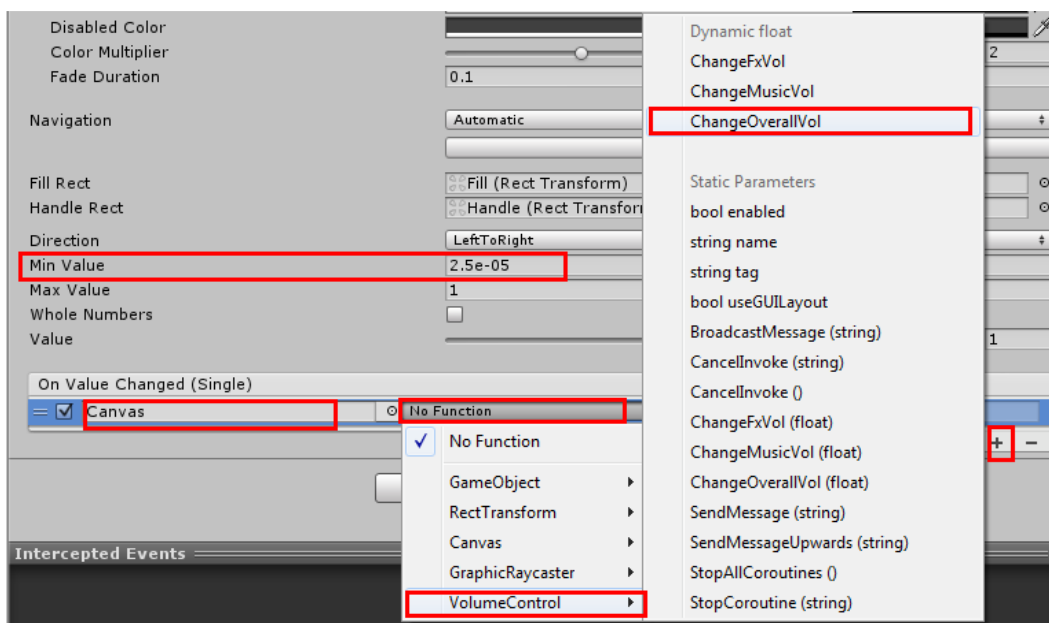


17. Pilih **Canvas** GameObject dan tambahkan script **VolumeControl** kedalamnya.

Kemudian ganti isi kolom **MyMixer** dari **Volume Control** dengan **MainMixer**:



18. Pilih komponen **OverallSlider**. Dari **Inspector** lihat pada komponen **Slider**, rubah **Min Value** menjadi 0.000025 (or **2.5e-05**). Kemudian dibawah list **On Value Changed**, klik tanda “+” untuk menambah sebuah aksi. Dari **Hierarchy** panel, drag **Canvas** kedalam slot **Object** dan pilih menu, pilih opsi **VolumeControl | ChangeOverallVol**, seperti screenshot dibawah ini.

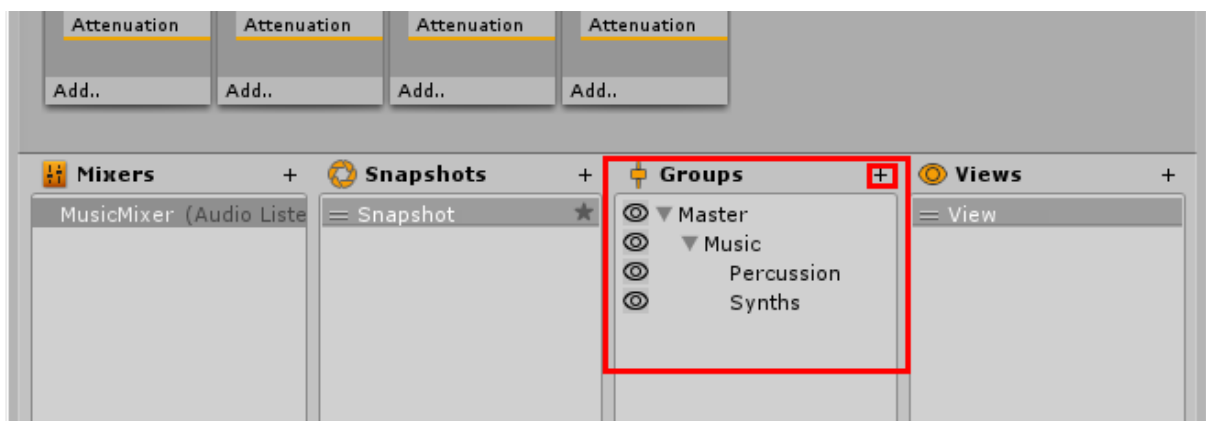


19. Ulangi langkah terakhir untuk **MusicSlider** dan **FxSlider**, tetapi pilih opsi **ChangeMusicVol** dan **ChangeFxVol** masing-masing pada menu.
20. Jalankan scene. Anda akan dapat menggunakan sliders ketika menekan tombol *Escape* pada keyboard dan mengatur volume settings.

3. Making a dynamic soundtrack with Snapshots

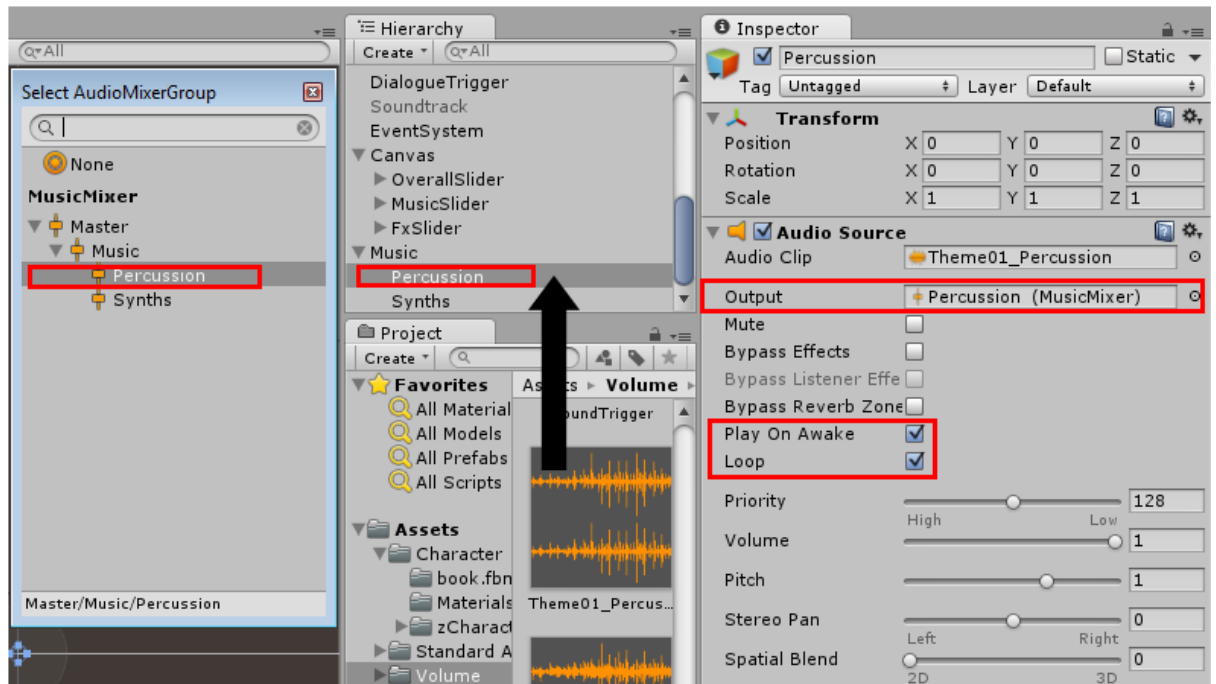
Untuk melakukan praktikum ini, kita membutuhkan Unity package dengan nama *DynamicSoundtrack*, dan dua soundtrack audio clips dengan format *.ogg*: *Theme01_Percussion* dan *Theme01_Synths*. Semua file tersebut terdapat pada folder 1362_09_06.

1. Import *DynamicSoundtrack* package dan dua file soundtrack audio clips dengan format *.ogg*: *Theme01_Percussion* dan *Theme01_Synths* kedalam Unity Project anda.
2. Buka level dengan nama **Dynamic**.
3. Dari **Project** view, gunakan menu **Create** untuk menambahkan **Audio Mixer** kedalam project. Ganti nama dengan **MusicMixer** kemudian double-click untuk membuka **Audio Mixer** window.
4. Dari **Groups** view, Pilih **Master** kemudian klik tanda “+” untuk menambahkan percabangan pada grup **Master**. Beri nama **Music** kemudian tambahkan dua percabangan dari **Music** dengan nama **Percussion** dan **Synths**:

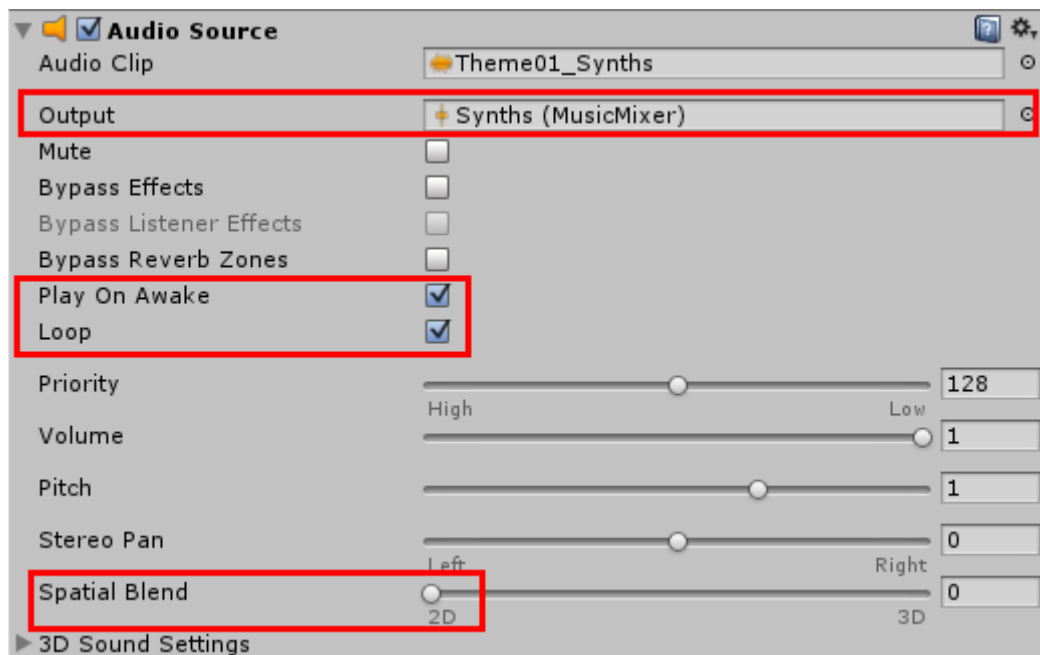


5. Dari **Hierarchy** view, buatlah **Empty** GameObject. Beri nama **Music** kemudian tambahkan dua **Empty Child** GameObjects sebagai percabangannya. Ganti nama keduanya dengan nama **Percussion** and **Synth**.
6. Dari **Project** view, drag **Audio Clip** dengan nama **Theme01_Percussion** kedalam **Percussion** GameObject di **Hierarchy**. pilih **Percussion** dan pada **Inspector** view,

akses komponen **Audio Source**. Rubah **Output** menjadi **Percussion (MusicMixer)**, pastikan opsi **Play On Awake** ter-centang, cek opsi **Loop**, dan pastikan **Spatial Blend** pada **2D**, seperti pada screenshot dibawah ini:



7. Drag file audio **Theme01_Synths** kedalam **Synths** GameObject. Dari **Inspector** view, rubah **Output** menjadi **Synths (MusicMixer)**, Pastikan opsi **Play On Awake** ter-centang, cek opsi **Loop**, dan pastikan **Spatial Blend** pada **2D**, seperti dibawah ini:



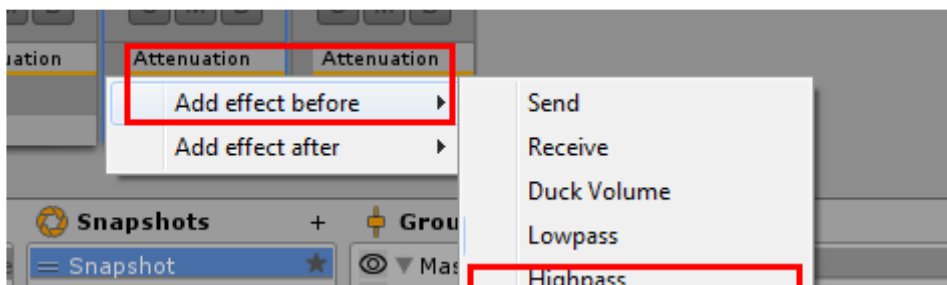
8. Buka **Audio Mixer** dan jalankan scene. Kita akan menggunakan mixer untuk mengatur soundtrack pada awal scene. Saat scene berjalan, klik pada **Edit in Play**

Mode button, seperti pada screenshot dibawah ini, terletak diatas **Audio Mixer**.

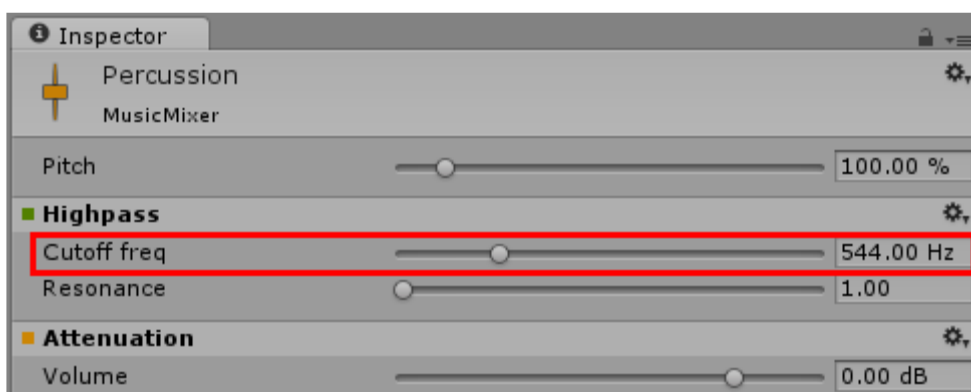
Kemudian turunkan volume pada **Synths** menjadi **-30 dB**:



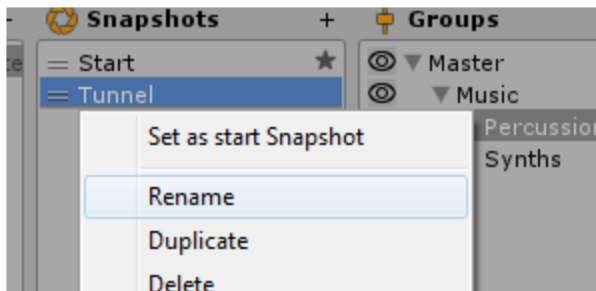
9. Pilih **Percussion** track. Right-click **Attenuation** dan tambahkan **High-pass** pada add effect before:



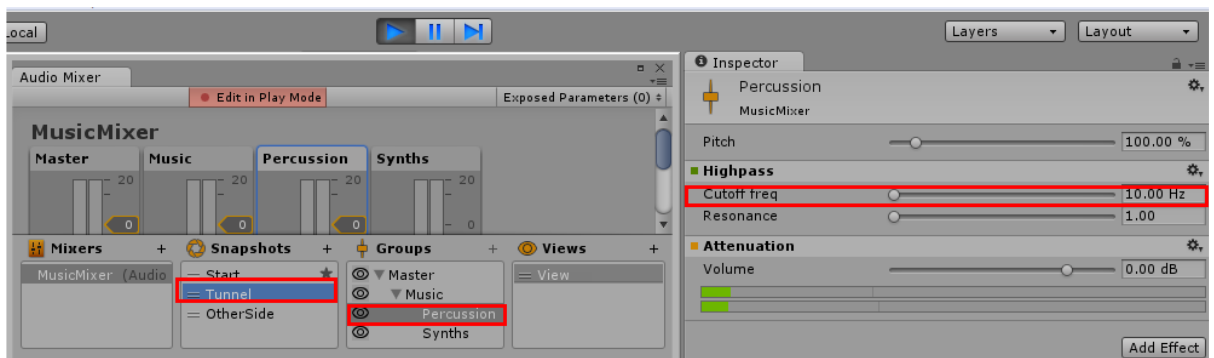
10. Dari **Inspector** view, rubah **Cutoff frequency** dari efek **High-pass** menjadi **544.00 Hz**:



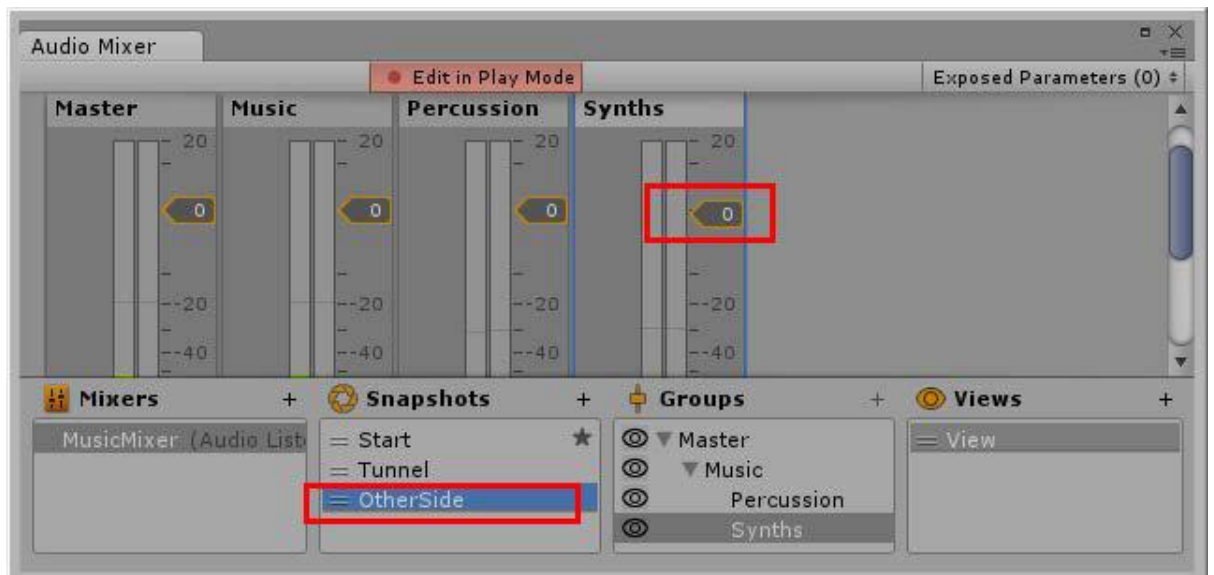
11. Dari **Snapshots** view, right-click pada **Snapshot** dan ganti nama dengan **Start**.
Kemudian right-click pada **Start** dan pilih opsi **Duplicate**. Ganti nama snapshot tersebut menjadi **Tunnel**, seperti dibawah ini:



12. Pilih **Tunnel** snapshot. Kemudian dari **Inspector** view, rubah **Cutoff frequency** pada efek **Highpass** menjadi **10.00 Hz**:

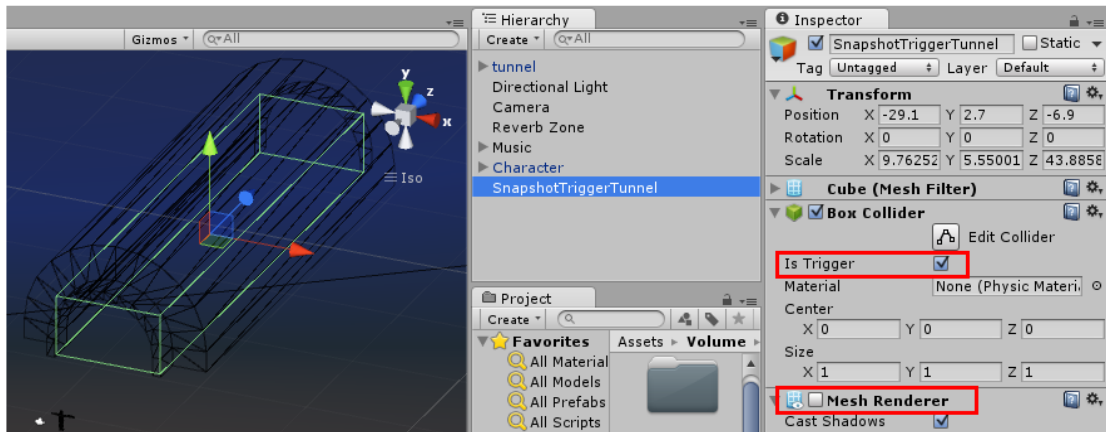


13. Ganti antara **Tunnel** and **Start** snapshots. Anda akan mendengarkan perubahannya.
14. Duplikat **Tunnel** snapshot, ganti dengan nama **OtherSide**, dan pilih itu.
15. Naikkan volume dari **Synths** track sampai dengan **0 dB**:

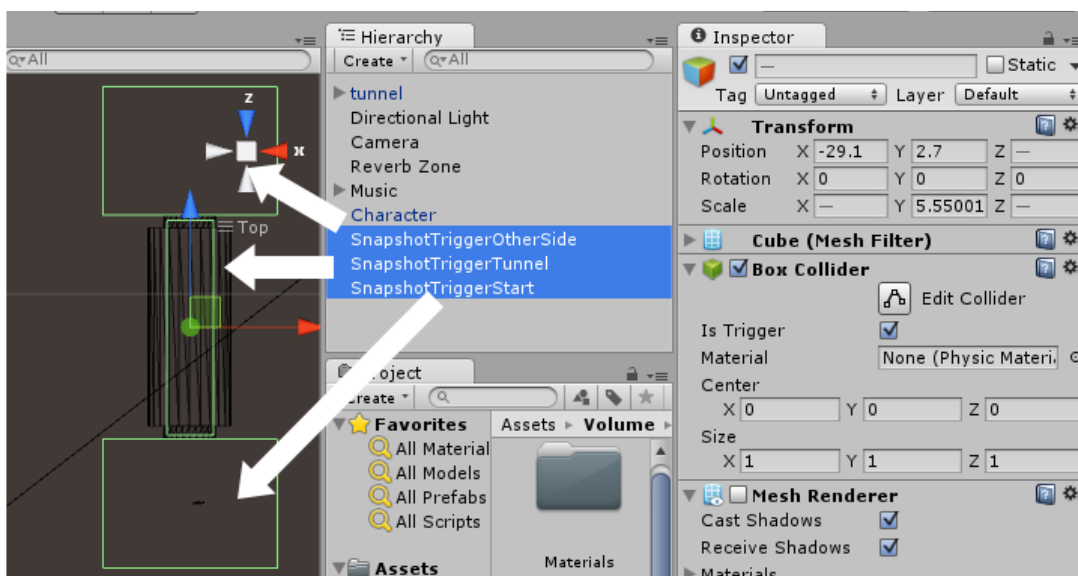


16. Saat ini kita memiliki tiga **Snapshots**, buat sebuah triggers untuk membuat transisi diantaranya. Dari **Hierarchy** view, gunakan menu **Create** untuk menambahkan **Cube** pada scene (**Create | 3D Object | Cube**).
17. Pilih **Cube** dan ganti dengan nama *SnapshotTriggerTunnel*. Kemudian dari **Inspector** view, akses komponen **Box Collider** dan centang opsi **Is Trigger**, seperti pada

screenshot dibawah ini. Lalu, uncheck komponen **Mesh Renderer**. Lalu sesuaikan ukuran dan posisi pada interior milik tunnel scene:



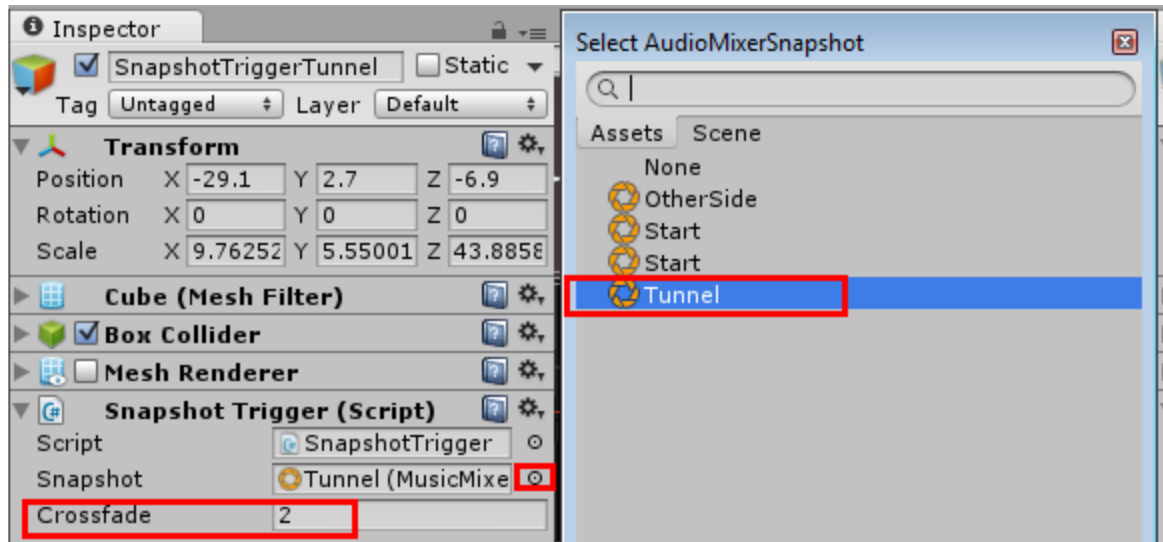
18. Buat dua duplikat dari *SnapshotTriggerTunnel* dan ganti dengan nama *SnapshotTriggerStart* dan *SnapshotTriggerOtherSide*. Lalu sesuaikan ukuran dan posisi, sehingga mereka menempati area sebelum pintu masuk tunnel. Seperti screenshot dibawah ini:



19. Pada **Project** view, Buatlah **C# Script** dan beri nama *SnapshotTrigger*:

```
using UnityEngine;
using UnityEngine.Audio;
using System.Collections;
public class SnapshotTrigger : MonoBehaviour
{
    public AudioManagerSnapshot snapshot;
    public float crossfade;
    private void OnTriggerEnter(Collider other)
    {
        snapshot.TransitionTo(crossfade);
    }
}
```

20. Simpan script anda lalu lampirkan pada *SnapshotTriggerTunnel*, *SnapshotTriggerStart*, and *SnapshotTriggerOtherSide* objects.
21. Pilih *SnapshotTriggerTunnel*. Kemudian dari **Inspector** view, akses komponen **Snapshot Trigger**, atur **Snapshot** menjadi **Tunnel**, dan **Crossfade** menjadi 2, seperti screenshot dibawah ini:



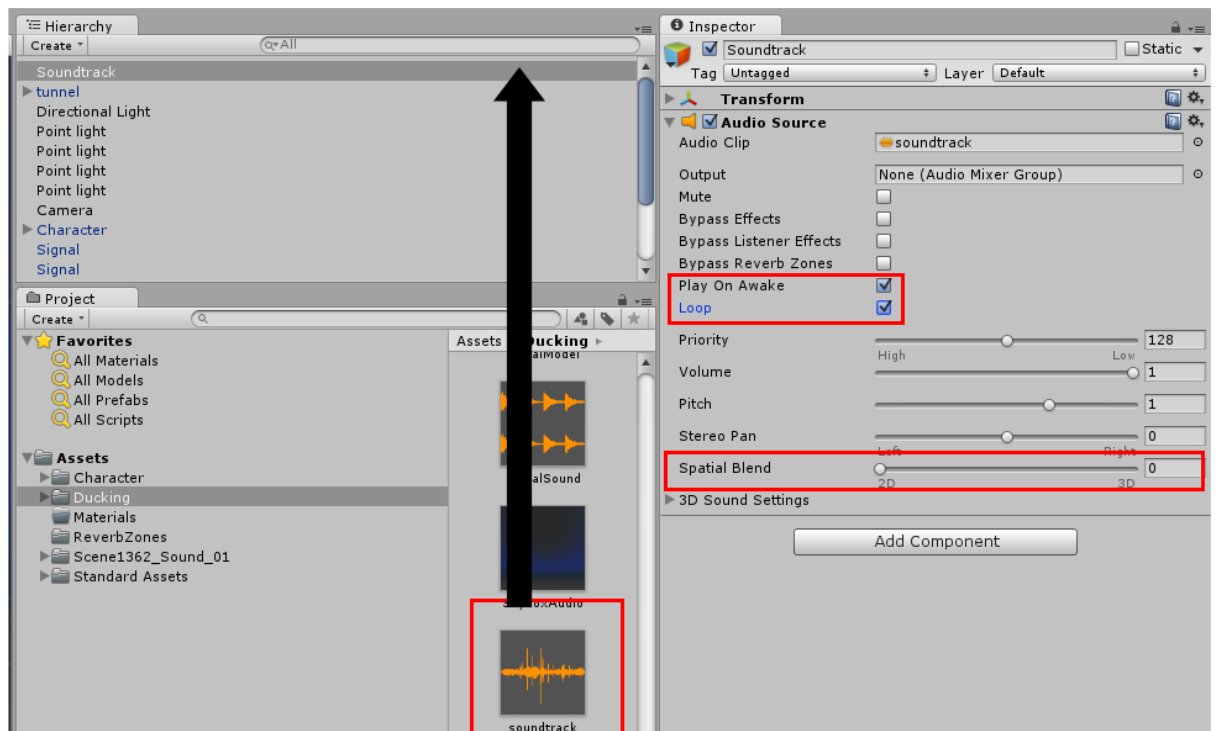
22. Rubah *SnapshotTriggerStart* and *SnapshotTriggerOtherSide* dengan mengatur **Snapshots** menjadi **Start** and **OtherSide** pada masing-masing.
23. Jalankan scene. background music akan berganti ketika karakter bergerak dari titik awal, melalui terowongan, dan sampai ke sisi lain.

4. Balancing in-game audio with Ducking

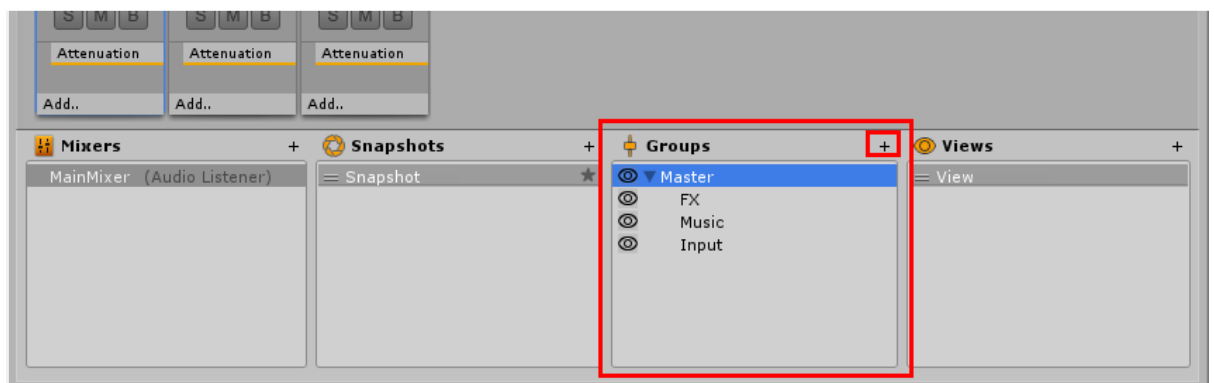
Untuk melakukan praktikum ini, kita membutuhkan *soundtrack.mp3* audio clip dan sebuah Unity package bernama *Ducking.unpackage*. Semua file tersebut terdapat pada folder 1362_09_07.

1. Import *Ducking.unpackage* and *soundtrack.mp3* kedalam project anda.
2. Buka **Ducking** scene (tersedia pada folder **Assets | Ducking**). Jalankan scene dan jalan menuju the tembok hijau semitransparent pada tunnel, gunakan W A S D keys (tekan Shift key untuk lari). Anda akan mendengar **robotDucking** audio clip ketika karakter bertabrakan dengan tembok.
3. Dari menu **Create** yang berada di atas **Hierarchy** view, pilih **Create Empty** untuk menambahkan GameObject baru pada scene. Ganti dengan nama **Soundtrack**.

4. Drag **soundtrack** audio clip yang sudah ter-import menuju **Soundtrack** GameObject. Kemudian pilih **Soundtrack** object dan dari **Inspector** view, komponen **Audio Source**, centang opsi **Loop**. Pastikan opsi **Play On Awake** ter-centang dan **Spatial Blend** pada **2D**, seperti pada screenshot dibawah ini:



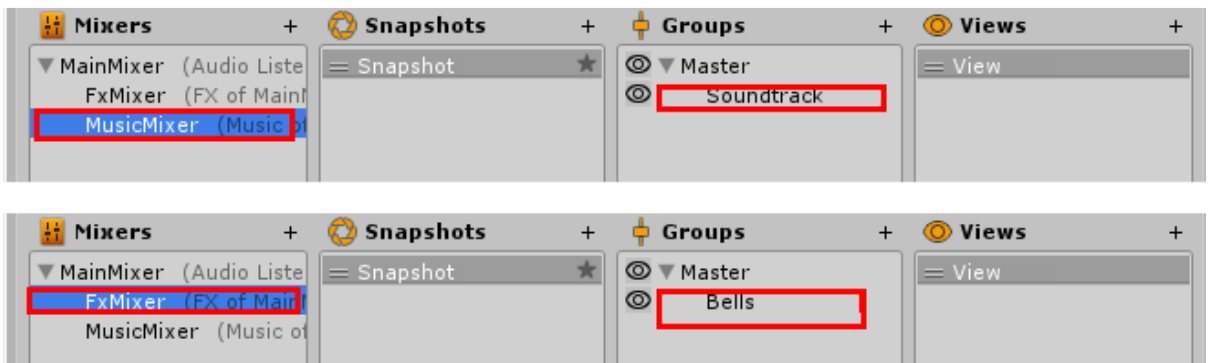
5. Dari **Project** view, gunakan menu **Create** untuk menambahkan **Audio Mixer** kedalam project. Ganti dengan nama **MainMixer** kemudian double-click untuk membuka **Audio Mixer** window.
6. Dari **Groups** view, Pilih **Master** kemudian klik tanda “+” untuk menambahkan percabangan pada grup **Master**. Beri nama **Music**, kemudian lakukan lagi hal tersebut dan beri nama **FX**, dan terakhir lakukan lagi dan beri nama **Input** seperti pada screenshot dibawah ini:



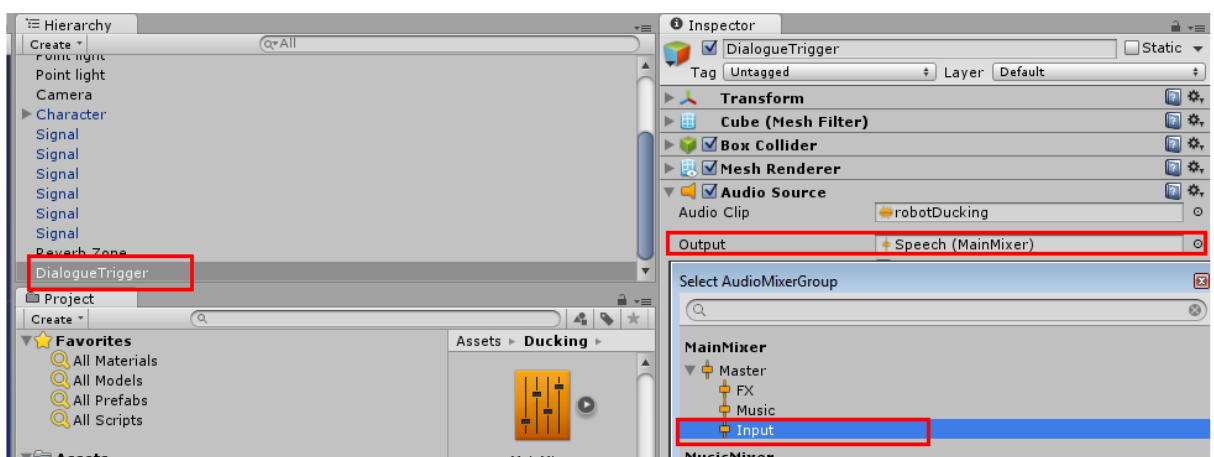
7. Dari **Mixers** view, Pilih **MainMixer** kemudian klik tanda “+” untuk menambahkan **Mixer** baru pada project. Beri nama **MusicMixer**, kemudian drag menuju **MainMixer** dan pilih **Music** sebagai **Output**. Ulangi operasi tersebut untuk menambahkan mixer dengan nama **FxMixer** pada project dan pilih **FX** sebagai **Output**:



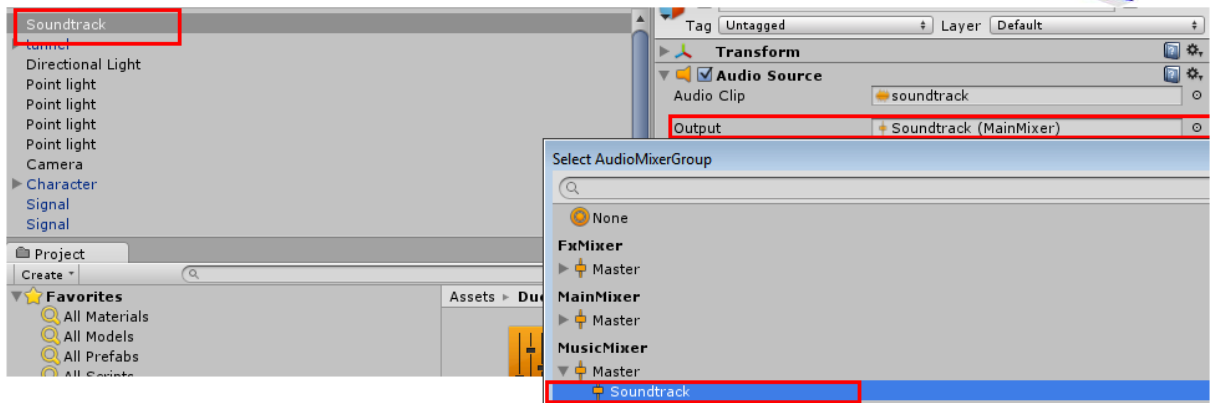
8. Pilih **MusicMixer** kemudian pilih **Master** di dalam **Groups** dan buatlah percabangan dengan nama **Soundtrack**. Lalu pilih **FxMixer** dan tambahkan dua percabangan pada **Master** di dalam **Groups** dengan nama **Bells**, seperti dibawah ini:



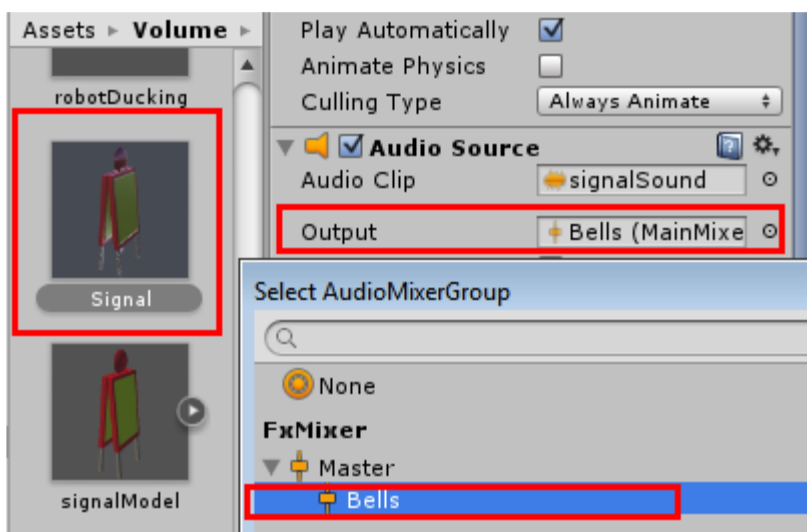
9. Dari **Hierarchy** view, pilih **DialogueTrigger** object. Kemudian pada **Inspector** view, rubah **Output** menjadi **MainMixer | Input** pada komponen **Audio Source**:



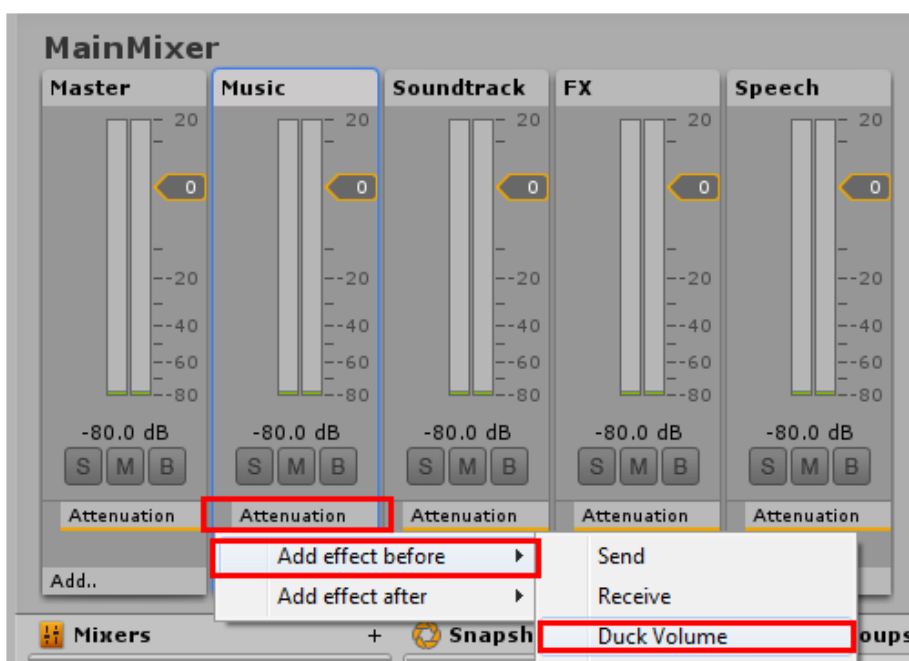
10. Pilih **Soundtrack** GameObject. Dari **Inspector** view, cari komponen **Audio Source** dan rubah **Output** menjadi **MusicMixer | Soundtrack**:



11. Kemudian dari folder **Assets** pada **Project** view, pilih **Signal** prefab. Dari **Inspector** view, akses komponen **Audio Source** dan rubah **Output** menjadi **FxMixer | Bells**:

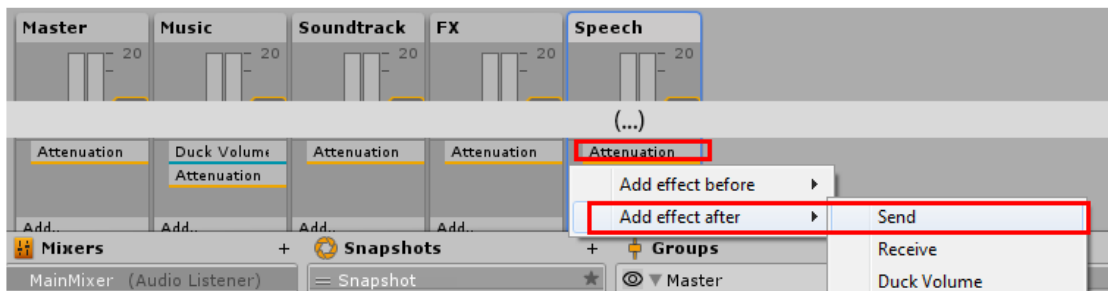


12. Buka **Audio Mixer** window. Pilih **MainMixer**, pilih kontrol **Music** track, right-click pada **Attenuation**, tambahkan **Duck Volume** effect before **Attenuation**:

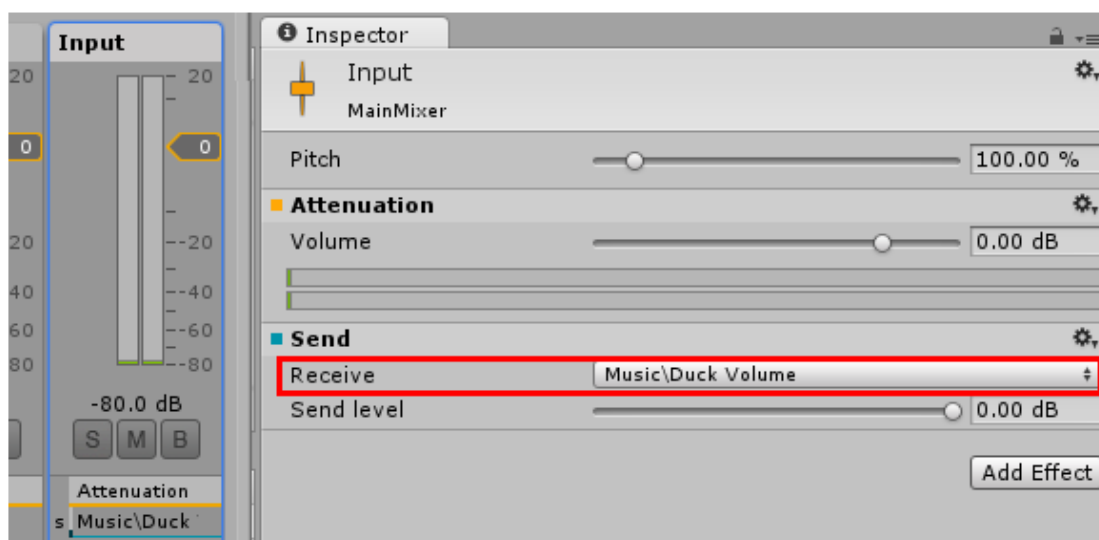


13. Pilih **Input** track, right-click pada **Attenuation**, tambahkan **Send** effect after

Attenuation:

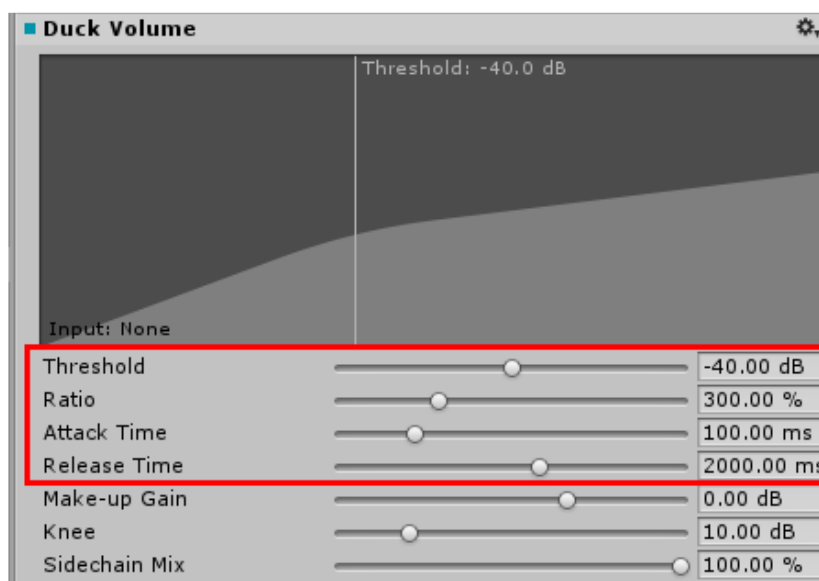


14. Dalam **Input** track, pada **Inspector** view rubah **Receive** setting dalam **Send** menjadi **Music\Duck Volume** dan rubah **Send level** menjadi 0.00 db, seperti dibawah ini:



15. Pilih **Music** track. Dari **Inspector** view, Tubah **Duck Volume** seperti berikut:

Threshold: -40.00 db; **Ratio:** 300.00 %; **Attack Time:** 100.00 ms; **Release Time:** 2000.00 ms, seperti pada screenshot dibawah ini:





16. Jalankan scene. Memasuki objek trigger akan menyebabkan volume soundtrack turun drastis, dan volume akan pulih dalam 2 detik.

F. TUGAS PRAKTIKUM

- a. Kerjakan seluruh project yang ada dimodul ini dan buatlah laporan praktikumnya. Serta berikan penjelasan pada masing-masing project apa yang kalian dapatkan setelah membuat project tersebut.
- b. Terapkan Dynamic Soundtrack pada lagu kesukaan kalian!

--- SELAMAT BELAJAR ---