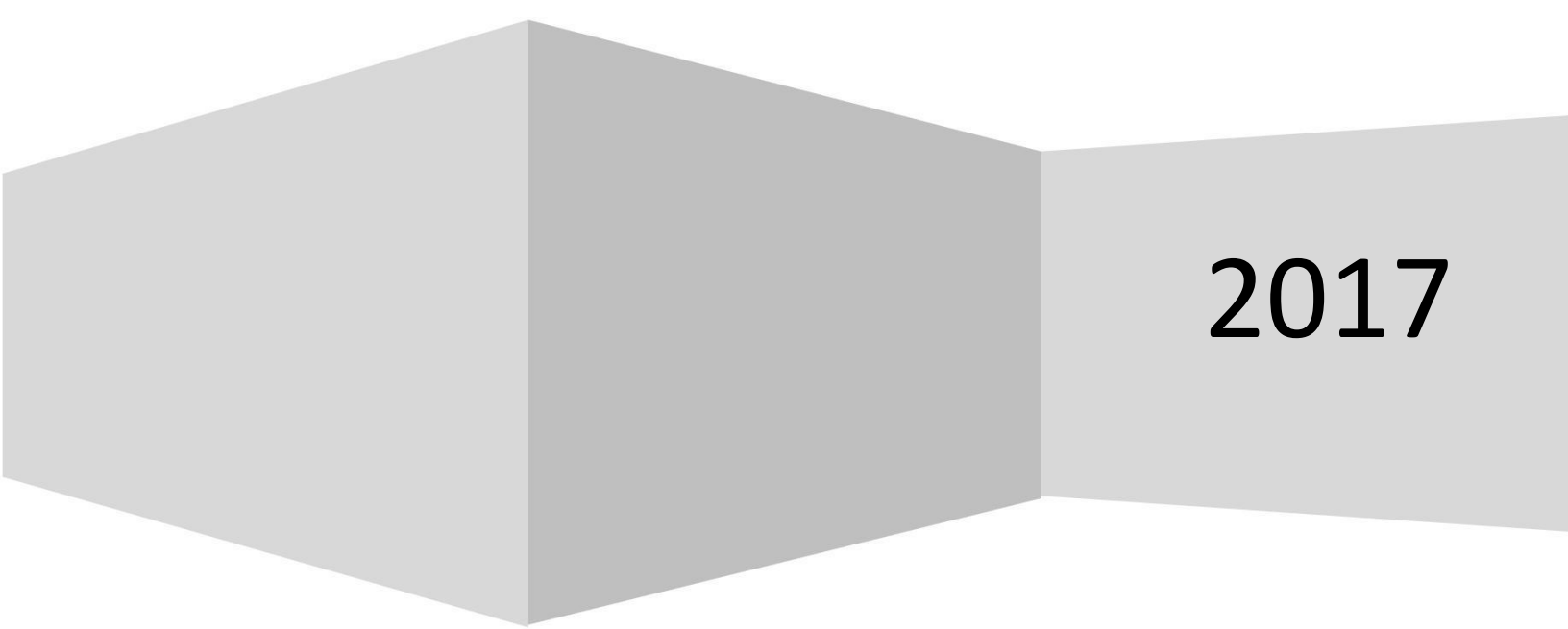


Teknik Informatika

Modul

Praktikum Pemrograman Game dan Multimedia



2017

I.	Contents	
II.	Pendahuluan	2
III.	Memulai Unity	5
IV.	Membuat Terrain	8
V.	Pencahayaan	10
A.	Point light.....	10
B.	<i>Directional light</i>	10
C.	Spotlight	11
D.	Area light.....	11
VI.	Third Person Shooter	12
VII.	First Person Shooter.....	14
A.	Membuat game fps.....	14
B.	Menambahkan efek ledakan.....	16
VIII.	Pengaturan Model	18
A.	Pengaturan Model yang dibuat sendiri.....	18
B.	Pengaturan animasi.....	18
IX.	Pengaturan Animasi Player dan Lawan	23
A.	Player.....	23
B.	Lawan statis	23
C.	Lawan dinamis.....	24
X.	Menambahkan efek pada player	27
XI.	Scoring.....	29
XII.	AugmentedReality Dengan Vuforia	33

II. Pendahuluan

Pembuatan game dapat menggunakan berbagai software yang dirancang untuk membangunnya. Beberapa software yang dapat digunakan antar lain:

1. *Game Maker*

Game Maker merupakan software pembuat game/game engineering yang dibuat oleh yoyo games. Game Maker sendiri menggunakan sistem drag and drop untuk membuat sebuah event, penempatan sprite, penataan ruang dan lain sebagainya. Game Maker menggunakan bahasa pemrograman GML (Game Maker Language) .

2. Construct 2

Construct 2 merupakan sebuah software pembuat game/game engineering yang menggunakan bahasa HTML 5, software ini dikhususkan untuk game 2D yang dikembangkan oleh Scirra. Construct 2 dapat dijadikan multi platform, yaitu dengan bantuan aplikasi tertentu seperti: Untuk membuild game ke PC : Windows, MAC maupun linux bisa menggunakan software bantuan yang bernama Node-Webkit, sedangkan untuk membuild ke platform IOS dan Android diperlukan software CocoonJS, AppMobi, dan PhoneGap.

3. RPG Maker

Software RPG Maker ini adalah software yang dikhususkan untuk membuat sebuah game RPG(Role Playing Game) dimana player memainkan sebuah karakter dalam menyelesaikan petualangannya, player dapat menyelesaikan berbagai macam quest-quest, upgrade skill, dan melawan boss. RPG Maker ini sendiri sebenarnya dikhususkan untuk single platform yaitu PC, tetapi dengan hadirnya versi terbarunya yaitu RPG Maker MV bisa membuild game ke berbagai platform seperti PC, IOS, Android dan masih banyak lagi. Bahasa pemrogramannya pun mengalami perubahan, awalnya RPG Maker ini menggunakan bahasa pemrograman Ruby tetapi setelah RPG Maker MV di luncurkan RPG Maker mengganti bahasa pemrogramannya menjadi javascript. RPG Maker ini menggunakan sistem GUI, bahkan fitur-fitur event disediakan dalam bentuk GUI, sehingga jika ingin menambahkan event dalam sebuah game, hanya mendrag/memilih event tersebut kedalam game. Tidak hanya itu saja RPG Maker ini menggunakan prinsip Tile Map Mapping yang semua sprite map yang dibutuhkan sudah disediakan oleh RPG Maker sendiri, juga sprite karakter, musuh dll, sudah di sediakan oleh RPG Maker.

4. *Cocos2D*

Cocos2D merupakan software pembuat game/game engine yang bersifat opensource. Cocos2D ini memiliki beberapa varian jenis, seperti Cocos2D, Cocos2D-X, Cocos2D-JS, Cocos2D-XNA, dan Cocos2D-Swift. cocos2D dapat membuat game dengan basis 2D. Bahasa pemrograman yang digunakan untuk membuat game menggunakan Cocos2D ini sangat bervariasi tergantung versi apa yang dipakai. Cocos2D adalah awal dari semua varian Cocos2D dan menggunakan bahasa Python, Cocos2D-X adalah versi terakhir dari varian Cocos2D dan Cocos2D-X ini memiliki lebih banyak fitur dari versi sebelumnya, bahasa yang digunakan Cocos2D-X adalah C++, Javascript, Lua.

5. GameSalad

GameSalad adalah software pembuat game/game engine yang bersifat multiplatform, GameSalad ini awalnya hanya bisa membuild game ke platform tertentu seperti IOS, Macbook tetapi setelah GameSalad meluncurkan versi terbarunya GameSalad sudah bisa membuild ke berbagai platform, seperti PC, IOS, Android, Windows Phone dll. GameSalad ini menyediakan fitur Drag and Drop, UI nya pun sangat mudah untuk dipahami. GameSalad ini tidak menekankan penggunaannya untuk menggunakan bahasa pemrograman karena bisa membuat game tanpa menggunakan bahasa pemrograman hanya perlu menyetting property-propertinya saja

6. Monogame

Monogame adalah software pembuat game/game engine yang bersifat multiplatform, platform-platform yang bisa dibuild dengan engine ini diantaranya: Android, IOS, PC, Mac, Play Station dan masih banyak lagi. Bahasa pemrograman yang digunakan untuk membuat game adalah C

7. Unity

Unity 3D adalah sebuah game engine yang berbasis cross-platform. Unity dapat digunakan untuk membuat sebuah game yang bisa digunakan pada perangkat komputer, ponsel pintar android, iPhone, PS3, dan bahkan X-BOX. Unity adalah sebuah tool yang terintegrasi untuk membuat game, arsitektur bangunan dan simulasi. Unity bisa untuk

games PC dan games Online. Untuk games Online diperlukan sebuah plugin, yaitu Unity Web Player, sama halnya dengan Flash Player pada Browser. Unity tidak dirancang untuk proses desain atau modelling, sehingga diperlukan 3D editor lain seperti 3dsmax atau Blender. Banyak hal yang bisa dilakukan dengan unity, ada fitur audio reverb zone, particle effect, dan Sky Box untuk menambahkan langit. Fitur scripting yang disediakan, mendukung 3 bahasa pemrograman, JavaScript, C#, dan Boo. Flexible and EasyMoving, rotating, dan scaling objects hanya perlu sebaris kode. Begitu juga dengan Duplicating, removing, dan changing properties. Visual Properties Variables yang di definisikan dengan scripts ditampilkan pada Editor. Bisa digeser, di drag and drop, bisa memilih warna dengan color picker. Berbasis .NET. Artinya penjalanan program dilakukan dengan Open Source .NET platform, Mono.

Contoh game yang dibuat menggunakan engine UNITY ada beberapa game yang memiliki kualitas bagus dan telah menjadi terkenal:

1. Bad Piggies

Bad piggies merupakan Salah satu Game paling Funny untuk dimainkan, Game ini dirilis pertama kali pada tahun 2012 lalu oleh raksasa developer game asal finlandia bernama ROVIO.

2. Dread Out

Game dreadout merupakan game salah satu game karya anak bangsa paling mendunia,. Game dreadout pertama kali dirilis pada mei 2014 lalu oleh developer game bernama Digital Happiness.

3. Temple Run 2

Game ini dirilis pertama kali pada January 2013 dan dikembangkan oleh developer game asal amerika bernama imangi studios.

4. Angry Bird ROVIO.

5. Pokemon Go

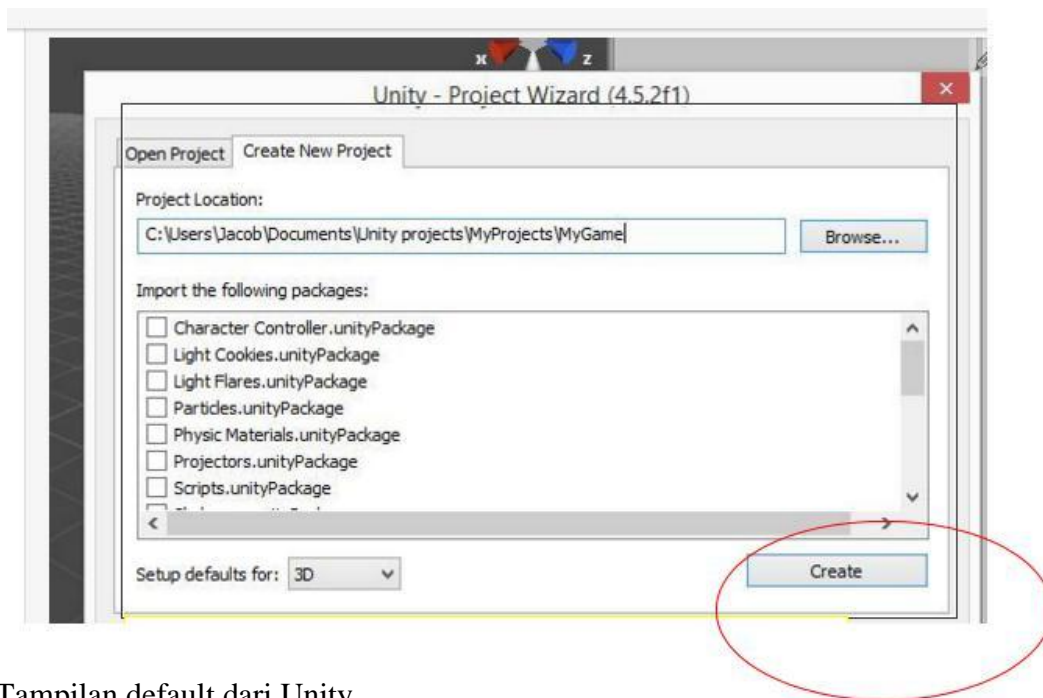
III. Memulai Unity

Tugas : Buatlah Project pada unity sesuai petunjuk II!

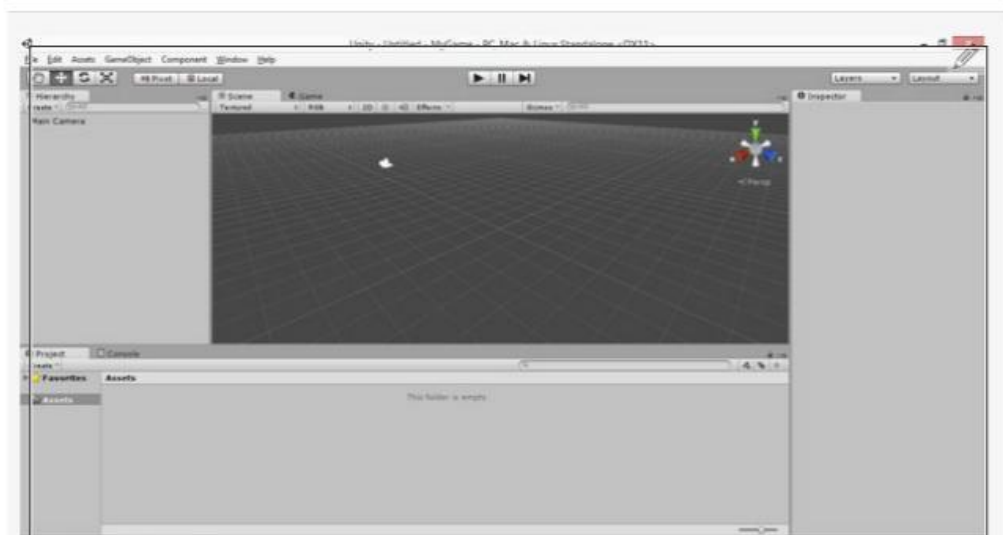
1. Klik File – New Project

Pilih lokasi folder untuk menyimpan project kemudian berikan nama

project Klik “Create.”



2. Tampilan default dari Unity



3. Menyiapkan layout

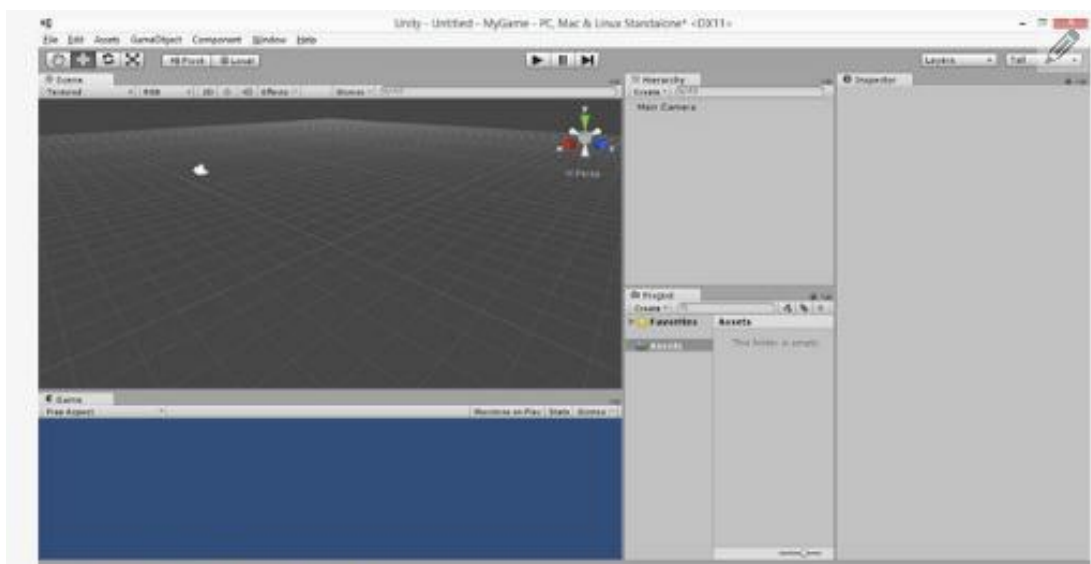
Scene - ini adalah tempat pembuatan game terjadi. Disini menampilkan elemen apa yang Anda miliki dalam permainan Anda . Ada satu blok di pojok kanan atas yang menunjukkan orientasi spasial scene.

Game - menunjukkan tampilan kamera utama saat permainan dimainkan. Anda dapat menguji permainan Anda di jendela ini dengan mengklik tombol Putar di bagian atas, tengah layar.

Hirarki - mencantumkan semua elemen yang telah Anda tambahkan ke dalam scene. Ini adalah kamera utama secara default. Anda dapat membuat elemen baru dengan mengklik Buat dan pilih jenis objek yang Anda inginkan. Hal ini juga bisa dilakukan dengan menggunakan menu dropdown GameObject di bagian atas layar.

Project - menunjukkan file yang digunakan untuk game. Anda bisa membuat folder, skrip, dll dengan mengklik Create di bawah jendela Project.

Inspector - ini adalah tempat Anda menyesuaikan aspek dari setiap elemen yang ada di tempat kejadian. Cukup pilih objek di jendela Hirarki atau klik dua kali pada objek di jendela Scene untuk menunjukkan atributnya di panel Inspector.



4. Menyimpan scene dan memulai pembuatan

Klik File - Save Scene. Simpan scene di bawah folder [Nama Proyek] - Assets. Assets adalah folder persiapan pembuatan dimana Anda ingin menyimpan scene dan skrip Anda.

Simpan scene sebagai Scene atau Main

Klik File - Build Settings.

Tambahkan scene saat ini untuk membangun.

Pilih platform yang diinginkan. Ada banyak pilihan, termasuk komputer, sistem permainan, dan ponsel pintar, namun jika Anda membuat proyek untuk pertama kalinya, Anda kemungkinan besar ingin memilih Web Player atau PC / Mac / Linux Standalone.

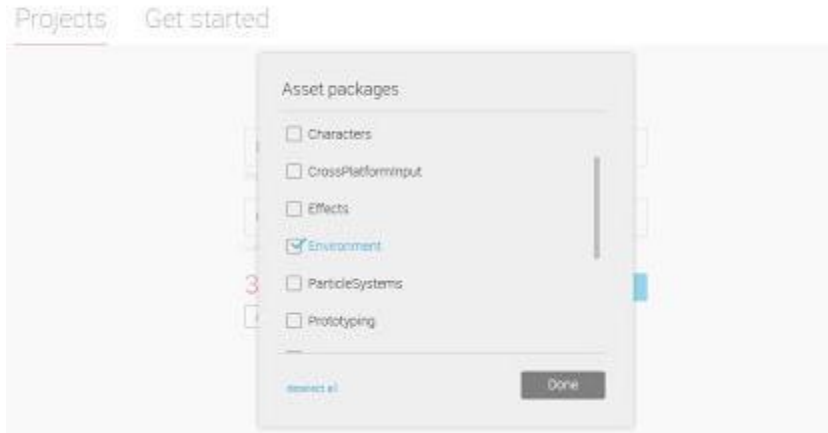
Klik Player Settings di bagian bawah jendela Build Settings. Ini akan membuka pilihan Pengaturan Pemain di Inspector. Di sini, Anda bisa mengganti nama perusahaan, nama produk (game), ikon default, dll.

Tutup jendela Build Settings. Anda akan kembali ke ini saat Anda siap menyelesaikan game Anda.

IV. Membuat Terrain

Tugas : Buatlah Terrain sesuai petunjuk III!

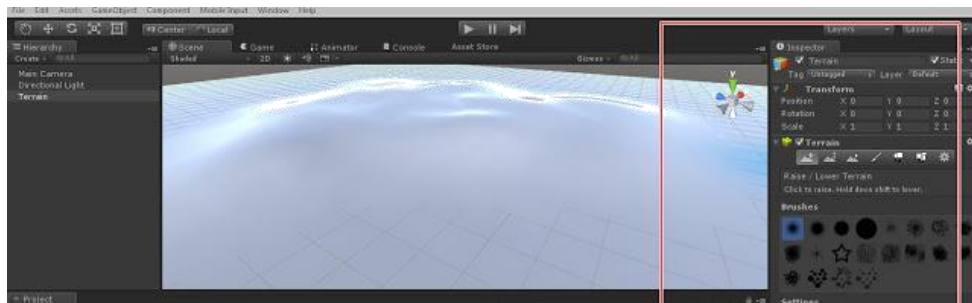
1 . Import Package Terrain asset , atau pada unity 5 , terdapat pada enviornment .



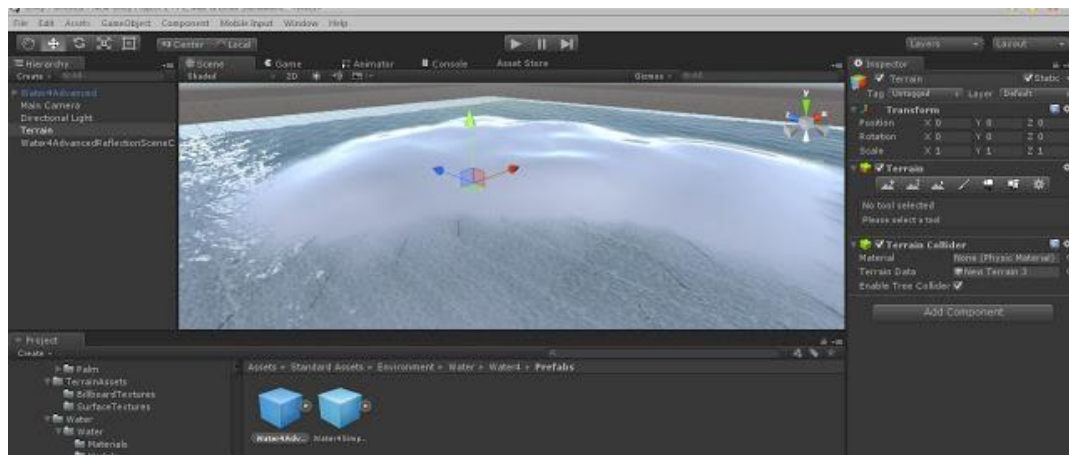
2. Siapkan Directional light , dan main camera , dan import charecter asset / charecter package



3. Tambahkan imajinasi anda sendiri dengan membuat object seperti bukit



4. Beri effect water basic di sekitar bukit



V. Pencahayaan

Tugas : Buatlah 4 pencahayaan sesuai petunjuk IV!

Tambahkan Lighting berikut pada terrain yang telah dibuat pada pertemuan sebelumnya dan gunakan masing masing pencahayaan.



A. Point light

Sumber cahaya ini bisa dibayangkan sebagai sebuah bola yang mengeluarkan cahaya yang tersebar dari seluruh permukaannya ke semua arah. Point light cocok digunakan untuk mensimulasikan cahaya dari bohlam lampu, obor, dan sumber yang sejenis. Point light bisa digunakan secara *realtime* maupun *baked*.

B. Directional light

Directional light, dengan cahaya seolah berasal dari bawah

Directional light mensimulasikan cahaya yang datang dari sumber jauh di atas, seperti matahari.

Posisi dari sumber cahaya ini tidak berpengaruh pada pencahayaan serta bayangan yang dihasilkan saat cahaya darinya mengenai objek. Namun, orientasi sudutnyalah yang berpengaruh.

C. Spotlight

Mensimulasikan sumber cahaya yang menyerupai senter. Cahaya yang muncul bisa dibayangkan seperti kerucut dengan ujung lancipnya dipotong. Sumber cahaya ini akan menyorot sebuah objek dan juga menghasilkan bayangan objek tersebut. Kita bisa mengatur jarak tempuh cahaya, berikut sudut sorot cahaya. Ia bisa digunakan secara *realtime* ataupun *baked*.

D. Area light

Posisinya sejajar dengan tempat duduk pada kursi, perhatikan bagian mana yang terang *Area light* akan menghasilkan cahaya untuk area tertentu, mengikuti bentuk segi empat. Ukuran segi empat ini akan mempengaruhi jarak tempuh cahaya. Bayangan yang dihasilkan dari sumber cahaya ini halus, berkat cahaya yang lembut dan intensitasnya yang berkurang secara mulus di ujung area pencahayaan. Karena banyaknya informasi yang harus diolah oleh sumber cahaya ini, ia hanya bisa digunakan secara *baked*.

VI. Third Person Shooter

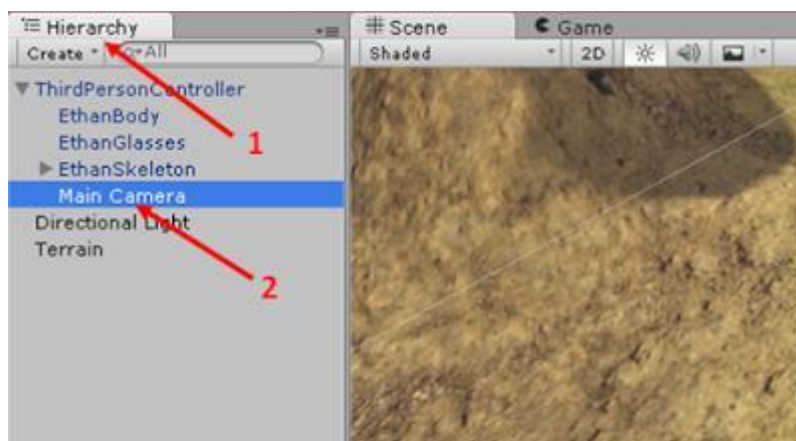
Tugas : Buatlah game TPS sesuai petunjuk V!

1. Buatlah Terrain

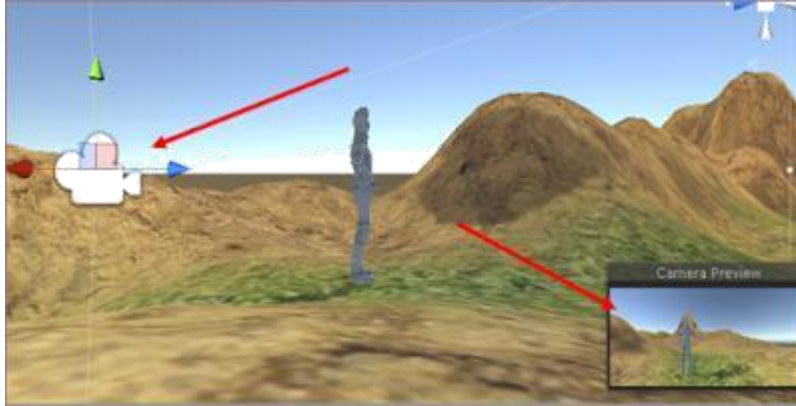
2. Pada Tab Project > Assets > Standard Assets > Characters > ThirdPersonController > Prefabs > ThirdPersonController ... lalu drag ke atas terrain, ubah Position di Tab Inspector



3. Pada Tab Hierarchy, drag dan pindahkan "Main Camera" ke Object ThirdPersonController yang ada di Tab Hierarchy.



4. Pada Main Camera, pindahkan ke dekat Object ThirdPersonController, sesuai dengan tampilan kamera yang kamu lihat di pojok kanan bawah.



5. Mainkanlah Game yang telah anda buat dengan mengklik tombol play seperti pada gambar dibawah.



6. Simpan Scene Game Anda dengan cara : Pada Menu Bar File > Save Scene as, kemudian beri nama dan klik Save.

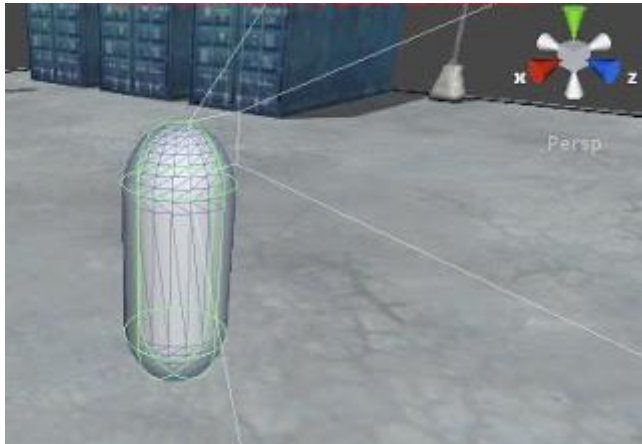
VII. First Person Shooter

Tugas : Buatlah game FPS sesuai petunjuk VI!

A.Membuat game fps

Karakter FPS yang akan digunakan adalah aset standar yang telah disediakan oleh Unity3d.

1. Klik dan drag aset First Person Controller (jendela project, folder Standard Assets > Prefabs) ke scene. Di scene akan muncul objek berbentuk silinder yang merepresentasikan pemain game.

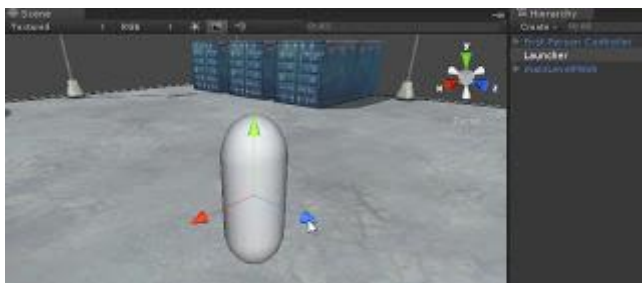


jika penempatan benar maka ketika tampilan Play dijalankan, anda sudah dapat mengeksplorasi area dengan menggunakan keyboard dan mouse sebagai alat gerak.

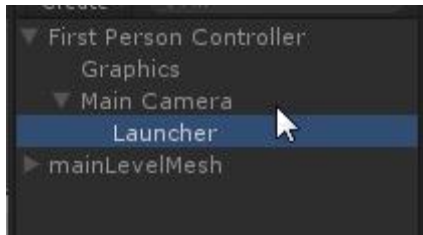
2. Selanjutnya menambahkan senjata. Menambahkan peluncur granat: Buat game objek “empty”, rename sebagai Launcher, tempatkan di tempat yang sama dengan aset FPS di scene dengan cara:

Klik FPS di panel Hierarchy, letakkan kursor mouse di Scene, tekan tombol "F" di keyboard.

Klik Launcher di panel Hierarchy lalu pilih menu Game Object > Move to view. Di scene akan terlihat launcher berpindah tempat ke tempat FPS asset.



Tetapi ketika game dijalankan, objek launcher belum bergerak mengikuti objek FPS. Untuk mengatasi masalah ini, drag objek Launcher ke dalam Main camera yang dimiliki oleh FPS di jendela Hierarchy.



Klik tombol Play, Ketika menjalankan objek FPS, perhatikan Scene View, objek Launcher akan ikut bergerak bersama objek FPS.

3. Membuat objek granat sederhana: Buat prefab baru, rename sebagai Missile. Kemudian buat objek bola (klik menu GameObject > Create Other > Sphere). Objek bola ini akan ditampilkan di jendela Hierarchy dan Scene View. Drag objek bola ini dari panel Hierarchy ke prefab Missile yang ada di panel project. Hapus objek Sphere di panel Hierarchy.

4. Membuat kode peluncur granat: Buat dokumen Javascript baru, beri nama MissileLauncher, isikan kode berikut:

```
var projectile : Rigidbody;
```

```
var speed = 20;
```

```
function Update()
```

```
{
```

```
    if( Input.GetButtonDown( "Fire1" ) )
```

```
    {
```

```
        var instantiatedProjectile : Rigidbody = Instantiate(projectile,
```

```
transform.position, transform.rotation );
```

```
        instantiatedProjectile.velocity =transform.TransformDirection( Vector3( 0, 0, speed )
```

```
        ); Physics.IgnoreCollision( instantiatedProjectile.collider,
```

```
        transform.root.collider );
```

```
    }
```

```
}
```

Attach script ini kedalam objek Launcher yang ada di dalam objek FPS. Untuk memastikan bahwa objek FPS telah memiliki script Missile Launcher, klik objek Launcher di panel

Hierarchy, di panel Inspector akan muncul Properti script Missile Launcher. Disini terlihat variabel projectile masih kosong.

5. Klik prefab Missile di panel Project, klik Components > Physics > Rigidbody
6. Klik Launcher di panel Hierarchy, drag prefab Missile dari panel Project ke variabel projectile di panel Inspector.
7. Jalankan game, klik mouse, sekarang anda dapat menembakkan granat berbentuk bulat.



B. Menambahkan efek ledakan.

Tugas : Tambahkan efek pada game FPS sesuai petunjuk VI B!

Buat javascript baru, beri nama Projectile. Isikan kode berikut:

```
var explosion : GameObject;  
function OnCollisionEnter( collision : Collision )  
{  
    var contact : ContactPoint = collision.contacts[0];  
    var rotation = Quaternion.FromToRotation( Vector3.up, contact.normal );  
    var instantiatedExplosion : GameObject = Instantiate( explosion,  
    contact.point, rotation );  
    Destroy( gameObject );  
}
```

Attach script ini ke prefab Missile. Selanjutnya membuat ledakan, buat prefab baru, beri nama Explosion. Efek ledakan akan menggunakan aset Particle standar dari Unity3d. Drag prefab Explosion dari panel Project Standard Assets > Particles > Explosion ke panel Hierarchy. Lakukan seting sesuka anda, kemudian drag Explosion dari panel Hierarchy ke prefab Explosion yang tadi telah dibuat.

9. Klik prefab Missile di panel project, drag prefab Explosion dari panel Project ke variabel Explosion (script Projectile) di panel Inspector.

10. Buat script baru, rename dengan Explosion, isikan kode:

```
var explosionTime = 1.0;  
function Start()  
{  
  Destroy( gameObject, explosionTime );  
}
```

attach script Explosion ke prefab Explosion. Jalankan game, sekarang setiap klik mouse, granat akan muncul dengan efek ledakan.

11. Menambahkan suara ledakan. Klik prefab Explosion di panel Project. Klik Component > Audio > Audio Source. Panel Inspector akan menampilkan Audio Source. Klik tanda bulatan kecil di samping kanan kotak Audio Clip. Pilih Rocket Launcher Impact.

.

VIII. Pengaturan Model

Tugas : animasikan model yang telah anda buat pada blendersesuai petunjuk VII!

A. Pengaturan Model yang dibuat sendiri

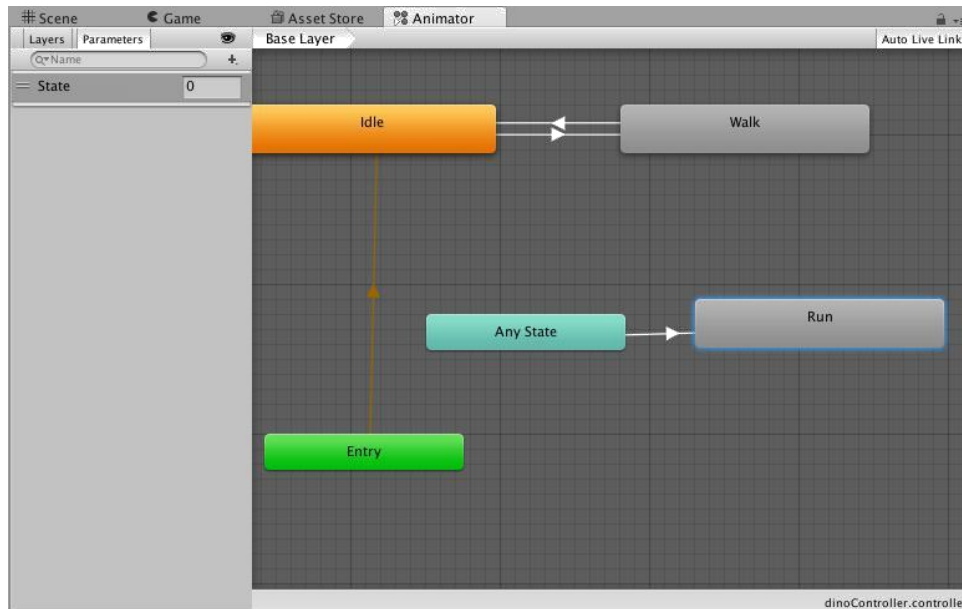
1. Karakter dibuat dengan blender
2. Dalam pembuatan karakter menggunakan blender gunakan tekstur yang diproyeksikan dengan UV. Model , animasi dan tekstur diimpor kedalam unity
3. Mengimpor dengan cara Asset-Import package-custom package
4. Buka folder Moel dan pasang objek *.blend yang telah dibuat
5. Tambahkan teksturnya dengan cara klik texture pada panel asset dan tarik ke mesh objek
6. Tambahkan komponen karakter kontroler dengan cara Component-physics-character-controller
7. Atur ukuran karakter kontroler sehingga meliputi seluruh tubuh objek *.blend misalkan radius:2 dan height 14.

B. Pengaturan animasi

FSM (Finite State Machine) yang mengatur bagaimana perubahan state animasi milik sebuah objek terjadi. Sebuah Animator akan menghubungkan berbagai Animation Clip dalam diagram pohon. Dalam Unity kemudian memiliki kebebasan untuk mengarahkan bagaimana Animation Clip ini terhubung, serta apa urutan kejadiannya. Lebih lanjut bahkan Animation Clip ini bisa di-*trigger* melalui script jika dibutuhkan.

objek yang bergerak mengikuti perintah pemain, sambil menjalankan animasi untuk gerakan tersebut, ataupun jika kita ingin mengotomasi beberapa animasi hingga menjadi animasi yang kompleks.

1. tamahkan model selanjutnya buka Animator aset tersebut, dan membuat Controller seperti berikut



Ada 3 state di situ, yakni Idle, Walk dan Run. Animasi untuk ketiganya adalah:

Idle: Comp|StandE

Walk: Comp|Walk

Run: Comp|Run

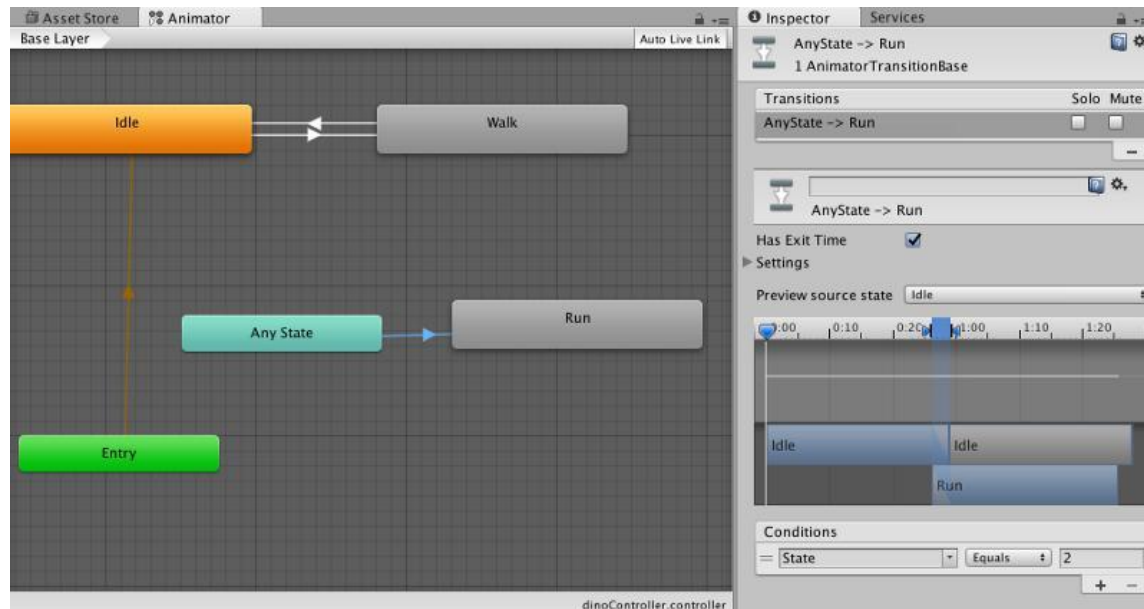
2. Selain state, juga menempatkan 3 transisi, yakni untuk Idle → Walk, Walk → Idle, serta Any State → Run. Di setiap transisi ini, kita tempatkan parameter apa yang menjadi pemicu perubahan state, sekaligus animasi. Parameter bisa ditambahkan di tab Parameter milik Animator. Kita berikan nama parameter kita State, dan nama ini akan dijadikan acuan untuk transisi sebagai berikut:

Idle → Walk: tidak ada parameter, otomatis terjadi perpindahan state ke Walk, setelah animasi Idle selesai

Walk → Idle: terjadi saat parameter State bernilai 0 (nol)

Any State → Run: terjadi saat parameter State bernilai 2

3. Untuk mengubah properti transisi ini, kita cukup mengklik panah transisi untuk menampilkan menu baru. Contohnya seperti berikut



4. Mengatur State sebagai parameter transisi

5. Setelah menentukan kondisi yang menyebabkan perubahan state, maka kita bisa dengan mudah men-trigger perubahan state ini menggunakan Script. Untuk itu, mari kita buat script baru, kita namakan “orang”

tambahkan berikut:

```
using UnityEngine;
```

```
using System.Collections;
```

```
public class orang : MonoBehaviour {
```

```
    public float walkSpeed;
```

```
    float walkSpeed_; //penampung nilai walkSpeed awal
```

```
    Animator animator;
```

```
    GameObject orang;
```

```

void Start () {
    animator = GetComponent<Animator>();
    walkSpeed_ = walkSpeed;
    idle();
}

void Update () {
    // model orang baru bergerak setelah frame ke
    10 if (Time.frameCount > 10) {
        if (transform.position.x <= 4) {
            run();
        } else {
            walk();
        }
        transform.Translate(0, 0, walkSpeed * Time.deltaTime);
    }
}

void run() {
    walkSpeed = walkSpeed_ * 2;
    animator.SetInteger("State", 2);
}

void walk() {
    walkSpeed = walkSpeed_;
    animator.SetInteger("State", 1);
}

void idle() {
    walkSpeed = 0;
}

```

```
        animator.SetInteger("State", 0);  
    }  
}
```

Dalam script tadi, kita tempatkan fungsi yang mengubah state, yakni run(), walk() dan idle() . Saat game dimulai, ia akan masuk ke state idle dengan mengubah properti State milik animator dengan nilai 0. Selanjutnya, dalam update() , fungsi walk() akan dijalankan, dan bila sudah masuk ke posisi tertentu, fungsi run() akan diaktifkan, di mana state dan kecepatan berjalan diubah.

IX. Pengaturan Animasi Player dan Lawan

Tugas : Buatlah game player dan lawan statis sesuai petunjuk VIII A-B!

Tampilkan 3 objek kubus, bola dan silinder

A. Player

Kubus diposisikan sebagai player, Masukkan script berikut:

```
using UnityEngine;
using System.Collections;

public class PlayerBehaviour : MonoBehaviour {
    public float movingSpeed = 15; //kecepatan bergerak

    void Update () {
        //Jika keyboard di tekan
        if (Input.GetKey(KeyCode.LeftArrow))
            this.transform.Translate(Vector3.left * movingSpeed * Time.deltaTime, Space.World);
        else if(Input.GetKey(KeyCode.RightArrow))
            this.transform.Translate(Vector3.right * movingSpeed * Time.deltaTime,
            Space.World); else if (Input.GetKey(KeyCode.UpArrow))
            this.transform.Translate(Vector3.forward * movingSpeed *
            Time.deltaTime, Space.World);
        else if (Input.GetKey(KeyCode.DownArrow))
            this.transform.Translate(Vector3.back * movingSpeed * Time.deltaTime, Space.World);
    }
}
```

B. Lawan statis

Silinder sebagai lawan statis, artinya lawan ini hanya berdiam tetapi memiliki reaksi jika terkena pukulan

Tambahkan script berikut:

```
Using UnityEngine;
Using System.Collections;
Public class TargetObject : MonoBehaviour {
Public string TriggerTag ;
Float multiply;
Float force;
// digunakan untuk
inisialisasi Void start () {
}
// update dilakukan satu kali tiap frame
Void Update () {
Void OnTriggerEnter(Collider collider){
    If (collider.gameObject.tag==TriggerTag) {
multiply=collider.gameObject.transform.localScale.x; Force=10*multiply*multiply;// fungsi
akan aktif jika terjadi tumbuan rigidbody dan collider.
This.GetComponent<Rigidbody>().AddExplosionForce(force,transform.position-
(multiply*collider.transform.forward),multiply,multiply);
}
}
}
```

C. Lawan dinamis

Tugas : Buatlah game player dan lawan statis sesuai petunjuk VIII C!

Lawan dinamis adalah lawan yang dapat aktif bereaksi terhadap player misalkan mengear pada radius tertentu.

Tambahkan scrip berikut pada objek bola :

```
using UnityEngine;
using System.Collections;

public class AICharacterBehaviour : MonoBehaviour {
    public GameObject target; //target yang dikejar
    public float movingSpeed = 2f; //kecepatan berpindah
    public float turnSpeed = 0.05f; //kecepatan berbelok

    void Update() {
        Vector3 gapPosition = target.transform.position - this.transform.position; //Gap
        antara posisi AI dengan target
        gapPosition = new Vector3(gapPosition.x, 0, gapPosition.z); //Nilai gap y dibuat 0 agar
        AI mengabaikan posisi atas dan bawah (Y) dari target dan hanya mengikuti arah ke kanan dan
        ke kiri (X dan Y)
        Quaternion lookRotation = Quaternion.LookRotation(gapPosition); //Rotasi untuk look
        atau melihat target
        this.transform.rotation = Quaternion.Lerp(this.transform.rotation, lookRotation,
        turnSpeed); //Membuat rotasi berubah secara smooth menggunakan fungsi lerp dari rotasi awal
        ke rotasi tujuan lookRotation

        this.transform.Translate(Vector3.forward * movingSpeed * Time.deltaTime); //Bergerak
        maju
    }
}
```


X. Menambahkan efek pada player

Tugas : Buatlah Tambahkan Efek pada player sesuai petunjuk IX!

1. Buat objek 3d bola beri nama “bola kekuatan”
2. Tambahkan tag “kekuatanplayer”
3. Jadikan “bolakekuatan” sebagai prefabs dengan mendrag dari hierarchy ke panel assets
4. Pada “player” Klik game object create empty child
5. Atur posisinya didepan player
6. Tambahkan script pada player

```
: Using unityengine;
```

```
Using systemcollections;
```

```
Public class kekuatan : MonoBehaviour { // deklarasi variable bola kekuatan
```

```
Public GameObject bolakekuatan;
```

```
Public GameObject Spawnpoint;
```

```
GameObject temp;
```

```
Float scale = 1 , maxscale=10;
```

```
Void Start () {
```

```
Bolakekuatan.GetComponent<MeshRenderer>().enabled=false;
```

```
}
```

```
// deteksi trigger kekuatanplayer pada penekanan tombol . keydown scale 0 kemudian  
selama keypres bertambah 0.1f per frame rate. Kekutan ditembakkan setelah tombol  
ctrl dilepas. Semakin lama tombol ditekan maka kekuatan semakin besar
```

```
If (Input.GetKey(KeyCode.LeftControl))
```

```
{
```

```
if (scale<=maxscale){
```

```
Scale+=0.1f;
```

```
}
```

```
}
```

```

If (input.getkeyup(keycode.leftcontrol)){
if (temp==null){
temp= (gameobject)
gameobject.instantiate(bolakekuatan,
spawnpoint.transfform.position,
spawnpoint.transfform.rotation);
temp.transform.localscale=newverctor3(scale,scale,scale);
Destroy(temp,2);
}
}
If (temp){
Temp.GetComponent<rigidbody>().AddRelativeForce(Vector3.forward*1000);
// menembak ke depan
}
}
}
7. Drag game object empty sebagai spawnpoint
8. Drag prefab sebagai bola kekuatan

```

XI. Scooring

Tugas : Buatlah game dengan scooring sesuai petunjuk X

1. Buatlah 3 Objek yang berposisi sebagai player, prefab dan sasaran Misal kamera, bola dan kubus
2. tambahkan game object empty sebagai child kamera
3. script 1 untuk kamera kemudian scrip 2 untuk kubus

script 1

```
using UnityEngine;
```

```
using System.Collections;
```

```
public class scrip1 : MonoBehaviour {
```

```
    public GameObject SpawnPoint;
```

```
    public GameObject Ball;
```

```
    GameObject shootingball;
```

```
    GameObject temp;
```

```
    GameObject camera1;
```

```
    bool goRight = true;
```

```
// Use this for initialization
```

```
void Start () {
```

```
    camera1 = GameObject.Find ("Main Camera");
```

```
}
```

```
// Update is called once per frame
```

```
void Update ()
```

```
{
```

```
    if (goRight) {
```

```
        camera1.transform.Rotate (0, 0.3f, 0);
```

```

        if (camera1.transform.eulerAngles.y > 0) {
            goRight = false;
        }
    } else {
        camera1.transform.Rotate (0, -0.3f, 0);
        if (camera1.transform.eulerAngles.y < 0) {
            goRight = true;
        }
    }
}

if (Input.GetKeyUp (KeyCode.Space)) {
    temp = (GameObject)
        GameObject.Instantiate (Ball,
            SpawnPoint.transform.position,
            SpawnPoint.transform.rotation);
    Destroy (temp, 5);
}

if (temp) {
    temp.GetComponent<Rigidbody> ().
        AddRelativeForce (Vector3.forward * 100);
}
}
}

```

```

using UnityEngine;
using System.Collections;

```

```

public class script2 : MonoBehaviour {
    public GameObject text3d;
    public string MyText;
    int score;
}

```

```

// Use this for initialization
void Start () {
    HideAgain ();
    text3d.GetComponent<TextMesh> ().text =
        MyText; score = 0;

}

// Update is called once per frame
void Update () {

}

void HideAgain () {
    text3d.GetComponent<MeshRenderer> ().enabled = false;
}

void OnTriggerEnter (Collider other) {
    if (other.gameObject.tag == "Ball") {
        Destroy (other.gameObject);
        score++;

        text3d.GetComponent<MeshRenderer> ().enabled = true;
        Invoke ("HideAgain", 3);
    }
}

void OnGUI () {
    GUILayout.BeginArea (new Rect (10, 10, 400, 200));
    GUILayout.Label ("Score : " + score.ToString ());
    GUILayout.EndArea ();
}
}

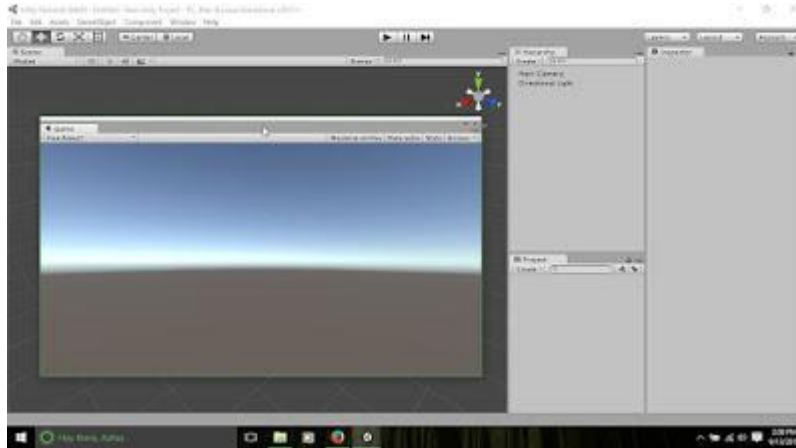
```


4. isikan parameter text 3d dengan game object 3d text dan My text “Berhasil “
5. isikan parameter spawnpoint dengan game object empty dan Ball dengan prefab bola
6. berikan tag Ball pada bola.

XII. Augmented Reality Dengan Vuforia

Tugas : Buatlah AR pada game anda sesuai petunjuk XI!

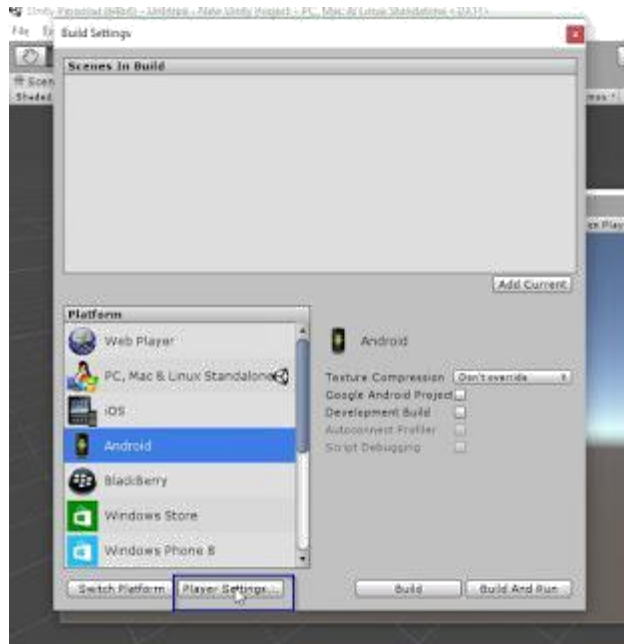
1. Buat project baru



2. Ubah project ke Android



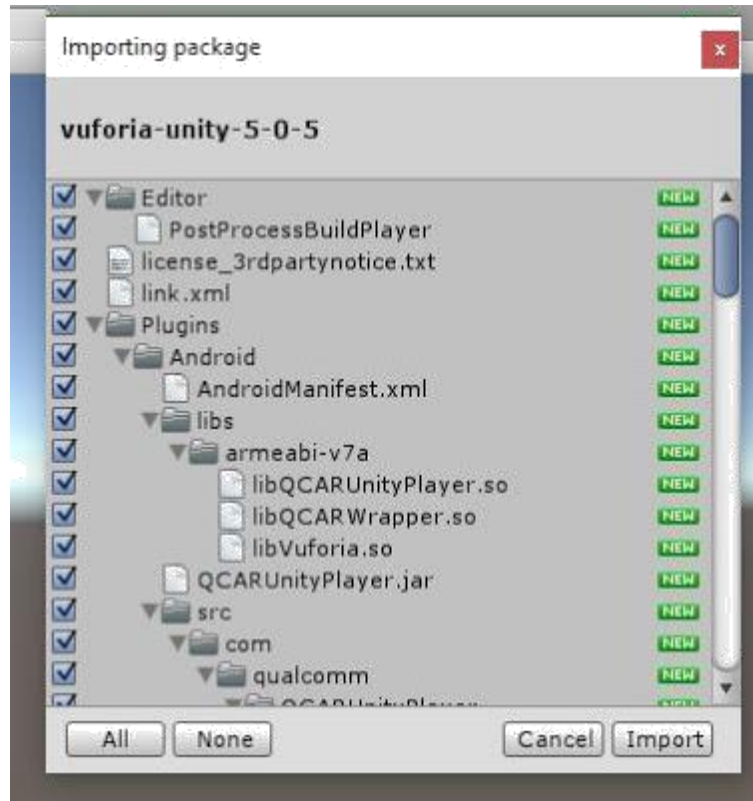
3. pengaturan pada player setting



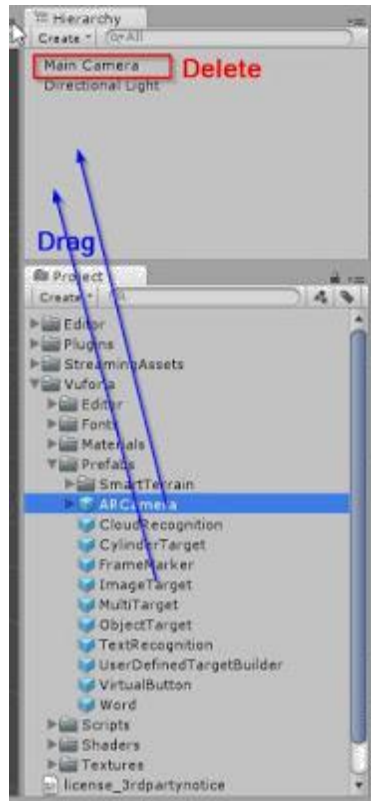
4. Atur nama company dan nama aplikasi juga pengaturan bundle identifier



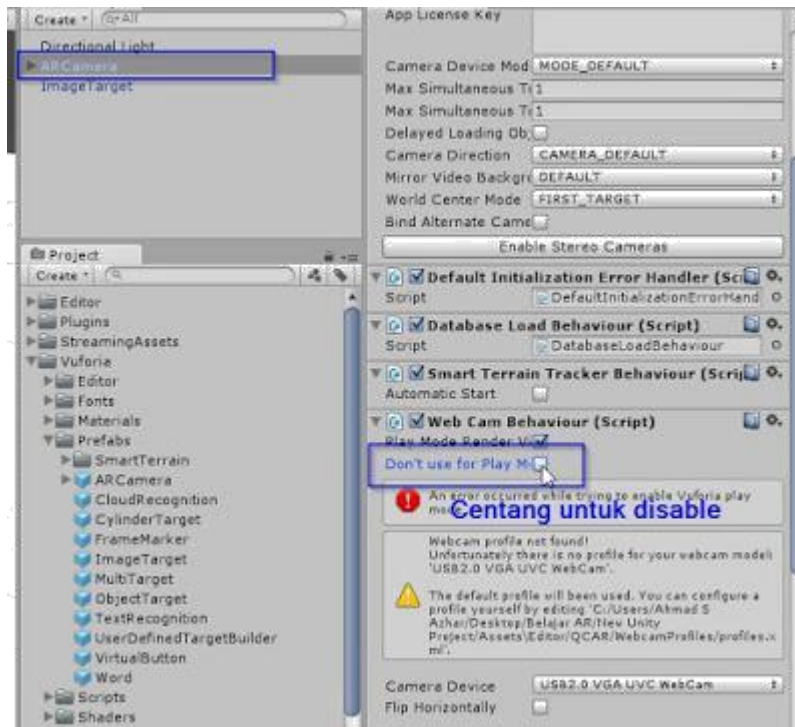
5. Import Vuforia SDK dengan double klik centang semua elemen import



6. Masukkan ARCamera dan ImageTarget ke dalam scene, delete Main Camera.

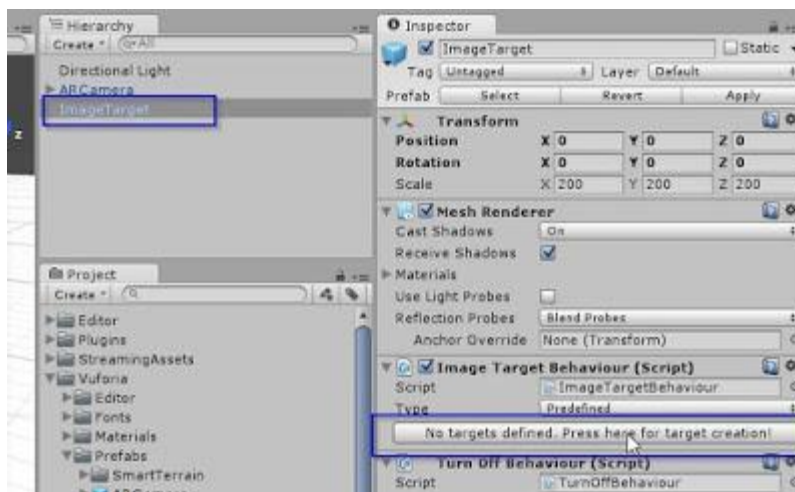


Vuforia mensupport debugging di PC menggunakan camera webcam. Tapi jika hardware belum di support Vuforia Anda dapat menonaktifkan fitur ini.

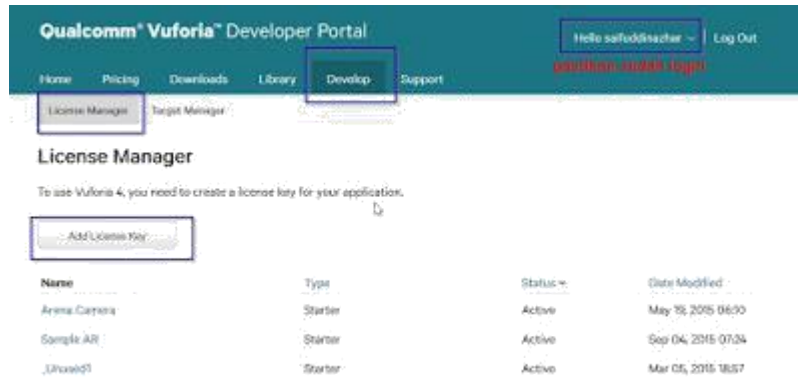


Disable 'Don't use play mode' jika error

7. Definisikan marker dengan klik ImageTarget kemudian klik 'No targets defined. Press here for target creation !'

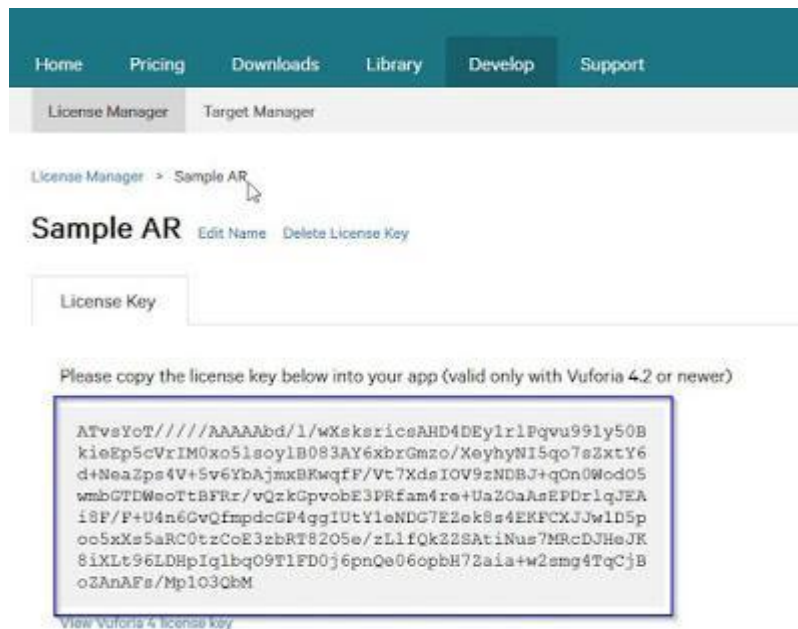


Anda akan diarahkan ke website Vuforia. Login untuk masuk ke akun Anda. Pertama klik 'Add License Key' pada 'License Manager' untuk membuat lisensi.

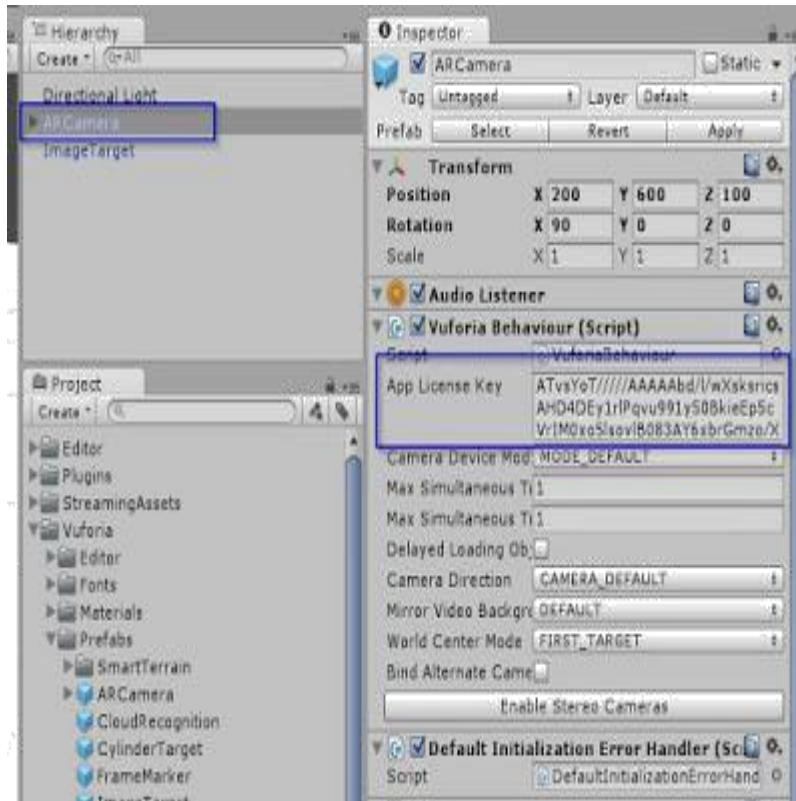


License manager pada Vuforia

Pada lisensi ada kode yang harus Anda letakkan di code Anda. Copy semua kode tadi kemudian masukkan di ARCamera pada project Anda



Masukkan ke ARCamera



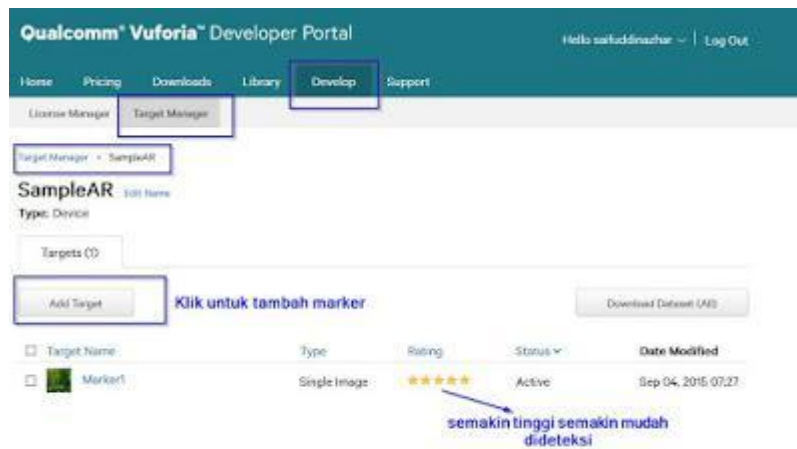
Kemudian definisikan marker pada image target. Pada bagian 'Target Manager' klik 'Add Database'.



Tambahkan database

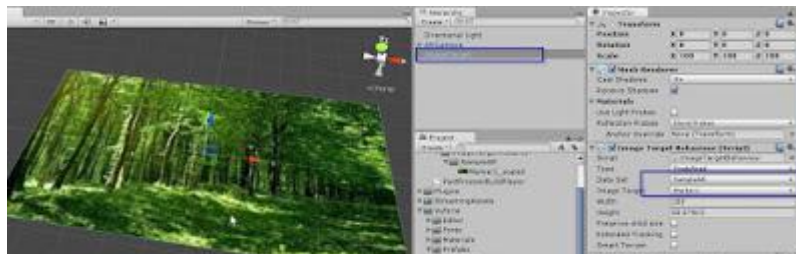
Masukkan nama database, kemudian upload marker. Marker yang Anda Upload akan

ditampilkan dan diberi rating

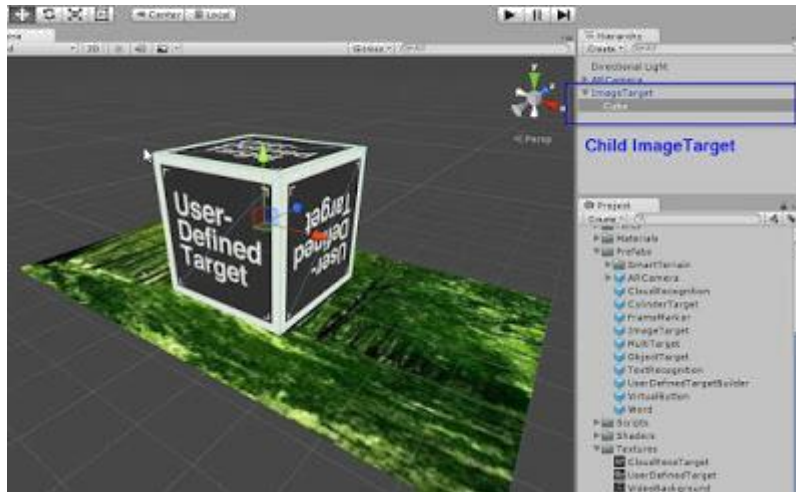


Setelah selesai download datasetnya, pilih 'Unity Editor'. Kemudian Anda akan mendapatkan file dengan ekstensi .unitypackage.

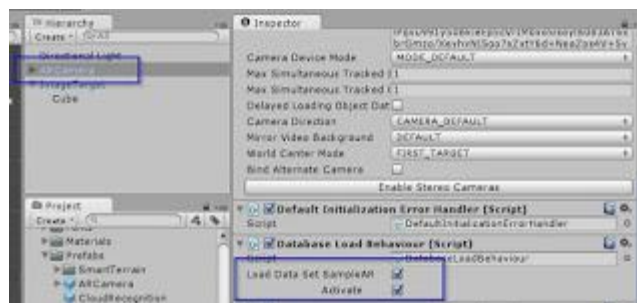
8. Import file dataset dengan double click file dataset dengan ekstensi .unitypackage yang Anda dapatkan dari Vuforia. Kemudian tentukan dataset yang digunakan pada ImageTarget



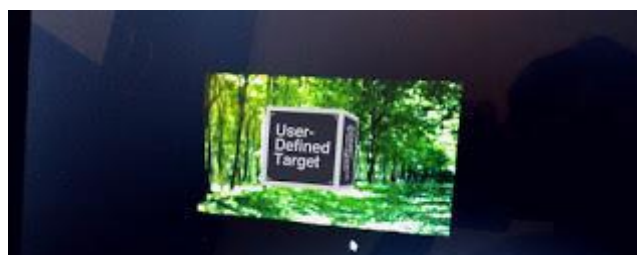
Letakkan object yang akan dimunculkan pada AR di atas Image Target. Buat object tersebut menjadi child dari ImageTarget. Pada contoh ini saya memakai object cube dengan klik GameObject >> 3D Object >> Cube



Konfigurasi ARCamera untuk me load dataset



9. Jalankan project dengan klik File >> Build Settings >> Build. Copy file .apk ke device Android Anda dan jalankan aplikasinya



Sumber: unity3d.com