Pertemuan Ke-12

Laporan

|  |  |
| --- | --- |
|  | **Praktikum 1 - Waiting for audio to finish playing before auto-destructing an object** |
|  | Buat project baru |
|  | Buatlah sebuah **GameObject** dan beri nama **AudioObject**. Kemudian tambahakan komponen **Audio** **Source** (menu Component | Audio | Audio Source). |
|  | 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** |
|  | Tambahkan C# script untuk **AudioObject** dan beri nama *AudioDestructBehaviour,* dan drag ke Audio Source |
|  | Pada Inspector view, (un-check) komponen **AudioDestructBehaviour** dari **AudioObject** |
|  | Buatlah C# Script dan beri nama *ButtonActions* |
|  | Buatlah **UI button** ganti nama dengan **PlaySoundButton** dan beri text *Play Sound* pada button, dan lampirkan **ButtonActions** script menuju button tersebut. |
|  | Pilih **PlaySoundButton** pada Hierarchy, buatlah new **on-click** event handler, drag **PlaySoundButton** kedalam slot **Object**, dan pilih **PlaySound()** function. |
|  | Pilih **PlaySoundButton** pada Hierarchy panel, drag **AudioObject** menuju **Audio Source**. Kemudian drag **AudioObject** script menuju **AudioDestructScriptedObject**, seperti pada screenshot dibawah ini |
|  | Buatlah **UI button** kedua dengan nama **DestoryWhenSoundFinishedButton**, dan beri nama *Destroy When Sound Finished* pada button, dan lampirkan **ButtonActions** script pada button tersebut. Atur posisinya agar tidak bertumpukan dengan button sebelumnya. |
|  | Pilih **DestoryWhenSoundFinishedButton** pada Hierarchy, buatlah new **on-click** event handler, drag **PlaySoundButton** kedalam slot GO, kemudian pilih **DestroyAfterSoundStops()** function. |
|  | Lakukan seperti pada button lainnya, pilih **DestoryWhenSoundFinishedButton** pada Hierarchy panel, Kemudian drag **AudioObject** script menuju **MyAudioDestructObect**. |
|  | Jalankan Game maka hasil akan seperti itu |

|  |  |
| --- | --- |
|  | **Praktikum 2 - Adding volume control with Audio Mixers** |
|  | Buat project 3D baru |
|  | Import **Volume.unitypackage** kedalam project. |
|  | 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: |
|  | * 1. Musik soundtrack yang berulang |
|  | * 1. Lonceng bordering |
|  | * 1. Suara robot ketika karakter bertabrakan dengan tembok |
|  | Dari Project view, gunakan menu **Create** untuk menambahkan **Audio Mixer** kedalam project. Ganti nama dengan *MainMixer* kemudian **double-click** untuk membuka Audio Mixer window |
|  | 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 |
|  | 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 |
|  | Pilih **MusicMixer** kemudian pilih **Master** di dalam Groups dan buatlah percabangan dengan nama *Soundtrack*. Lalu pilih **FxMixer** dan tambahakan dua percabangan pada **Master** di dalam Groups dengan nama *Speech* dan *Bells*, seperti dibawah ini |

|  |  |
| --- | --- |
|  | Dari Hierarchy view, pilih **DialogueTrigger** object. Kemudian pada Inspector view, rubah Output menjadi **FxMixer** | **Speech** pada komponen **Audio Source** |
|  | Pilih **Soundtrack** GameObject. Dari Inspector view, cari komponen **Audio Source** dan ubah Output menjadi **MusicMixer** | **Soundtrack** |
|  | Kemudian dari folder Assets pada Project view, pilih **Signal prefab**. Dari Inspector view, akses komponen **Audio Source** dan ubah Output menjadi **FxMixer** | **Bells** |
|  |  |
|  | 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 |
|  | 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* |
|  | Buatlah C# Script dan beri nama VolumeControl |
|  | Dari Hierarchy view, gunakan menu Create untuk menambah **Panel** ke dalam **scene** (Create | UI | Panel). Maka akan muncul panel dengan nama **Canvas** pada scene |
|  |  |
|  | Dari Hierarchy view, gunakan menu Create untuk menambah **Slider** ke dalam **scene** (Create | UI | Slider). Buatlah hal tersebut pada objek **Panel** sebagai percabangannya |
|  | Ganti nama **slider** tersebut menjadi *OverallSlider*. **Duplikat** **slider** tersebut dan ganti nama menjadi *MusicSlider*. Kemudian pada Inspector view, **Rect Transform** ubah parameter **Pos Y** menjadi **-40** |
|  | Duplikat **slider** **MusicSlider** dan ganti nama menjadi *FxSlider*. Kemudian rubah parameter **Pos Y** menjadi **-70** |
|  | Pilih **Canvas GameObject** dan tambahakn script **VolumeControl** kedalamnya. Kemudian ganti isi kolom **MyMixer** dari **Volume Control** dengan **MainMixer** |
|  | Pilih komponen **OverallSlider**. Dari Inspector lihat pada komponen **Slider**, ubah **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 |
|  | Ulangi langkah terakhir untuk **MusicSlider** dan **FxSlider**, tetapi pilih opsi **ChangeMusicVol** dan **ChangeFxVol** masing-masing pada menu |
|  | Jalankan scene. Anda akan dapat menggunakan sliders ketika menekan tombol Escape pada keyboard dan mengatur volume settings |

|  |  |
| --- | --- |
| **Praktikum 3 - Making a dynamic soundtrack with Snapshots** | |
|  | Buat project 3D baru |
|  | Import **DynamicSoundtrack** package dan dua file soundtrack audio clips dengan format **.ogg**: *Theme01\_Percussion* dan *Theme01\_Synths* kedalam Unity Project anda |
|  | Buka scene **Dynamic** |
|  | Dari **Project** view, gunakan menu Create untuk menambahkan **Audio Mixer** kedalam project. Ganti nama dengan *MusicMixer* kemudian double-click untuk membuka **Audio Mixer window** |
|  | 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* |
|  | Dari **Hierarchy** view, buatlah Empty **GameObject**. Beri nama *Music* kemudian tambahakan dua Empty **Child GameObjects** sebagai percabangannya. Ganti nama keduanya dengan nama *Percussion* and *Synth* |
|  | 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 i |

|  |  |
| --- | --- |
|  | Drag file audio **Theme01\_Synths** kedalam **Synths GameObject**. Dari Inspector view, rubah Output menjadi **Synths (MusicMixer)**, Pastikan opsi **Play On Awake** tercentang, cek opsi **Loop**, dan pastikan **Spatial Blend** pada **2D**. Caranya sama dengan langkah sebelumnya |
|  | 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 |
|  | Pilih **Percussion track**. Right-click **Attenuation** dan tambahkan **High-pass** pada add **effect before** |
|  | Dari Inspector view, rubah **Cutoff frequency** dari efek **High-pass** menjadi **544.00 Hz** |
|  | Dari Snaphots 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 |
|  | Pilih **Tunnel** **snapshot**. Kemudian dari Inspector view, rubah **Cutoff frequency** pada efek Highpass menjadi **10.00** Hz |
|  | Ganti antara **Tunnel** dan **Start** snapshots. Anda akan mendengarkan perubahannya. |
|  | Duplikat **Tunnel** snapshot, ganti dengan nama *OtherSide*, dan klik pada sanpshot itu. |
|  | Naikkan volume dari **Synths** track sampai dengan 0 dB |
|  | 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). |
|  | Pilih **Cube** dan ganti dengan nama *SnapshotTriggerTunnel*. Kemudian dari Inspector view, akses komponen **Box Collider** dan centang opsi **Is Trigger**, seperti screenshot dibawah ini. Lalu, uncheck komponen **Mesh Renderer**. Lalu sesuaikan ukuran dan posisi pada interior milik tunnel scene |
|  | 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 |
|  |  |

|  |  |
| --- | --- |
|  | Pada **Project** view, Buatlah C# Script dan beri nama *SnapshotTrigger.* Simpan script anda lalu lampirkan pada **SnapshotTriggerTunnel**, **SnapshotTriggerStart**, and **SnapshotTriggerOtherSide** objects |
|  | Pilih **SnapshotTriggerTunnel**. Kemudian dari Inspector view, akses komponen **Snapshot Trigger**, atur **Snapshot** menjadi **Tunnel**, dan **Crossfade** menjadi **2**, seperti screenshot dibawah ini |

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

|  |  |
| --- | --- |
|  | **Praktikum 4 - Balancing in-game audio with Ducking** |
|  | Buat project 3D baru. Import **Ducking.unitypackage** and **soundtrack.mp3** kedalam project anda. |
|  | 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 |
|  | Dari menu Create yang berada di atas Hierarchy view, pilih **Create Empty** untuk menambahkan **GameObject** baru pada scene. Ganti dengan nama *Soundtrack*. |
|  | 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 |
|  | Dari **Project** view, gunakan menu Create untuk menambahkan **Audio Mixer** kedalam project. Ganti dengan nama *MainMixer* kemudian **double-click** untuk membuka Audio Mixer window |
|  | 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 |
|  | 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 |
|  | Pilih **MusicMixer** kemudian pilih **Master** di dalam **Groups** dan buatlah percabangan dengan nama *Soundtrack*. Lalu pilih **FxMixer** dan tambahakan percabangan pada **Master** di dalam **Groups** dengan nama *Bells*, seperti dibawah ini |
|  | Dari **Hierarchy** view, pilih **DialogueTrigger** object. Kemudian pada **Inspector** view, rubah Output menjadi **MainMixer | Input** pada komponen **Audio Source** |
|  | Pilih **Soundtrack** GameObject. Dari Inspector view, cari komponen **Audio Source** dan rubah Output menjadi **MusicMixer | Soundtrack** |
|  | Kemudian dari folder Assets pada **Project** view, pilih **Signal prefab**. Dari **Inspector** view, akses komponen **Audio Source** dan rubah Output menjadi **FxMixer | Bells** |
|  | Buka **Audio Mixer** window. Pilih **MainMixer**, pilih kontrol **Music** track, right-click pada **Attenuation**, tambahkan **Duck Volume** effect before **Attenuation** |
|  | Pilih **Input** track, right-click pada **Attenuation**, tambahakan **Send** effect after **Attenuation** |
|  | 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 |
|  | Pilih **Music** track. Dari Inspector view, Rubah **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 |
|  | Jalankan scene. Memasuki objek trigger akan menyebabkan volume soundtrack turun drastis, dan volume akan pulih dalam 2 detik |

|  |  |
| --- | --- |
| **Tugas - Making a dynamic soundtrack** | |
|  | Buat project 3D baru |
|  | Import **DynamicSoundtrack** package dan dua file soundtrack audio clips dengan format **.ogg**: *Theme01\_Percussion* dan *Theme01\_Synths* kedalam Unity Project anda |
|  | Buka scene **Dynamic** |
|  | Dari **Project** view, gunakan menu Create untuk menambahkan **Audio Mixer** kedalam project. Ganti nama dengan *MusicMixer* kemudian double-click untuk membuka **Audio Mixer window** |
|  | 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 *Genshin* dan *Woah* |
|  | Dari **Hierarchy** view, buatlah Empty **GameObject**. Beri nama *Music* kemudian tambahakan dua Empty **Child GameObjects** sebagai percabangannya. Ganti nama keduanya dengan nama *Genshin* dan *Woah* |
|  | Dari **Project** view, drag **Audio Clip** dengan nama *Theme01\_Percussion* kedalam **Genshin** **GameObject** di Hierarchy. pilih **Genshin** dan pada Inspector view, akses komponen **Audio Source**. Rubah Output menjadi **Genshin** (MusicMixer), pastikan opsi **Play On Awake** ter-centang, cek opsi **Loop**, dan pastikan **Spatial Blend** pada **2D**, seperti pada screenshot dibawah i |

|  |  |
| --- | --- |
|  | Drag file audio **XXXTentacion-Whoah** kedalam **Woah GameObject**. Dari Inspector view, rubah Output menjadi **Whoah (MusicMixer)**, Pastikan opsi **Play On Awake** tercentang, cek opsi **Loop**, dan pastikan **Spatial Blend** pada **2D**. Caranya sama dengan langkah sebelumnya |
|  | 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 |
|  | Pilih **Percussion track**. Right-click **Attenuation** dan tambahkan **High-pass** pada add **effect before** |
|  | Dari Inspector view, rubah **Cutoff frequency** dari efek **High-pass** menjadi **544.00 Hz** |
|  | Dari Snaphots 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 |
|  | Pilih **Tunnel** **snapshot**. Kemudian dari Inspector view, rubah **Cutoff frequency** pada efek Highpass menjadi **10.00** Hz |
|  | Ganti antara **Tunnel** dan **Start** snapshots. Anda akan mendengarkan perubahannya. |
|  | Duplikat **Tunnel** snapshot, ganti dengan nama *OtherSide*, dan klik pada sanpshot itu. |
|  | Naikkan volume dari **Whoah** track sampai dengan 0 dB dan **Genshin** track sampai -80 db |
|  | 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). |
|  | Pilih **Cube** dan ganti dengan nama *SnapshotTriggerTunnel*. Kemudian dari Inspector view, akses komponen **Box Collider** dan centang opsi **Is Trigger**, seperti screenshot dibawah ini. Lalu, uncheck komponen **Mesh Renderer**. Lalu sesuaikan ukuran dan posisi pada interior milik tunnel scene |
|  | 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 |
|  |  |

|  |  |
| --- | --- |
|  | Pada **Project** view, Buatlah C# Script dan beri nama *SnapshotTrigger.* Simpan script anda lalu lampirkan pada **SnapshotTriggerTunnel**, **SnapshotTriggerStart**, and **SnapshotTriggerOtherSide** objects |
|  | Pilih **SnapshotTriggerTunnel**. Kemudian dari Inspector view, akses komponen **Snapshot Trigger**, atur **Snapshot** menjadi **Tunnel**, dan **Crossfade** menjadi **2**, seperti screenshot dibawah ini |

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

|  |  |
| --- | --- |
| No | KODE PEMROGRAMAN |
| 1 | Snapshoot Trigger  using UnityEngine;  using UnityEngine.Audio;  using System.Collections;  public class SnapshotTrigger : MonoBehaviour {  public AudioMixerSnapshot snapshot;  public float crossfade;  private void OnTriggerEnter(Collider other) {  snapshot.TransitionTo(crossfade);  }  } |
| 2 | Volume Control  using System.Collections;  using System.Collections.Generic;  using UnityEngine.Audio;  using UnityEngine;  public class VolumeControl : MonoBehaviour  {  public AudioMixer 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);  }  } |
| 3 | ButtonAudioDestruct  using System.Collections;  using System.Collections.Generic;  using UnityEngine;  public class AudioDestructBehaviour : MonoBehaviour  {  private AudioSource audioSource;  void Start()  {  audioSource = GetComponent<AudioSource>();  }  private void Update()  {  if (!audioSource.isPlaying)  Destroy(gameObject);  }  } |
| 4 | ButtonActions  using System.Collections;  using System.Collections.Generic;  using UnityEngine;  public class ButtonActions : MonoBehaviour  {  public AudioSource audioSource;  public AudioDestructBehaviour audioDestructScriptedObject;  public void PlaySound()  {  if (!audioSource.isPlaying)  audioSource.Play();  }  public void DestroyAfterSoundStops()  {  audioDestructScriptedObject.enabled = true;  }  } |

|  |  |
| --- | --- |
| KESIMPULAN | |
| Kita dapat memasukkan Suara atau Sound ke dalam game dan Sound tersebut akan berbunyi jika di suatu kondisi tertentu | |
| SWAFOTO MAHASISWA+PROJECT |  |
|  | |

Format laporan:

NIM : 1841720182

Nama : Subhan Indra Prayoga

Kelas : TI-3H

Swafoto : Di Atas

Deskripsi Game :

1. Nama Game : Insert Sound in Game
2. Alur Game (print screen dan : Pada modul ini kita belajar menambahkan Sound dalam

penjelasan) game

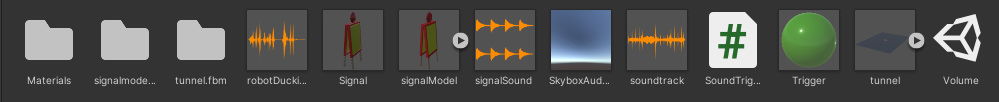
1. Komponen materi yang dipakai : 1. Waiting for audio to finish playing before auto-

destructing an object

* 1. Adding volume control with Audio Mixers
  2. Making a dynamic soundtrack with Snapshots
  3. Balancing in-game audio with Ducking

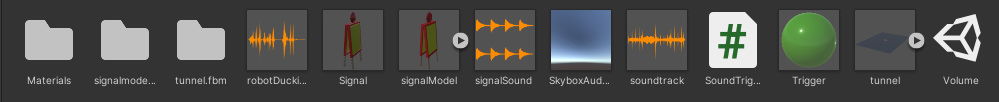
1.  Asset yang dipakai (sertakan :

screenshoot, jika terdapat asset

yang digunakan berasal dari

internet atau sumber lain,

cantumkan link)



Link Github : https://github.com/SubHans/Komputasi\_Multimedia\_Sem6