

LAPORAN

TUGAS KELOMPOK

INTERAKSI MANUSIA DAN KOMPUTER

SEMESTER GENAP 2018 – 2019



Disusun oleh:

Rock Bottom/G-Cube – Kelas A – Angkatan 2016

Ketua: Muhammad Rifqy Aulia Akbar – 140810160055

Anggota:

- Muhammad Raihan Akbar – 140810160013
- Patricia Joanne – 140181060065

Program Studi S-1 Teknik Informatika

Departemen Ilmu Komputer

Fakultas Matematika & Ilmu Pengetahuan Alam

Universitas Padjadjaran

2019

DAFTAR ISI

DAFTAR ISI.....	2
KATA PENGANTAR	3
1. RANCANGAN APLIKASI.....	4
1.1. Analisis Kebutuhan	4
1.1.1. Analisis Target Pengguna	4
1.1.2. Analisis Kebutuhan Perangkat Lunak.....	4
1.1.3. Analisis Kebutuhan Perangkat Keras.....	4
2. RANCANGAN UMUM APLIKASI	5
2.1. Storyboard Aplikasi VR.....	5
3. PEMBUATAN ASET	6
4. PEMBUATAN LINGKUNGAN SIMULASI	9
5. PENGATURAN JOYSTICK/CONTROLLER	12
6. PEMBUATAN HALAMAN MENU SIMULASI.....	13
6.1. Pemasangan Prefab GoogleVR dan Penyimpanan Objek 3D.....	13
6.2. Pembuatan Tombol Menu dan Petunjuk Simulasi	13
6.3. Pembuatan Tombol Tingkat (Level) Simulasi	18
7. PEMBUATAN HALAMAN SIMULASI.....	20
8. PENGUJIAN.....	21
9. HIGHLIGHT	23
BIODATA SINGKAT ANGGOTA TIM.....	27

KATA PENGANTAR

Puji syukur ke hadirat Tuhan Yang Maha Esa atas segala rahmatNya sehingga laporan proyek mata kuliah Interaksi Manusia dan Komputer ini dapat tersusun hingga selesai. Tidak lupa kami juga mengucapkan banyak terima kasih atas bantuan dari pihak yang telah berkontribusi dengan memberikan sumbangan baik materi maupun pikirannya baik dari teman, keluarga, maupun dosen yang kami hormati Bapak Erick.

Harapan kami semoga laporan ini dapat menambah pengetahuan dan pengalaman bagi para pembaca dan untuk ke depannya dapat memperbaiki bentuk maupun menambah isi laporan agar menjadi lebih baik lagi.

Karena keterbatasan pengetahuan maupun pengalaman kami, kami yakin masih banyak kekurangan dalam laporan ini. Oleh karena itu, kami sangat mengharapkan saran dan kritik yang membangun dari pembaca demi kesempurnaan laporan ini.

Jatinangor, Juni 2019

Penyusun

1. RANCANGAN APLIKASI

1.1. Analisis Kebutuhan

1.1.1. Analisis Target Pengguna

Target pengguna permainan VR “G-Cube” terdiri dari berbagai kalangan baik anak-anak, remaja, hingga dewasa yang ingin melatih kemampuan mengingat, berpikir luas, dan juga ingin meringankan pikiran melalui permainan.

1.1.2. Analisis Kebutuhan Perangkat Lunak

Dalam pembuatan permainan VR ini dibutuhkan *software* atau perangkat lunak yang terdiri dari:

- Unity3D v2018.3.12
- Blender v27.9
- Autodesk Maya 3D 2018
- Cinema 4D R20 2018

1.1.3. Analisis Kebutuhan Perangkat Keras

Permainan VR ini dibuat dengan perangkat keras dengan spesifikasi sebagai berikut:

- OS: Windows 10
- RAM: 8 GB
- Harddisk: 1 TB
- Graphic: Nvidia GeForce

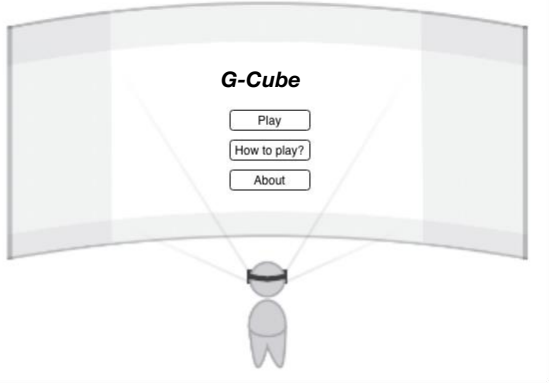
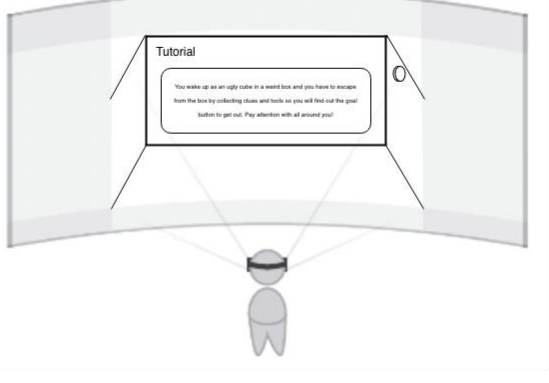
Sementara untuk spesifikasi minimum perangkat yang dapat digunakan untuk memainkan permainan VR ini adalah:

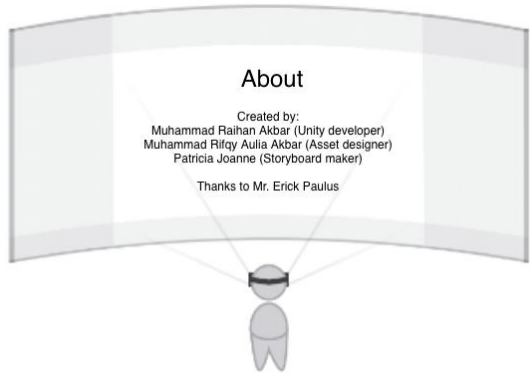
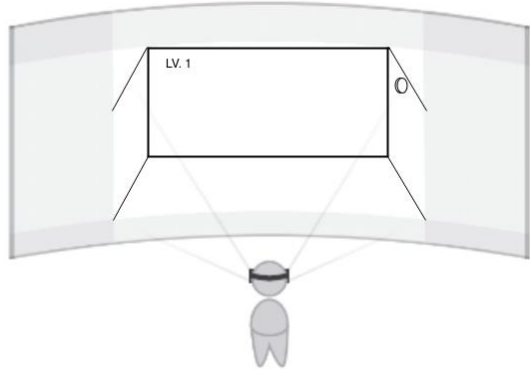
- OS: Android 6.0 (Marshmallow)
- RAM: 2 GB
- Memory: 8 GB
- Resolusi: 720p
- Ukuran layar: 5” (inch)
- Spesifikasi tambahan: sensor *gyroscope* dan *accelerometer*

2. RANCANGAN UMUM APLIKASI

“G-Cube” adalah sebuah permainan VR berbasis level yang bertujuan untuk melatih kemampuan mengingat, berpikir luas, dan juga ingin meringankan pikiran melalui permainan. Melalui permainan ini diharapkan penggunaanya dapat memiliki daya perspektif yang tinggi. Nama “G-Cube” diambil dari kata *gecko-cube* dimana *gecko* sendiri berarti cicak. “G-Cube” terdiri dari beberapa level namun sebelum langsung mencoba level-level tersebut, pengguna harus menyelesaikan tutorial terlebih dahulu. Tutorial dapat diakses melalui halaman utama atau menu dari permainan. Selain tutorial, pengguna juga dapat memulai permainan langsung tanpa mengikuti tutorial dengan mengklik *Play* atau melihat halaman *About* yang berisi nama tim pengembang dan mentor.

2.1. Storyboard Aplikasi VR

Ilustrasi <i>storyboard</i>	Deskripsi <i>storyboard</i>
	<p><u>Halaman utama: menu</u></p> <p>Pada halaman ini, pengguna dapat melihat tombol “<i>play</i>”, “<i>how to play</i>”, dan “<i>about</i>”.</p>
	<p><u>Halaman tutorial</u></p> <p>Pada halaman ini, pengguna dapat menguji coba permainan ini dalam mode tutorial sehingga paham dengan cara bermain permainan ini. Pengguna dapat melakukan berbagai interaksi-interaksi utama dapat permainan. Setelah tutorial selesai, maka pengguna akan diarahkan ke halaman permainan atau bisa juga keluar ke halaman <i>menu</i>.</p>

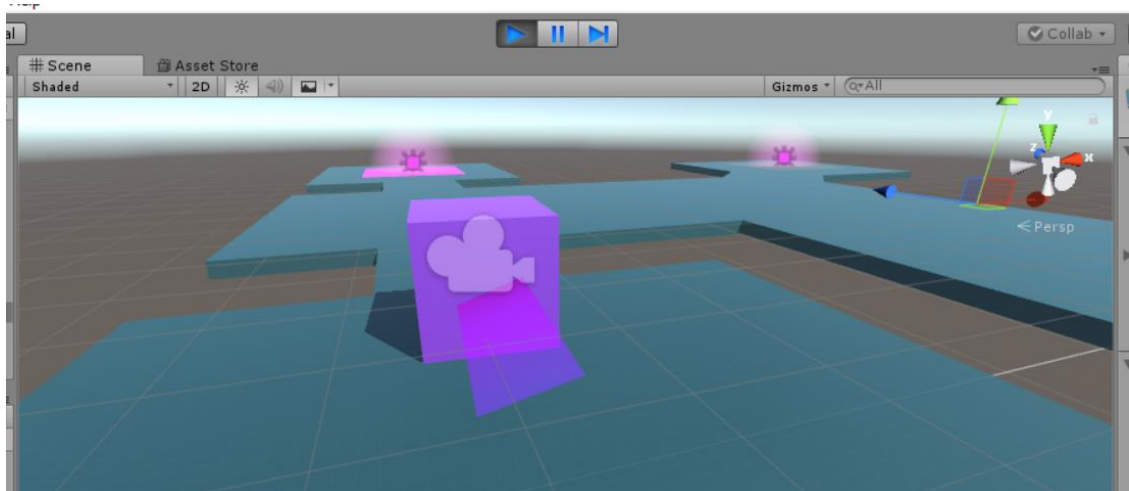
	<p><u>Halaman <i>about</i></u></p> <p>Pada halaman ini, pengguna dapat melihat nama-nama anggota tim pengembang aplikasi dan juga dosen pembimbing.</p>
	<p><u>Halaman permainan</u></p> <p>Pada halaman ini, pengguna berada dalam permainan. Di sinilah interaksi-interaksi utama dilakukan seperti: berjalan, berputar, melihat, menekan tombol, menggerakkan tuas, dan lain-lain. Permainan ini berupa tingkatan level.</p>

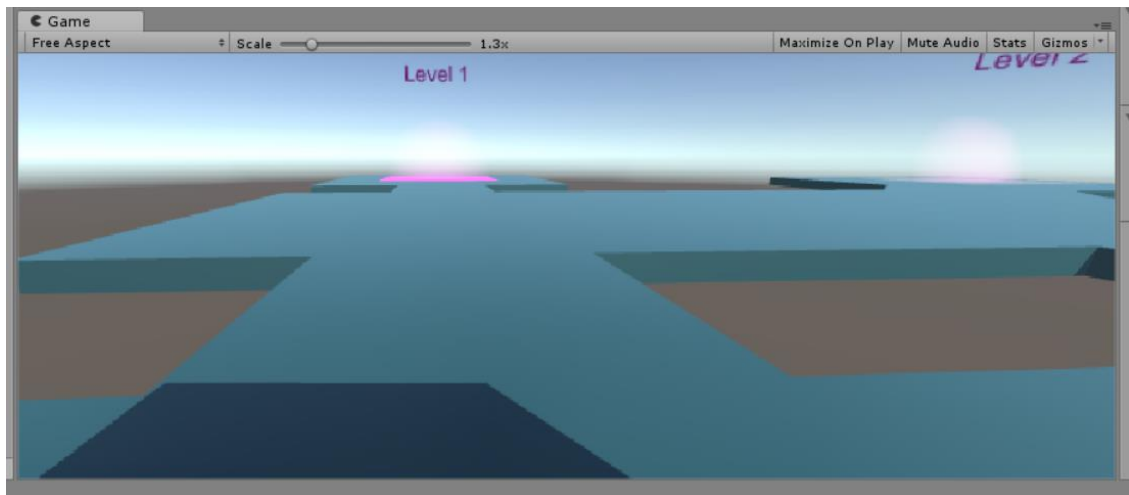
3. PEMBUATAN ASET

Asset sebagian besar dibuat menggunakan Blender dan Cinema 4D. Adapula beberapa asset *open source* yang diambil dari internet.

Berikut ini adalah daftar *asset* yang dibutuhkan:

- *Player*





Player dalam permainan ini adalah sebuah kubus dimana kubus ini di-generate menggunakan fitur yang ada pada Unity dan diberi warna ungu, lalu *player* diberikan *trail* supaya lebih indah dilihat.

- *Level design*

Dalam pengaplikasiannya *level design* dijadikan tantangan, dimana *player* harus membiasakan diri untuk menyelesaikan level, langkah pembuatan level adalah sebagai berikut:

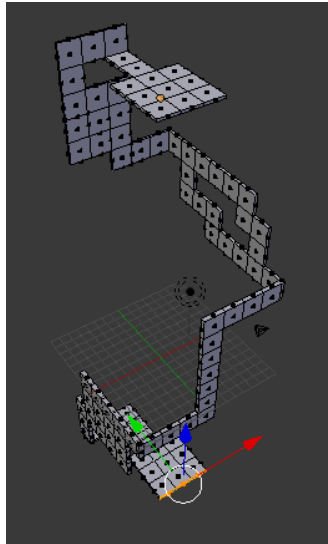
1. Gambar di kertas

Level didesain terlebih dahulu di kertas agar tahu bagaimana gambaran level yang akan dibentuk.



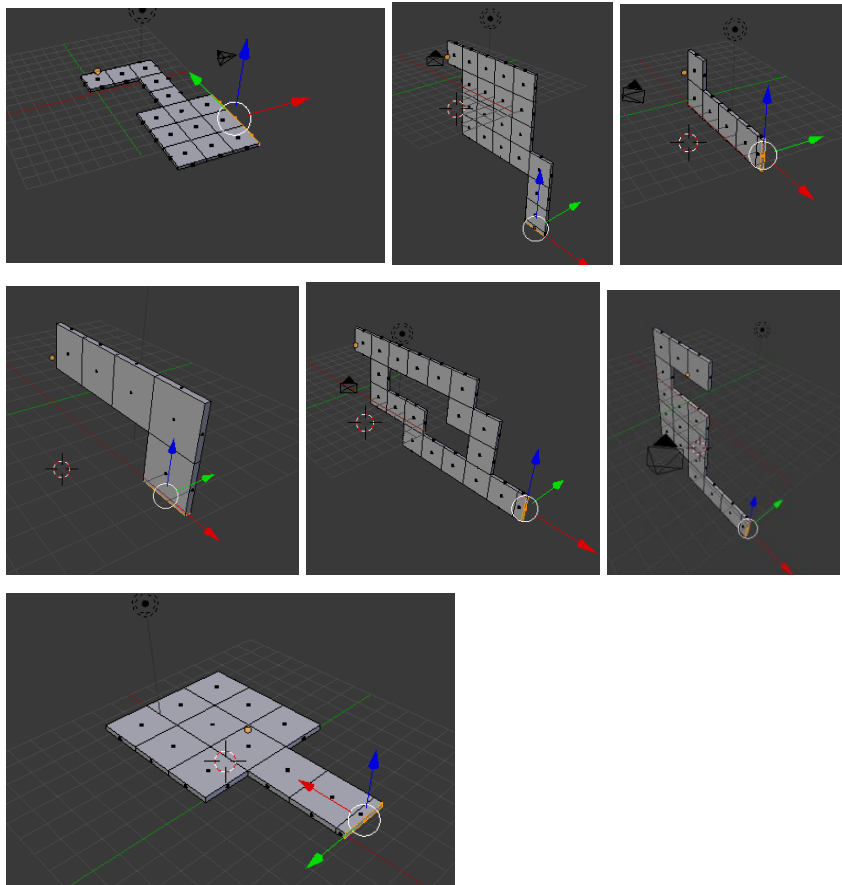
2. Pemodelan

Model dibuat secara utuh di Blender, untuk teknik pembuatannya hanya perlu menggunakan *tools extrude* dalam Blender.



3. Memisahkan

Dikarenakan program yang digunakan merubah arah gravitasi sesuai *plane* maka hasil model di pisah-pisah menjadi beberapa bagian .



- *Blocker*

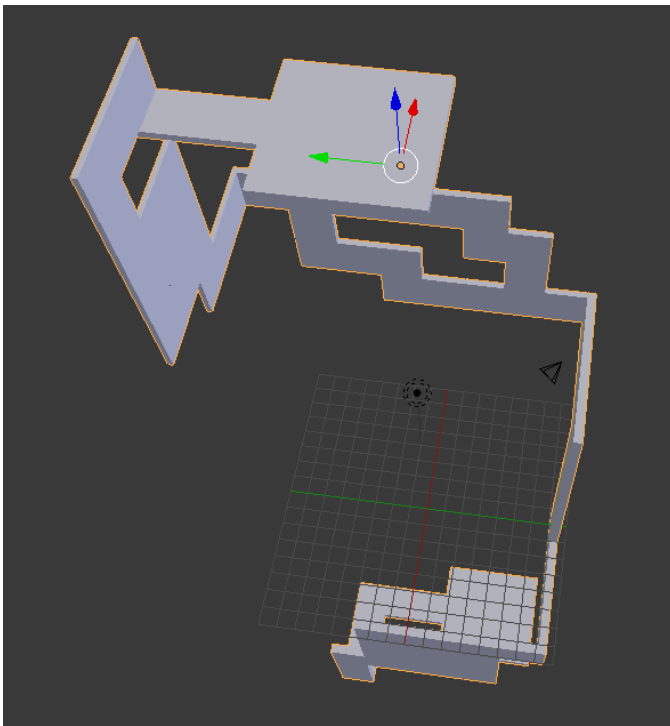
Blocker diambil dari situs archive3d.net.



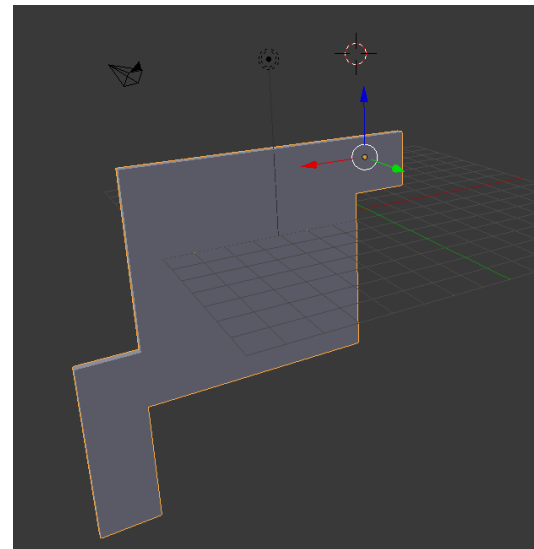
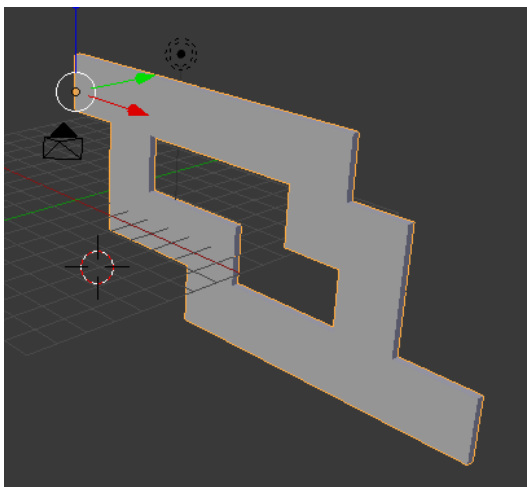
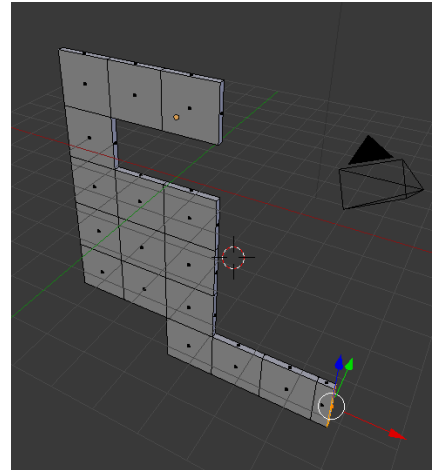
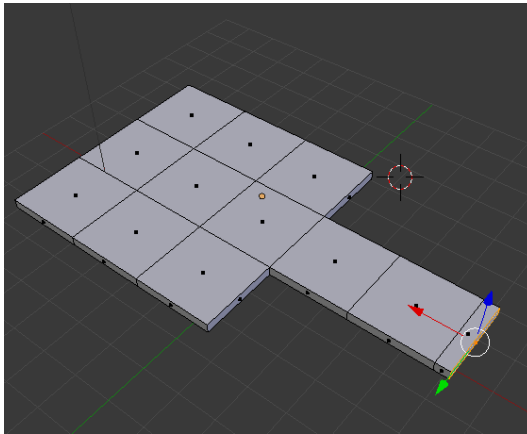
- Audio
 - Finish Trumpet Sound (DOTA TI Chat Wheel 2018)
 - Fae – Meydän (<http://freemusicarchive.org/music/Meydan/Fae/Fae>)
 - Effemeah Weeps – Uncan ([http://freemusicarchive.org/music/Uncanny/~Effemeah Weeps](http://freemusicarchive.org/music/Uncanny/~Effemeah%20Weeps))

4. PEMBUATAN LINGKUNGAN SIMULASI

- 1) Buatlah asset desain level di aplikasi 3D modeler seperti Blender atau Cinema4D.

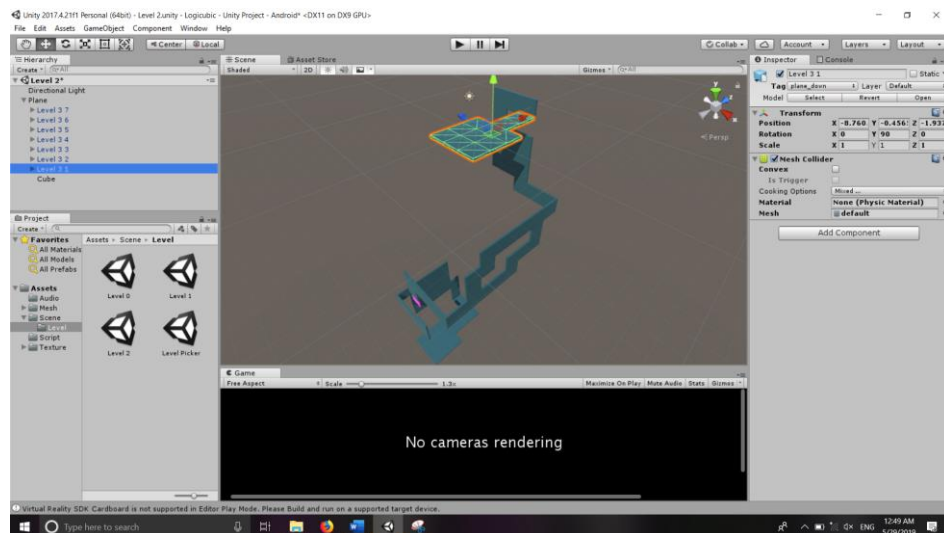


Namun perlu diingat bahwa bidang setiap level harus dipisahkan berdasarkan orientasi arah gravitasinya sehingga diperlukan satu model untuk bidang yang arah gravitasinya ke bawah (normal), satu model untuk yang mengarah ke depan, ke kanan, dan seterusnya.

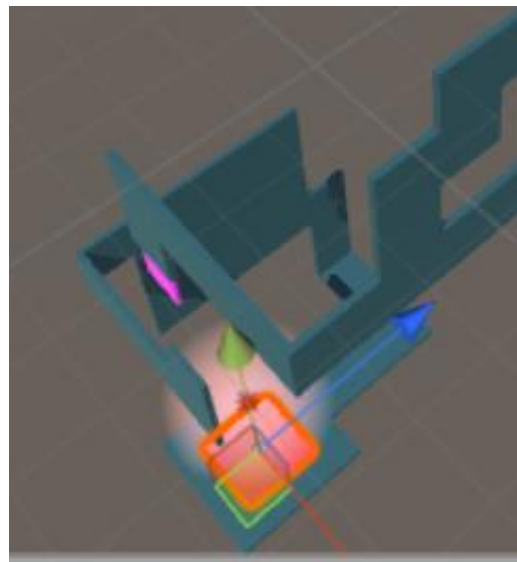
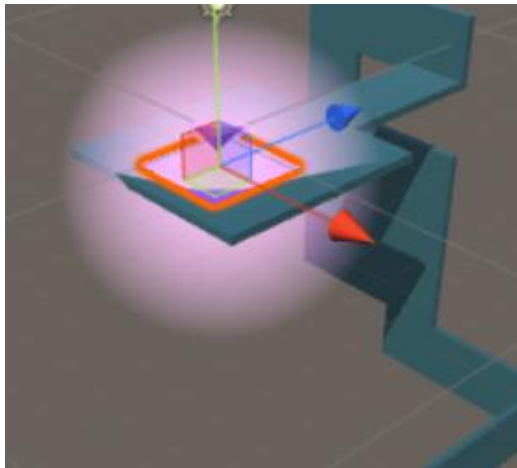
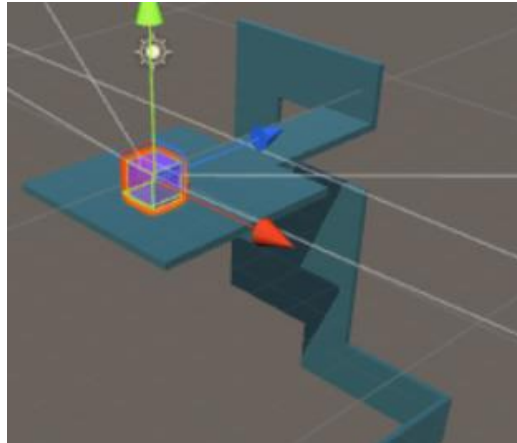


Desain level juga dapat dibuat langsung di Unity dengan menambah manual tiap bidang levelnya dengan *Create > 3D object > Cube*.

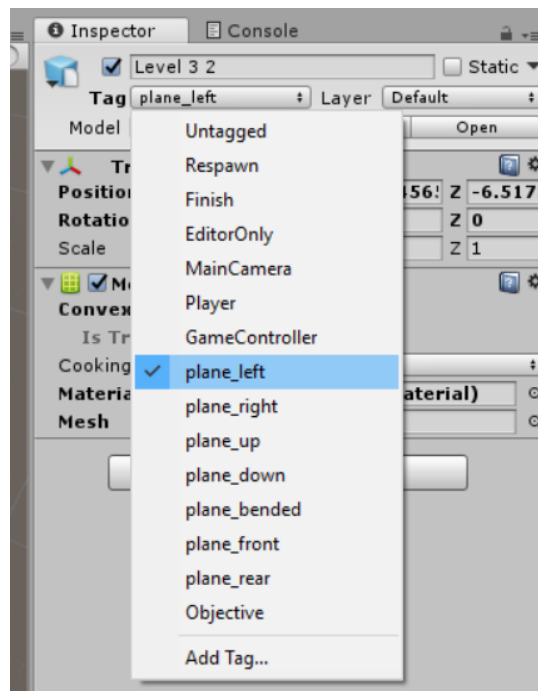
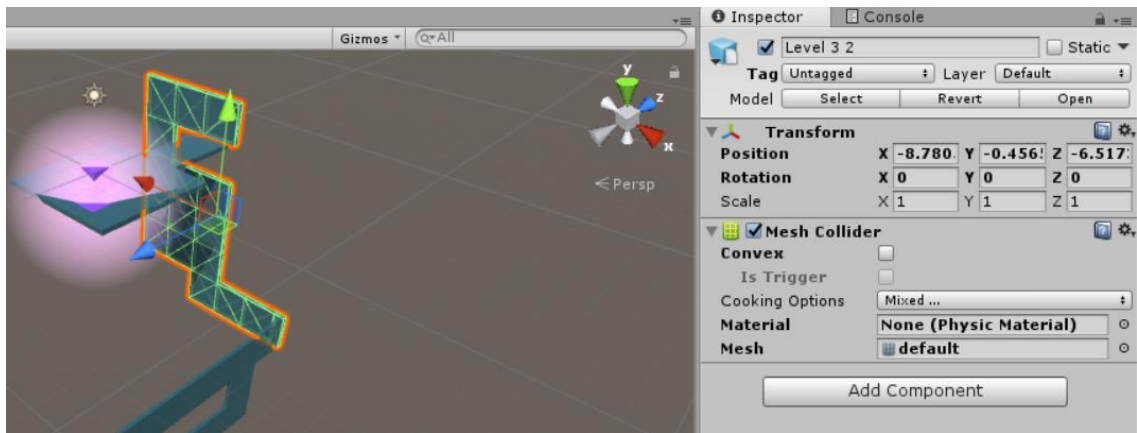
2) Rangkaian potongan bidang level tadi di dalam Unity.



- 3) Copy-paste objek “Cube Player” sebagai playernya, “Start” untuk indikasi tempat mulainya, dan “Finish” untuk indikasi tempat tujuannya.



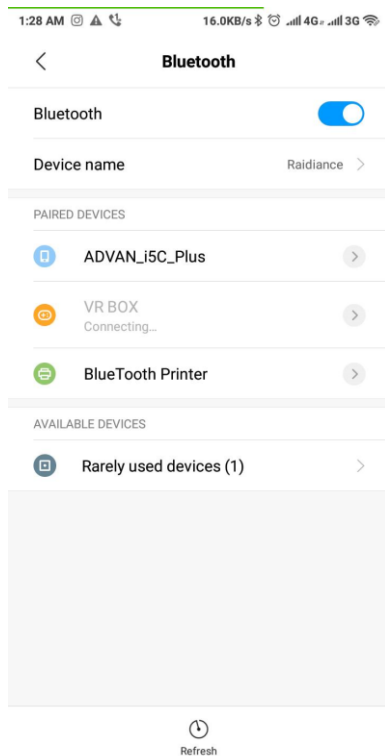
- 4) Beri tag setiap bidangnya sesuai arah gravitasinya. Jika suatu bidang dikehendaki untuk memiliki arah gravitasi ke kiri, maka pilihlah tag *plane_left*.



5) Pengaturan untuk satu lingkungan simulasi telah selesai. Level tersebut sudah dapat dicoba langsung.

5. PENGATURAN JOYSTICK/CONTROLLER

Untuk memainkan “G-Cube” dibutuhkan perangkat seperti VR Headgear dan Bluetooth VR Controller. VR Headgear digunakan dengan cara memasang *smartphone* ke dalamnya. Sementara untuk Bluetooth VR Controller, cara menggunakannya hanya perlu menyalakan tombol *power on* lalu sambungkan dengan perangkat *smartphone* yang telah dipasang tadi melalui jaringan *bluetooth*. Setelah itu *controller* sudah dapat berfungsi dan dapat digunakan selama permainan.



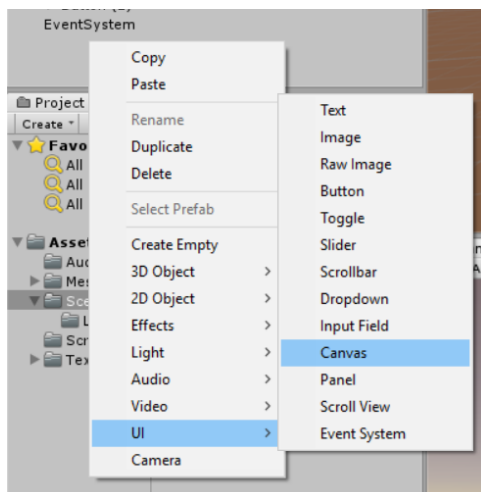
6. PEMBUATAN HALAMAN MENU SIMULASI

6.1. Pemasangan Prefab GoogleVR dan Penyimpanan Objek 3D

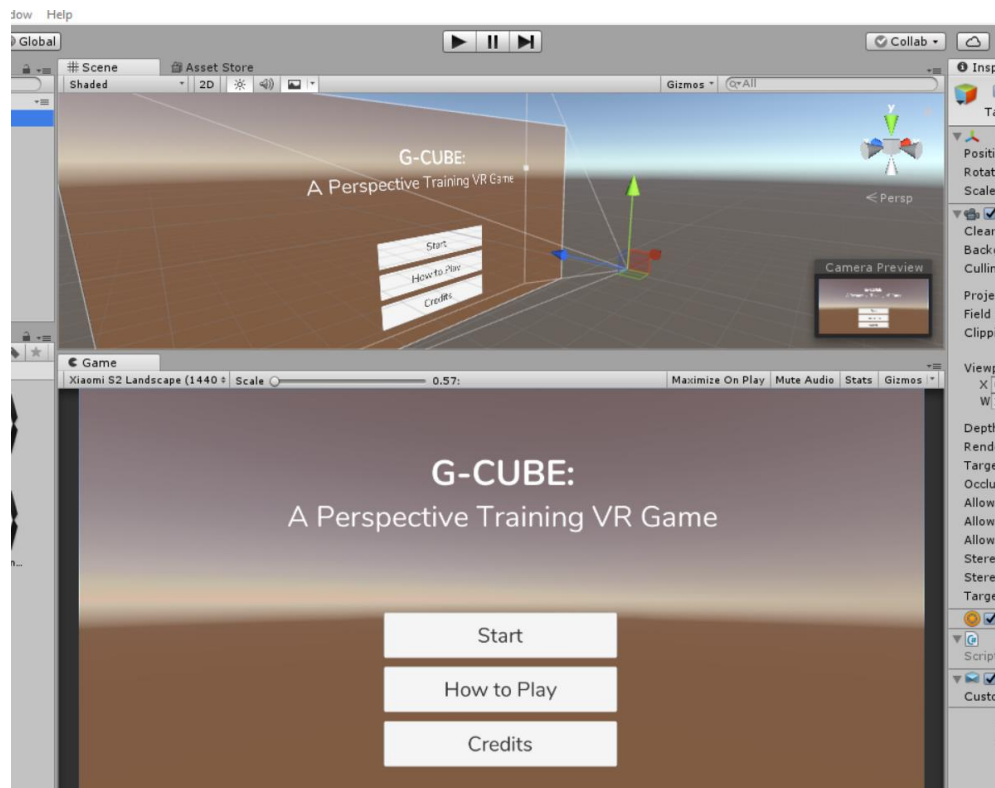
Kami tidak menggunakan Prefab GoogleVR untuk project G-Cube ini. Objek 3D disimpan di folder berbeda untuk orisinalitas baru setelahnya diimport ke dalam Unity dalam folder Mesh.

6.2. Pembuatan Tombol Menu dan Petunjuk Simulasi

Menu G-Cube dibuat manual di dalam Unity. Caranya buat objek *Canvas* terlebih dahulu sebagai wadah elemen UI. Klik kanan di *Project Hierarchy* > *UI* > *Canvas*



Tambahkan *Panel*, *Text*, dan *Button*. Lalu atur posisi tiap objeknya agar terlihat *Main Camera*. Terakhir, kustomisasi elemen setiap objeknya agar mendapat hasil sesuai yang diinginkan.



Agar tombol-tombol itu bekerja sebagai mana fungsinya, harus diberikan script agar dapat pindah scene. Buat script baru dengan kode sebagai berikut

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine.SceneManagement;
using UnityEngine;

public class ButtonSceneChanger : MonoBehaviour {

    int isCompleteTutorial;

    // Use this for initialization
    public void SceneChange (string sceneName){
        SceneManager.LoadScene(sceneName);
    }

    public void StartGameScene (){
```

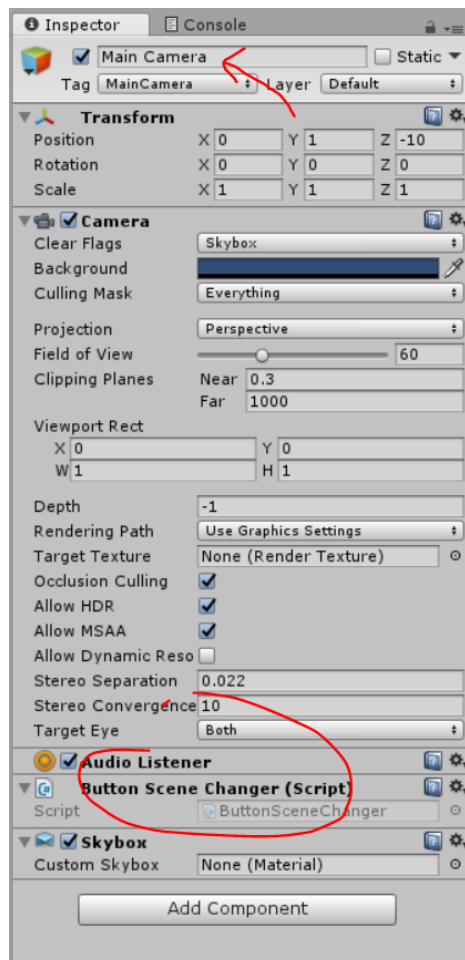
```

// is the player have completed the tutorial?
isCompleteTutorial = PlayerPrefs.GetInt("isCompleteTutorial", 0);
Debug.Log(isCompleteTutorial);

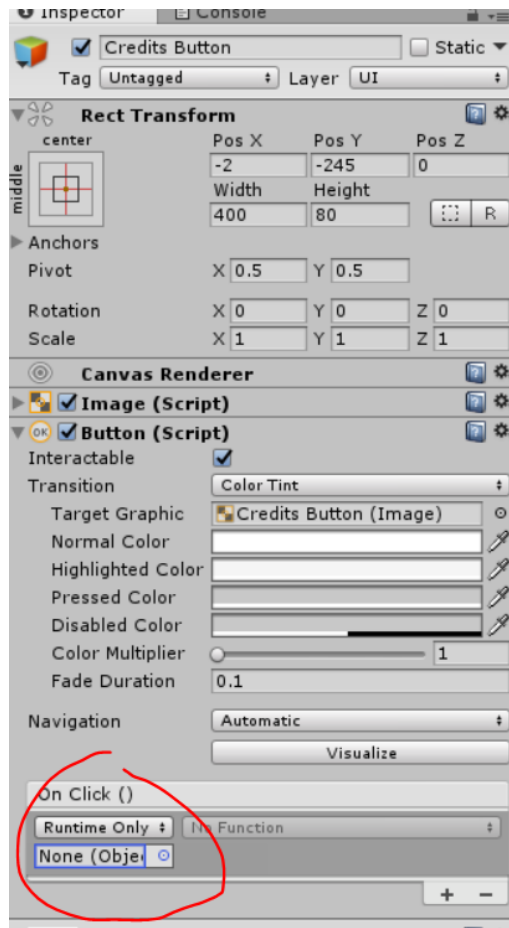
// if not yet, go to tutorial
if(isCompleteTutorial == 0)
    SceneManager.LoadScene("Level 0");
// if yes, go to level picker
else
    SceneManager.LoadScene("Level Picker");
}
}

```

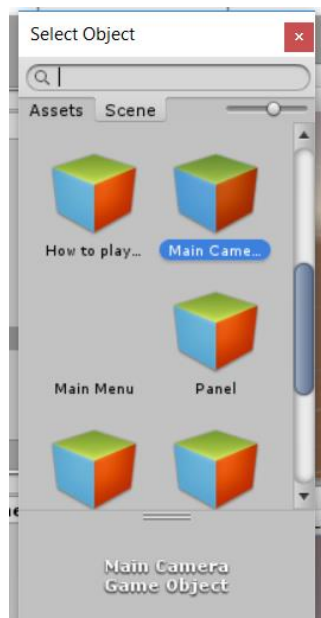
Lalu sematkan script ButtonSceneChanger ke dalam *Main Camera*.



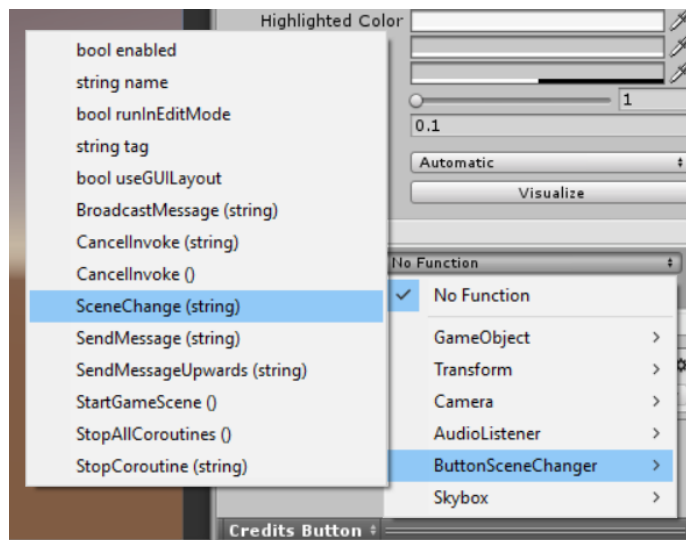
Untuk memanggil fungsi SceneChange caranya dengan memanggil fungsinya dari onClick *buttonnya*.



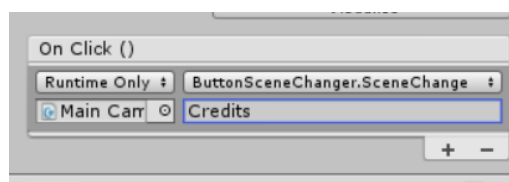
Klik lingkaran biru di sebelah kanan None (Object) lalu pilih *Main Camera* karena didalam objek *Main Camera* lah script *ButtonSceneChanger* disematkan.



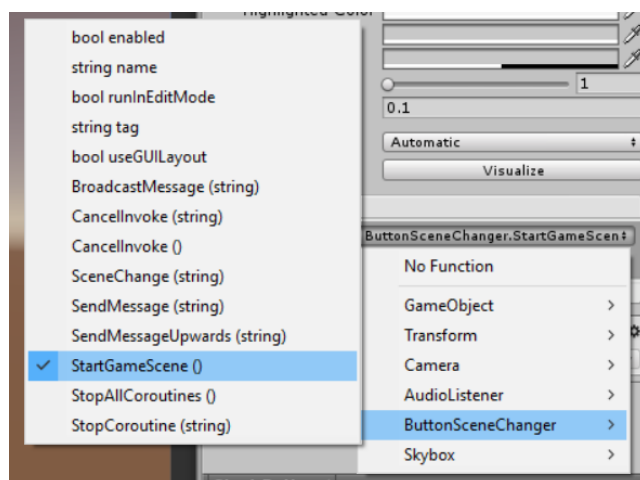
Lalu pilih tab yang bertuliskan *No Function*, pilih *ButtonSceneChanger* (Script untuk pindah scene), Lalu pilih *SceneChange* (string).



Terakhir isi tujuan Scene yang mau dituju dalam kotak di sampingnya. Misal untuk Credits Button, kita ingin agar setelah tombol ini ditekan akan pindah ke scene Credits.

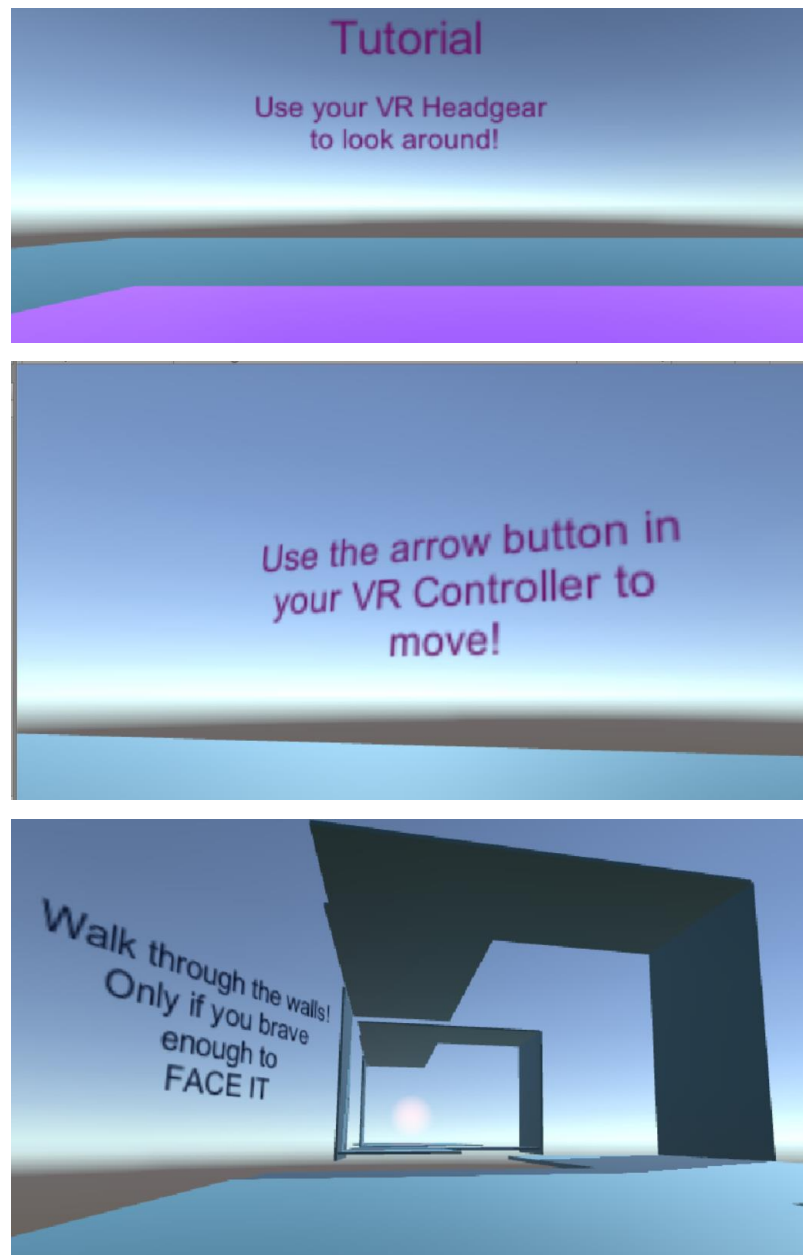


Lakukan yang sama untuk tombol How to Play Button yang harus pindah ke scene level 0 atau level tutorial. Khusus tombol Start Button, pilih fungsi yang akan dipanggil ketika `OnClick StartGameScene()`.



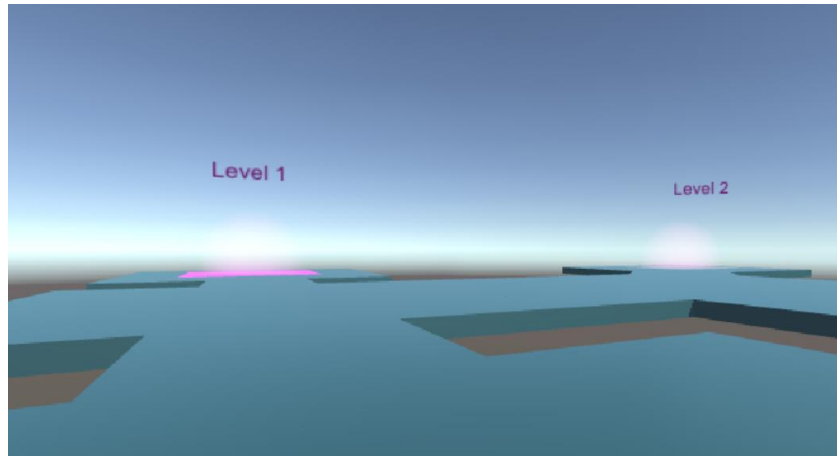
Fungsi `StartGameScene()` adalah agar pemain akan dibawa ke level 0 untuk memulai tutorial terlebih dahulu. Jika dan hanya jika pemain sudah menyelesaikan Level tutorial, barulah ketika ditekan tombol Start akan dibawa ke *Level Picker*, tempat dimana memilih level untuk dimainkan dilakukan.

Pada saat memulai permainan G-Cube, pemain akan langsung memasuki level tutorial. Pemain akan melihat petunjuk tentang bagaimana cara melihat dunia virtual dalam G-Cube serta kendali atas kontrol gerakannya.



6.3. Pembuatan Tombol Tingkat (Level) Simulasi

Setelah level tutorial diselesaikan, pemain akan diarahkan ke *level picker*, tempat dimana pemain dapat memilih level permainan. Di sini terdapat portal yang akan memindahkan pemain ke level yang dimaksud.

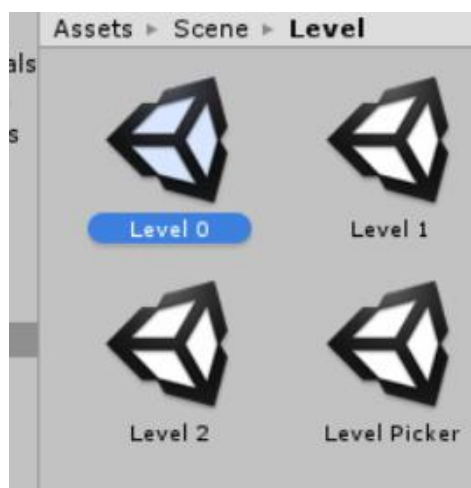


Level baru akan terbuka seiring dengan progress pemain memainkan G-Cube. Cara membuat “portal” yang memindahkan ke level lain adalah dengan menggunakan `UnityEngine.SceneManagement`.

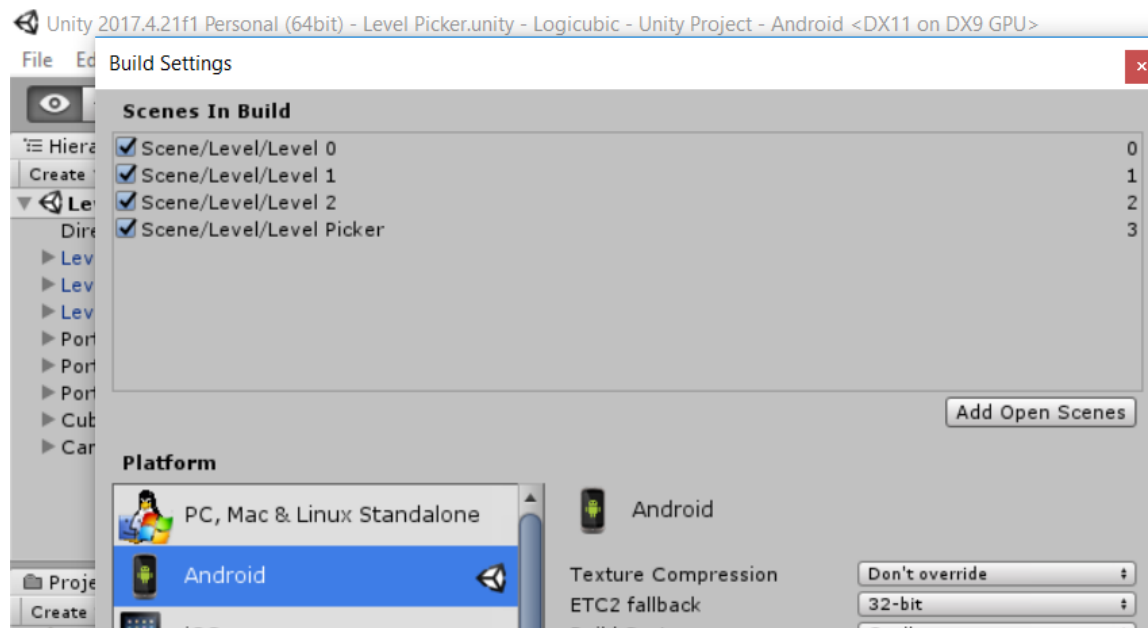
```
void OnCollisionEnter (Collision collisionInfo){
    if (collisionInfo.collider.tag == "Player"){
        SceneManager.LoadScene(targetLevel);
    }
}
```

Sumber: <https://docs.unity3d.com/ScriptReference/SceneManager.SceneManager.LoadScene.html>

Code di atas akan disematkan ke dalam objek portal tersebut. Code itu jika diterjemahkan menjadi “*Jika portal bersentuhan dengan player, maka pindah ke level **targetLevel***”. `targetLevel` yang dimaksud di sini adalah nama *scene* yang dideklarasikan dengan tipe data variabel string.



Untuk menggunakan loadScene, scene yang bersangkutan harus terdaftar di **Scenes in Build**. Diakses dengan membuka tab File > Build Settings... (Ctrl + Shift + B). Jika belum terdaftar, untuk menambahkan scene dapat mengklik tombol **add open scenes**.



7. PEMBUATAN HALAMAN SIMULASI

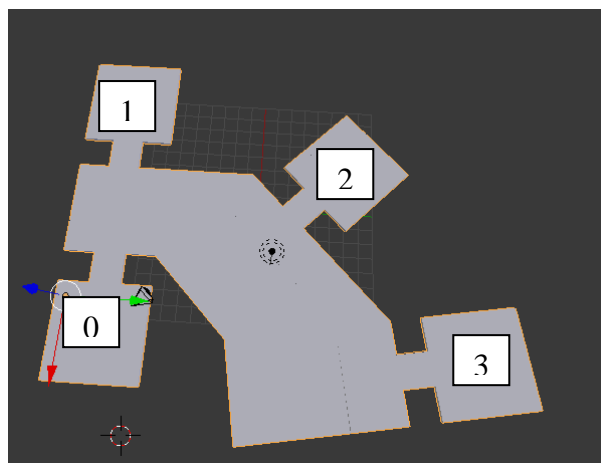
Pembuatan halaman simulasi dibagi menjadi 3 bagian yaitu:

1. Halaman Awal

Halaman awal adalah halaman ketika permainan pertama kali dimainkan yaitu halaman menu utama beserta dengan halaman *about*.

2. Halaman Pemilihan Level

Halaman pemilihan level dibuat seperti halaman permainan, dimana hal ini bertujuan untuk memberikan kesan aplikasi dan juga membiaskan pemain dalam permainan.



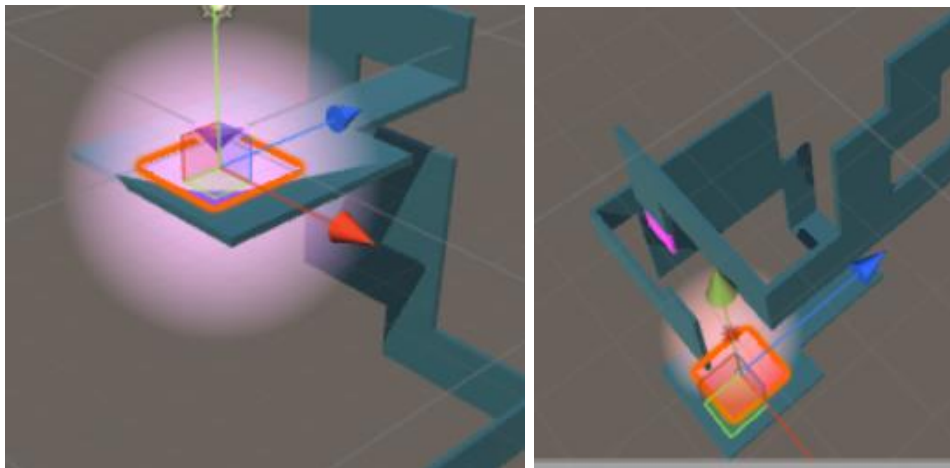
Setiap angka pada gambar di atas mewakili:

- 0 : Awal mula pemain saat memilih *level picker*
- 1 : Masuk ke level tutorial/keluar dari level tutorial
- 2 : Masuk ke level 1/keluar dari level 1
- 3 : Masuk ke level 2/keluar dari level 2

Saat pertama kali bermain, level yang dapat diakses hanyalah level tutorial.

3. Halaman Permainan

Halaman permainan dibuat dengan memasukkan *asset* yang ada ke dalam *scene* di Blender, lalu menambahkan warna dan menggabungkan *asset-asset*, kemudian *asset* diberikan *script* yang sesuai dengan seharusnya. Pemain akan muncul pada *plane* berwarna ungu dan bertujuan menuju *plane* berwarna merah.



Jika *player* sudah berhasil mencapai titik akhir, maka *player* akan dikembalikan ke *level picker* dan level lainnya akan terbuka. Dalam permainan, *player* akan dibantu dengan teks yang menunjukkan *hint* yang di-generate dalam Unity.

8. PENGUJIAN

Aplikasi yang kami buat selanjutnya kami uji kepada beberapa responden sejumlah 5 orang. Untuk penilaian aplikasinya sendiri kami buat berdasarkan 12 aspek heuristic VR yang terdiri dari:

- 1. Natural engagement: Interaksi mendekati harapan pengguna.
- 2. Compatibility with the user's task: Lingkungan maya dan perilaku setiap objek mirip dengan aslinya.

3. Natural expression of action: Memberikan kebebasan bagi pengguna dalam berinteraksi dan eksplorasi secara alami.
4. Close coordination: Representasi yang nyata tanpa adanya lag pada tampilan. Pemutakhiran tampilan harus kurang dari 200 ms untuk menghindari masalah mabuk.
5. Realistic feedback: Umpan balik dari interaksi pengguna dapat terlihat dengan cepat dan sesuai dengan ekspektasi user.
6. Faithful viewpoints: Representasi visual pada dunia virtual dapat menyesuaikan persepsi normal pengguna dan perubahan sudut pandang oleh gerakan kepala tidak menyebabkan penundaan/delay.
7. Navigation and orientation support: Pengguna tahu di mana keberadaan mereka dalam dunia virtual yang dimasukinya.
8. Clear entry and exit points: Dapat masuk dan keluar dari dunia virtual dengan mudah.
9. Consistent departures: Adanya konsistensi dalam representasi objek.
10. Support for learning: Objek yang diperlihatkan dapat memberi wawasan/pembelajaran bagi pengguna atau tersedianya fitur untuk memahami cara pemakaian aplikasi.
11. Clear turn-taking: Adanya waktu giliran dalam berkomunikasi. Prinsip ini hanya ada dalam lingkungan virtual di mana avatar yang digunakan pengguna dapat berinteraksi dengan avatar lainnya.
12. Sense of presence: Keberadaan pengguna pada lingkungan virtual yang dimasukinya sebisa mungkin harus terasa nyata.

Hasilnya adalah sebagai berikut:

Aspek heuristik	Rating responden (1-5)					Rata-rata
	A	B	C	D	E	
Natural engagement	5	4	3	4	4	4
Compatibility with the user's task	5	3	4	3	4	3.8
Natural expression of action	5	4	3	4	5	4.2
Close coordination	3	3	3	2	4	3
Realistic feedback	5	4	5	5	4	4.6
Faithful viewpoints	4	4	3	4	5	4
Navigation and orientation support	5	5	5	4	5	4.8
Clear entry and exit points	5	5	5	5	5	5
Consistent departures	5	4	5	5	5	4.8

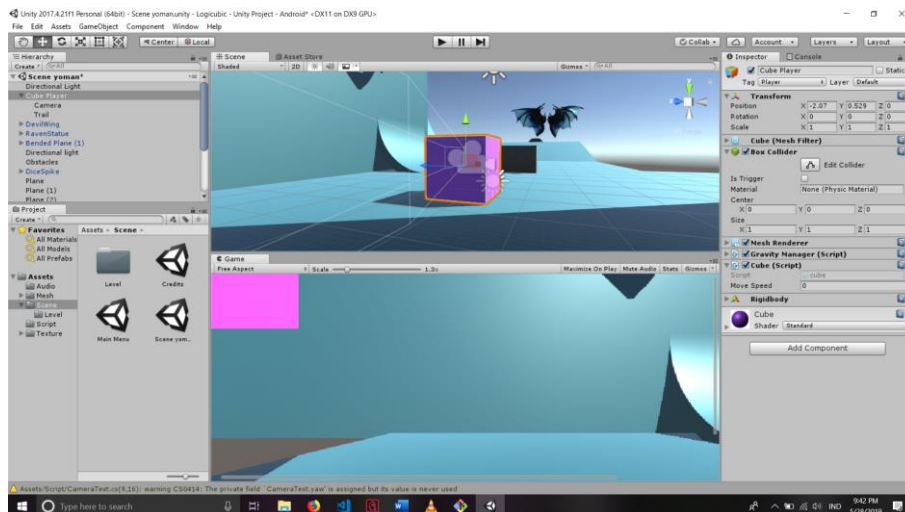
Support for learning	5	4	4	5	4	4.4
Clear turn-taking	5	5	5	5	5	5
Sense of presence	5	3	4	4	3	3.8

Kesimpulannya adalah aplikasi ini siap digunakan namun masih harus memperbaiki sisi keindahan dari aplikasinya agar lebih menarik dan lebih mirip dengan dunia nyata. Selain itu keindahan lingkungan permainan juga mendukung agar pemain tidak merasa pusing atau mabuk saat memainkan permainan ini.

9. HIGHLIGHT

Collision dengan sisi “muka”

Katakanlah kita mempunyai suatu objek.



Di sini objek yang dimaksud adalah objek kubus (Cube) bernama “**Cube Player**”

Trigger dari *OnCollisionEnter* akan terpicu jika sisi kubus manapun bertabrakan dengan objek lainnya sementara kita hanya ingin sisi bagian depan (muka) saja untuk memicu *OnCollisionEnter*. Ada banyak cara untuk mengakalinya, salah satunya adalah dengan menambah objek baru sebagai child objek Cube Player itu. Lalu menggunakan fungsi *ContactPoint* dan *thisCollider*.

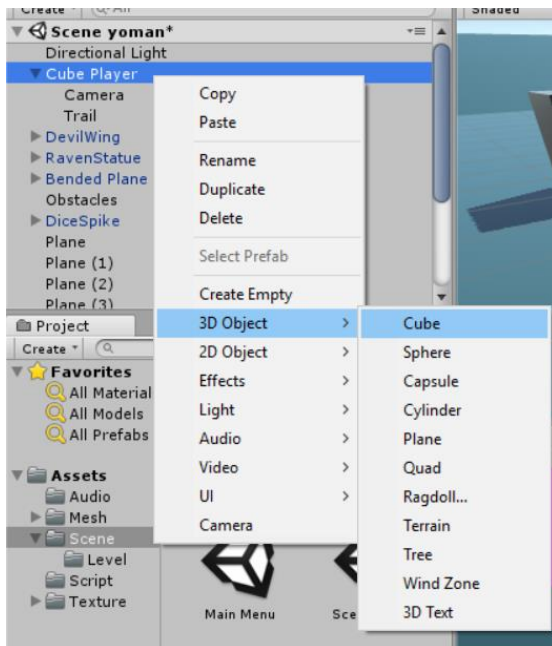
Sumber:

<https://docs.unity3d.com/ScriptReference/Collider.OnCollisionEnter.html>

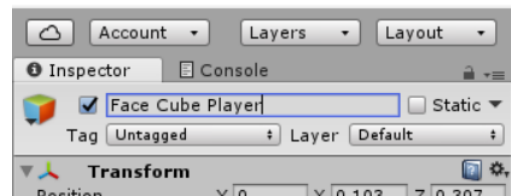
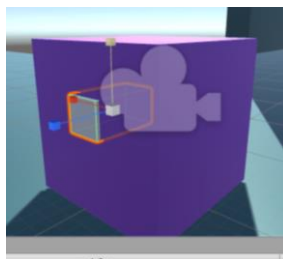
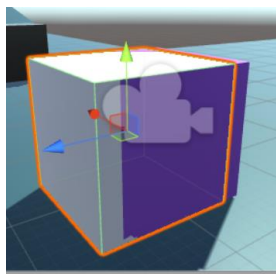
<https://docs.unity3d.com/ScriptReference/ContactPoint.html>

Cara membuat *child cube player*:

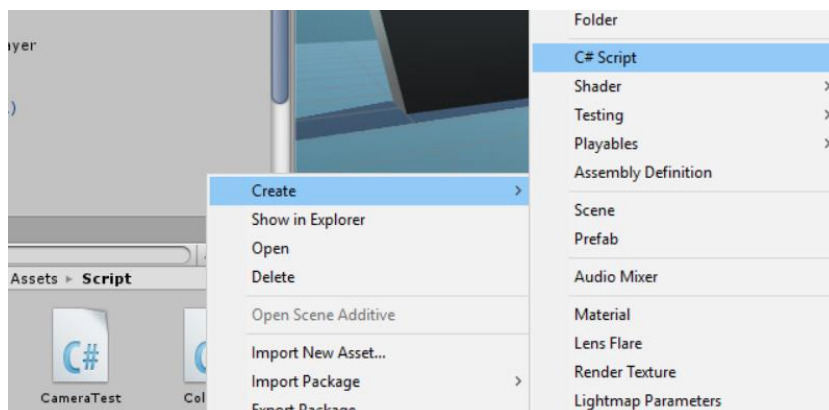
Klik kanan pada Cube Player > 3D Object > Cube.



Atur *cube* baru sehingga hanya ada di sisi bagian depan (muka) Cube Player. Jangan lupa mengganti nama *child cubenya*, misal Face Cube Player.



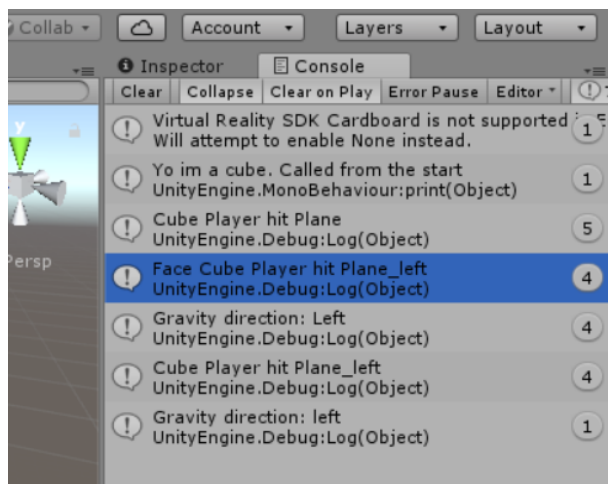
Kembali ke objek Cube Player, masuk ke bagian *code script*. Jika belum ada script, bisa buat C# script yang baru terlebih dahulu.



Di sini sudah ada script bernama GravityManager.cs yang berfungsi untuk mengatur perubahan arah gravitasi ketika Cube Player menabrak dinding.

```
void OnCollisionEnter (Collision collisionInfo) {  
    foreach (ContactPoint contact in collisionInfo.contacts){  
        Debug.Log(contact.thisCollider.name + " hit " +  
contact.otherCollider.name);  
    }  
}
```

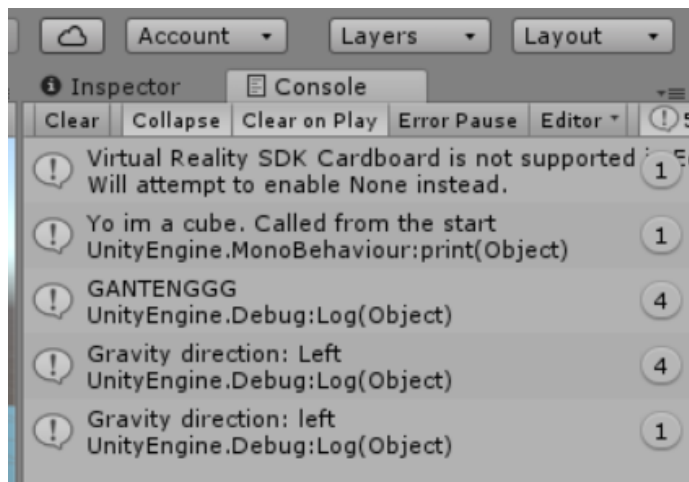
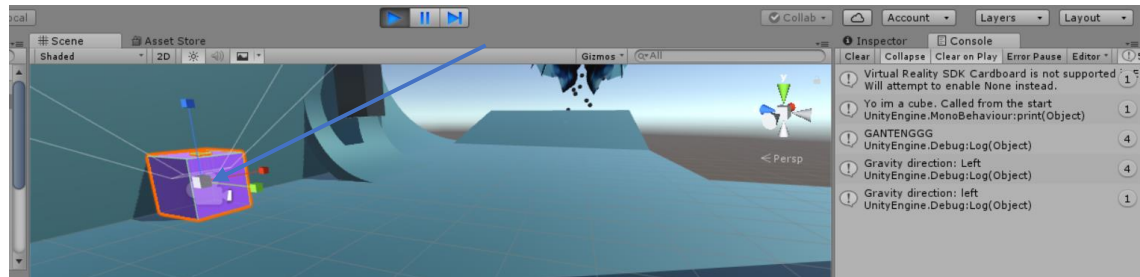
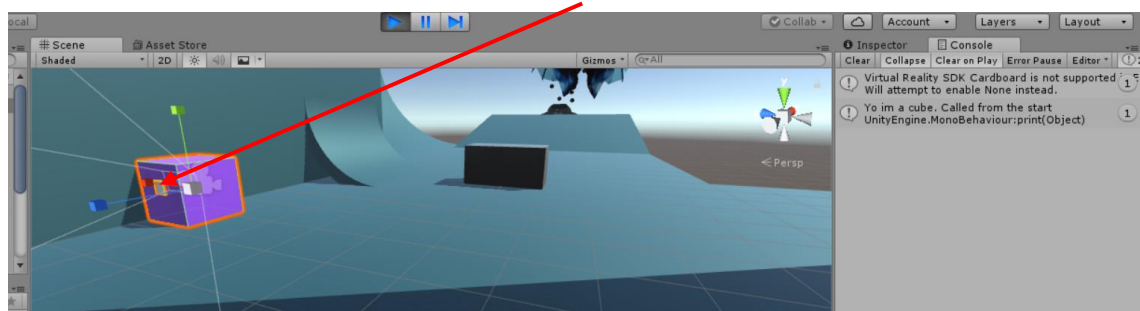
contact.thisCollider.name akan memberi info siapa dengan siapa yang bertabrakan. Tentu hasilnya dapat dilihat di bagian console.



Sekarang jika ingin trigger hanya akan dilakukan jika bertabrakan dengan Face Cube Player cukup dengan menambahkan IF.

```
void OnCollisionEnter(Collision collisionInfo) {  
    foreach (ContactPoint contact in collisionInfo.contacts){  
        //Debug.Log(contact.thisCollider.name + " hit " + contact.otherCollider.  
name);  
        // Only change gravity if cube is collision with the FACE  
        if (contact.thisCollider.name == "Face Cube Player") {  
            Debug.Log("GANTENGGG");  
        }  
    }  
}
```

Hasilnya, jika yang tabrakan bukan dengan Face Cube Player, tidak akan muncul pesannya.



Sumber: <https://gamedev.stackexchange.com/questions/112571/can-i-use-two-collision-boxes-on-the-same-gameobject>

BIODATA SINGKAT ANGGOTA TIM

Bagian ini berisi biodata singkat anggota tim yang terdiri dari ketua dan 2 anggota.

	<p>Nama lengkap: Muhammad Rifqy Aulia Akbar NPM: 140810160055 Jenis kelamin: Pria Tempat/tanggal lahir: Bandung/28 Februari 1998 E-mail: mrifqyaa@gmail.com No HP: 081213000725 Alamat: Jl. Purbasari 1 no. 46, RT 02/05, Sangkuriang, Cipageran, Cimahi Utara, Jawa Barat</p>
	<p>Nama lengkap: Muhammad Raihan Akbar NPM: 140810160013 Jenis kelamin: Laki laki Tempat/tanggal lahir: Jakarta / 19 Januari 1999 E-mail: raihanakbar1999@gmail.com No HP: 0896 5614 9543 Alamat: Pondok Ohim, Cisaladah, Kec. Jatinangor, Kab. Sumedang, Jawa Barat</p>
	<p>Nama lengkap: Patricia Joanne NPM: 140810160065 Jenis kelamin: Perempuan Tempat/tanggal lahir: Jakarta, 24 Juni 1999 E-mail: realicejoanne@yahoo.co.id No HP: 081932698624 Alamat: Wisma Mayangsari, Jl. Lapang Bola RT 03/05, Cikeruh, Kec. Jatinangor, Kab. Sumedang, Jawa Barat</p>