

MODUL 7
USING CAMERA II

A. TUJUAN

- Mahasiswa mampu menggunakan dan mengimplementasikan pengolahan kamera dengan 3D Animation
- Mahasiswa mampu membuat telescopic camera yang memperbesar setiap kali tombol kiri mouse ditekan.
- Mahasiswa mampu membuat pandangan yang lebih luas dari tempat kejadian bisa sangat baik untuk navigasi dan informasi.
- Mahasiswa mampu menggunakan Render Tekstur untuk membuat kamera pengintai dalam game yang mentransmisikan videonya ke monitor.

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 : 3 jam pelajaran

D. DASAR TEORI

Secara default Scene anda memang hanya memiliki satu buah Camera. Namun Unity tidak pernah membatasi untuk dapat memiliki lebih dari satu camera. Tentu saja diperlukan sebuah pengaturan khusus agar masing-masing camera dapat berjalan dan tidak saling tumpang tindih. Pada minggu ini akan mempelajari beberapa hal untuk membuat dan menentukan fungsi camera dengan fitur-fitur camera yang ada pada Unity, antara lain:

1. ZOOMING A TELESCOPIC CAMERA

Dalam proyek ini, kita akan membuat kamera telescopic yang memperbesar setiap kali tombol kiri mouse ditekan. Hal ini dapat sangat berguna, misalnya, jika kita memiliki penembak jitu dalam permainan kami.

2. DISPLAYING A MINI-MAP

Dalam banyak permainan, pandangan yang lebih luas dari tempat kejadian bisa sangat berharga untuk navigasi dan informasi. Mini-Map yang besar untuk memberikan pemain yang perspektif tambahan yang mungkin mereka butuhkan ketika di pertama atau orang ketiga.

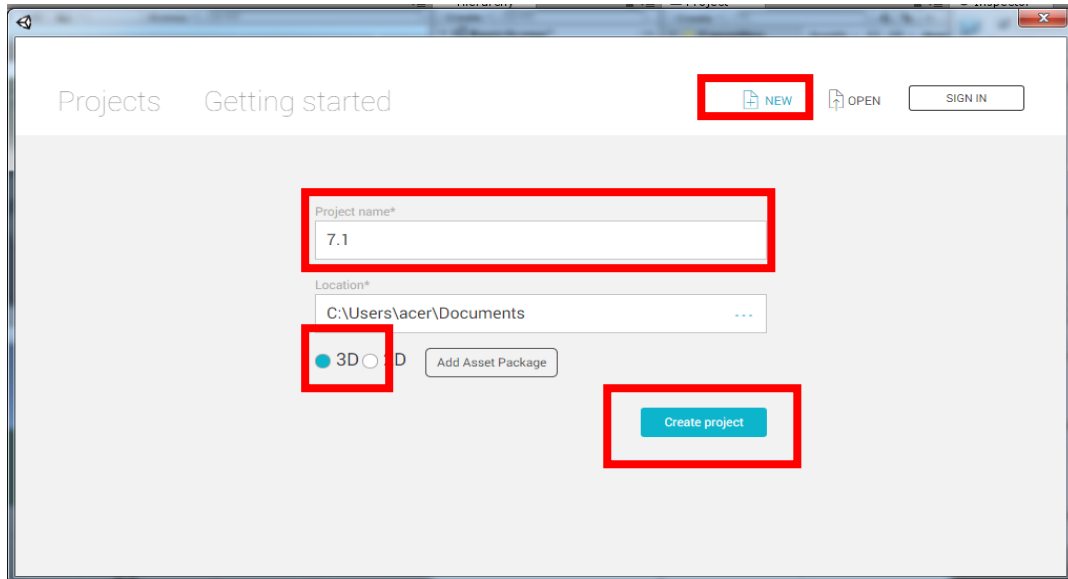
3. CREATING AN IN-GAME SURVEILLANCE CAMERA

Meski menggunakan viewport kedua dapat berguna dalam banyak situasi, akan ada saat-saat ketika Anda perlu output gambar yang diberikan dari kamera untuk tekstur pada saat runtime. Untuk menggambarkan hal ini kita akan menggunakan Render Texture untuk membuat kamera pengintai dalam game yang mentransmisikan videonya ke monitor.

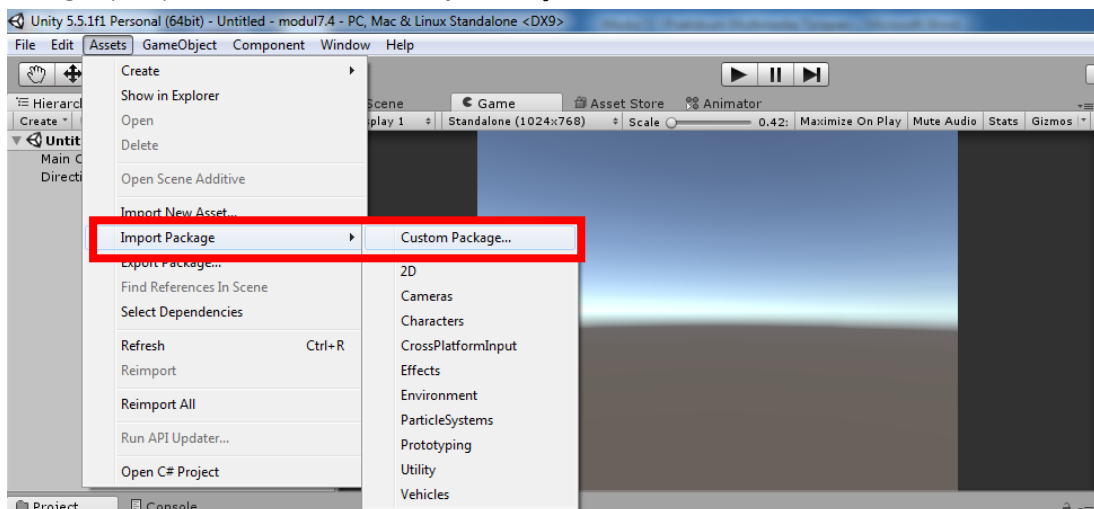
LATIHAN PRAKTIKUM

E. ZOOMING A TELESCOPIC CAMERA

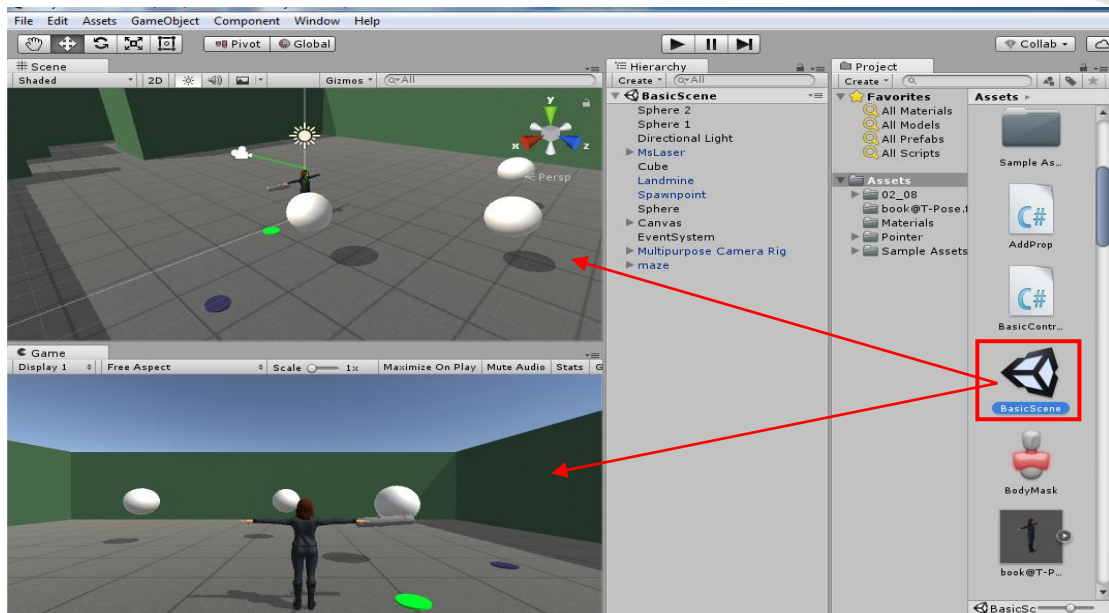
1. Pada form pertama pilih **New** → **Project Name (7.1)** → **3D** → **create Project**.



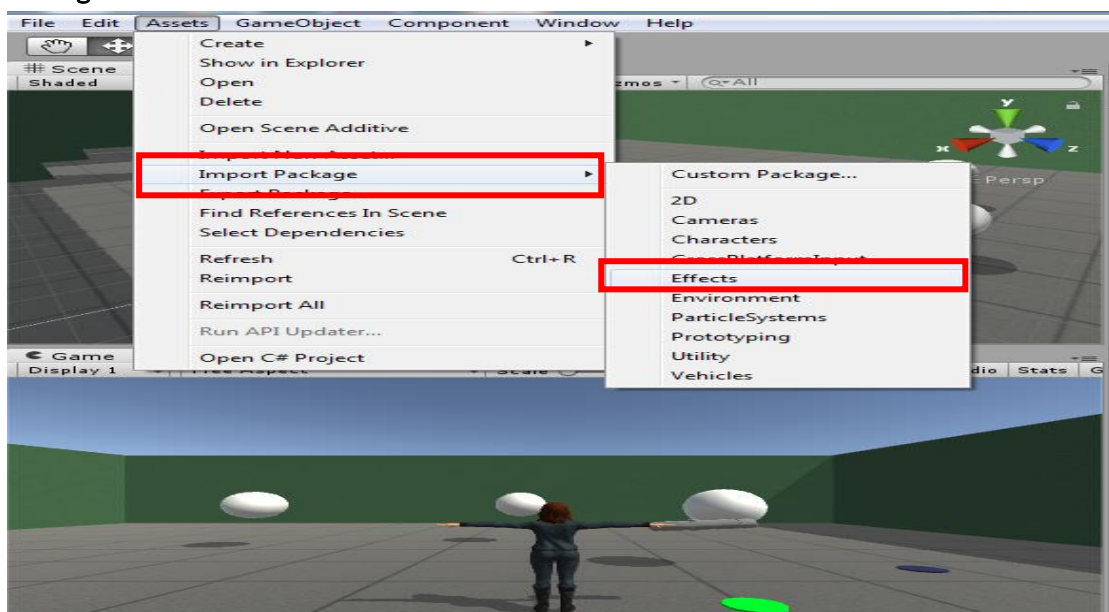
2. Untuk **Projek** ini, telah disiapkan **BasicScene Unity Package**, yang berada pada folder 1362_05_codes.
3. Mengimpor paket BasicScene menjadi **Project** baru.



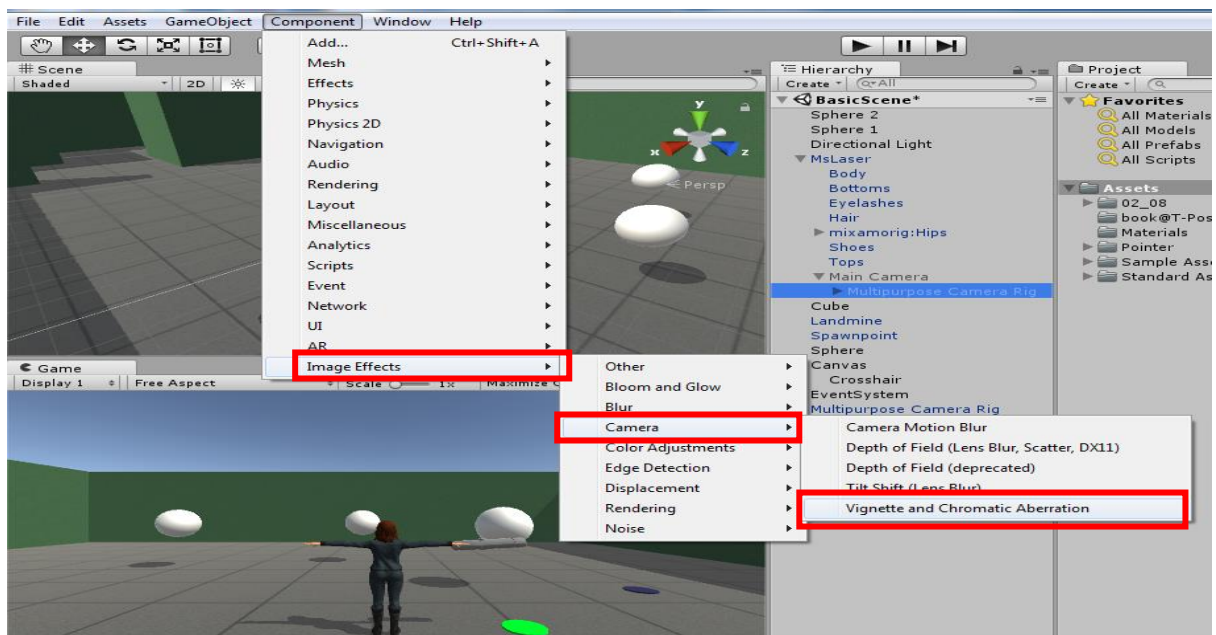
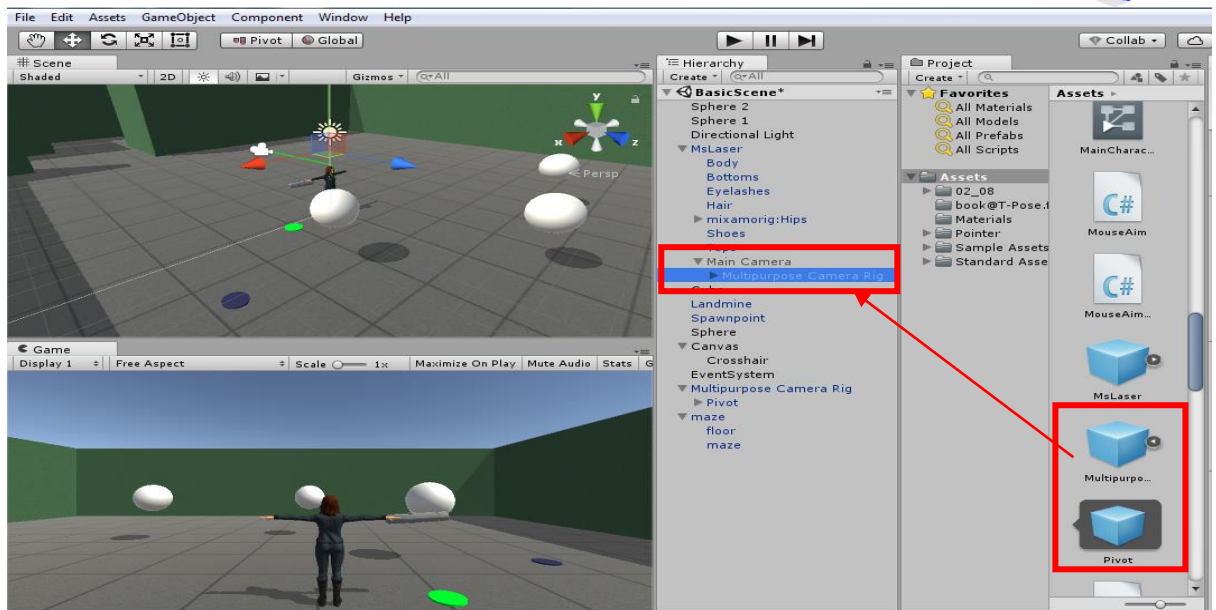
4. Dari tampilan **Project** Ini, ditampilkan sebuah karakter animasi dan beberapa geometri tambahan. bukalah **BasicScene** yang berada di **Asset (Klik 2 kali)**.



5. Mengimpor unity **Effects Package** dengan cara memasukkan ke Asset → Import Package → Effects



6. Pilih **Main Kamera** GameObject dalam **Multipurpose Camera Rig** → **Pivot** GameObject dan menerapkan efek gambar **Vignette** (dengan menavigasi ke **Component** → **Image Effects** → **Camera** → **Vignette and Chromatic Aberration**).



7. Pada tampilan Projek buatlah script dengan klik menu drop-down dan pilih C# Script. Ubahlah rename menjadi TelescopicView dan membukanya di editor.
8. Masukkan Script dibawah ini ke editor:

```

using UnityEngine;
using System.Collections;
using UnityEngine.StandardAssets.ImageEffects;

public class TelescopicView : MonoBehaviour{
    public float zoom = 2.0f;
    public float speedIn = 100.0f;
    public float speedOut = 100.0f;
    private float initFov;
    private float currFov;
    private float minFov;
    private float addFov;
    private VignetteAndChromaticAberration v;
    public float vMax = 10.0f;

    void Start(){
        initFov = Camera.main.fieldOfView;
        minFov = initFov / zoom;
        v = this.GetComponent<VignetteAndChromaticAberration>()
        as VignetteAndChromaticAberration;
    }

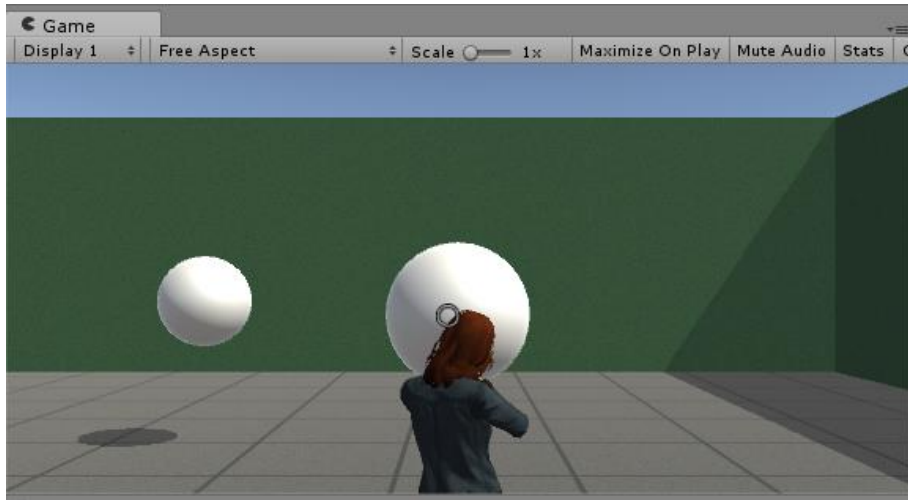
    void Update(){
        if (Input.GetKey(KeyCode.Mouse0))
            ZoomView();
        else
            ZoomOut();
        float currDistance = currFov - initFov;
        float totalDistance = minFov - initFov;
        float vMultiplier = currDistance / totalDistance;
        float vAmount = vMax * vMultiplier;
        vAmount = Mathf.Clamp (vAmount,0,vMax);
        v.intensity = vAmount;
    }

    void ZoomView(){
        currFov = Camera.main.fieldOfView;
        addFov = speedIn * Time.deltaTime;
        if (Mathf.Abs(currFov - minFov) < 0.5f)
            currFov = minFov;
        else if (currFov - addFov >= minFov)
            currFov -= addFov;
        Camera.main.fieldOfView = currFov;
    }

    void ZoomOut()
    {
        currFov = Camera.main.fieldOfView;
        addFov = speedOut * Time.deltaTime;
        if (Mathf.Abs(currFov - initFov) < 0.5f)
            currFov = initFov;
        else if (currFov + addFov <= initFov)
            currFov += addFov;
        Camera.main.fieldOfView = currFov;
    }
}

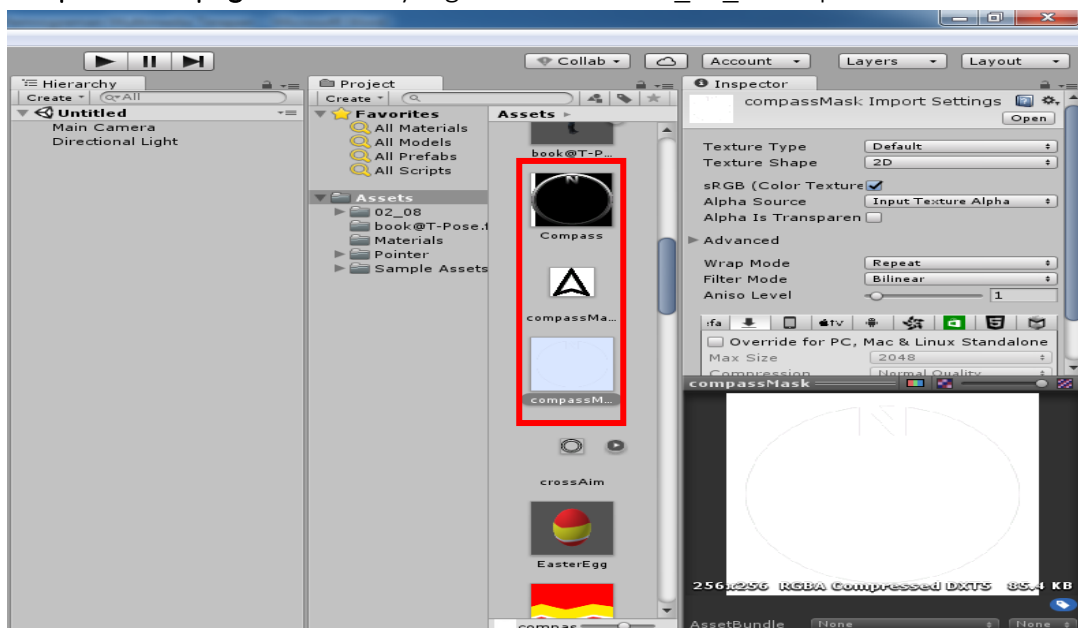
```

9. Simpan script dan drag ke **Main Kamera** GameObject dalam **Multipurpose Camera Rig** → **Pivot** GameObject.
10. Jalankan projek anda dan lihat efek sketsa animasi zooming dengan mouse:

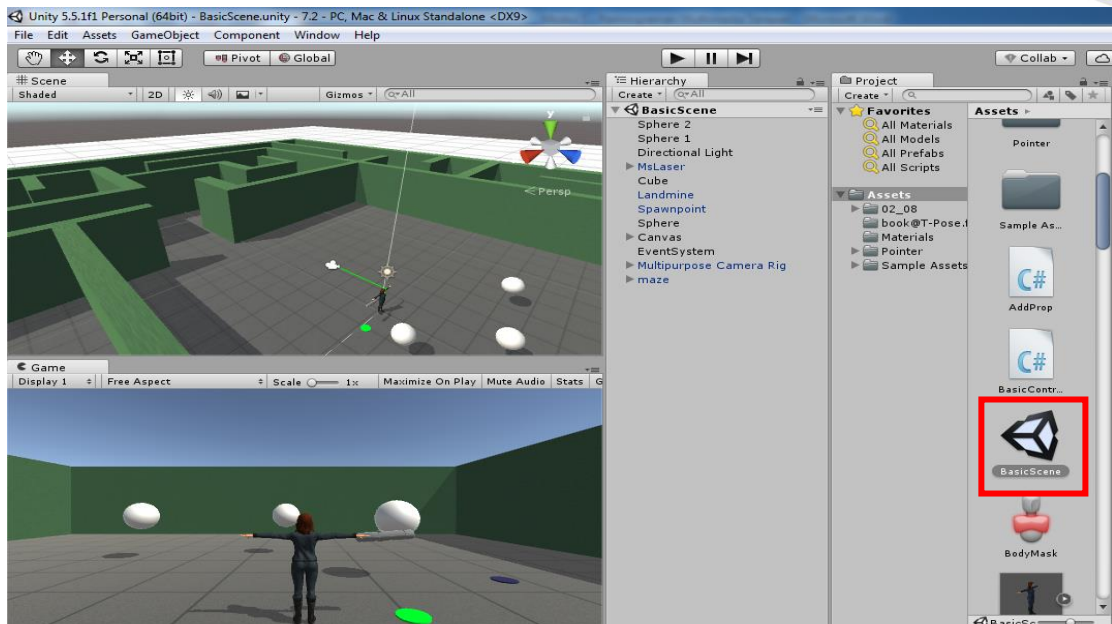


F. DISPLAYING A MINI-MAP

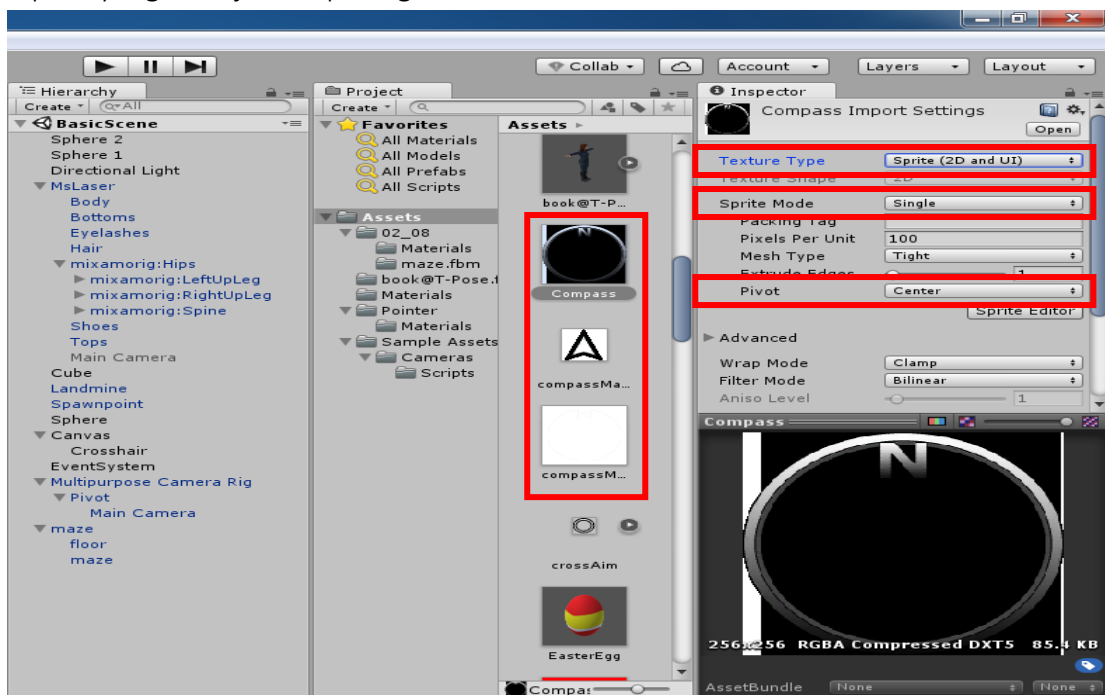
1. Untuk projek ini, telah disiapkan BasicScene Unity Package dan diperlukan juga mengimpor tiga file gambar bernama **Compass.png**, **compass Marker.png**, dan **compass Mask.png**. Semua file yang tersedia di 1362_05_05 map.



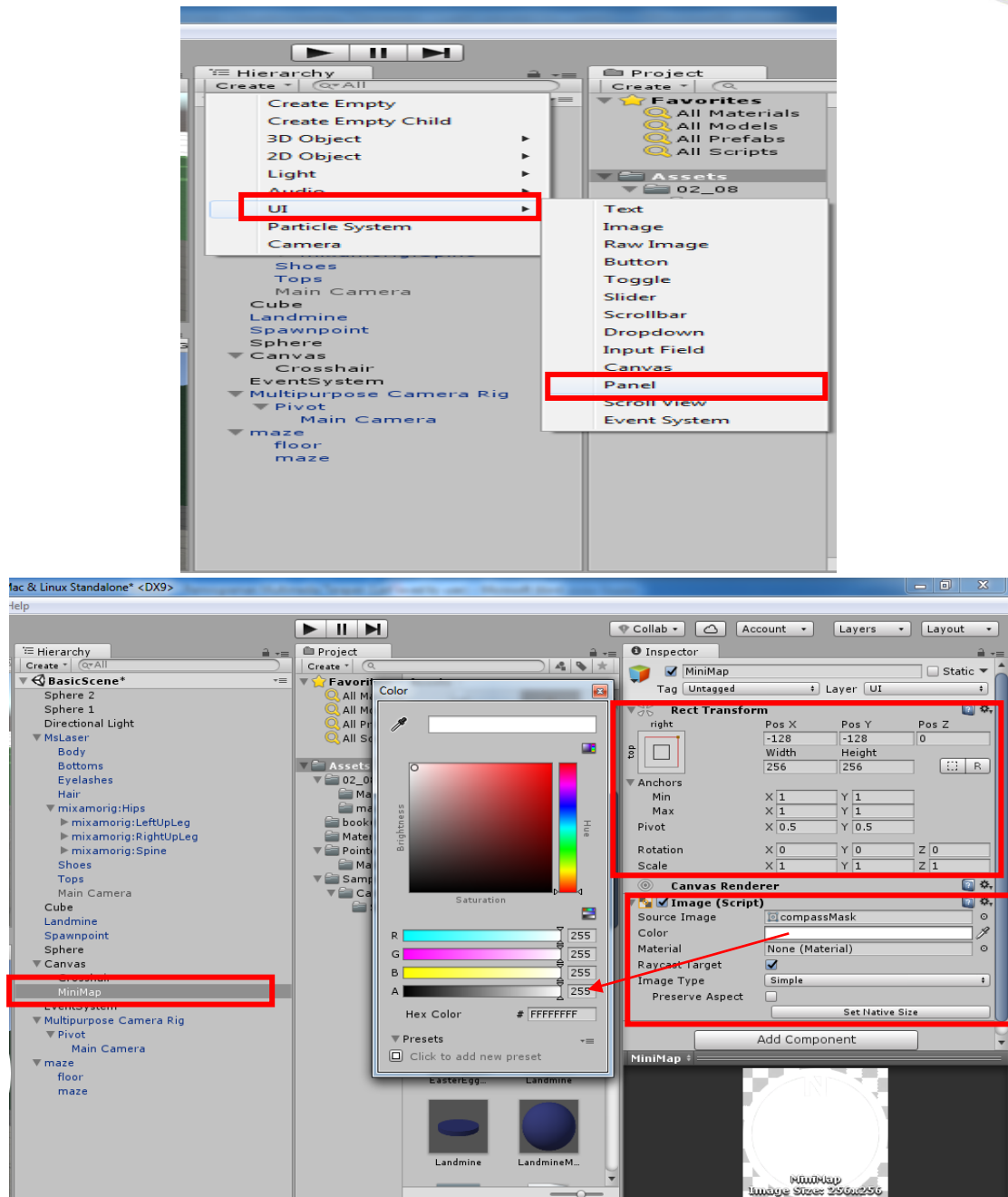
2. Mengimpor paket BasicScene menjadi **Project** baru. Juga, mengimpor file **.png** yang disediakan. Membuka tingkat BasicScene.



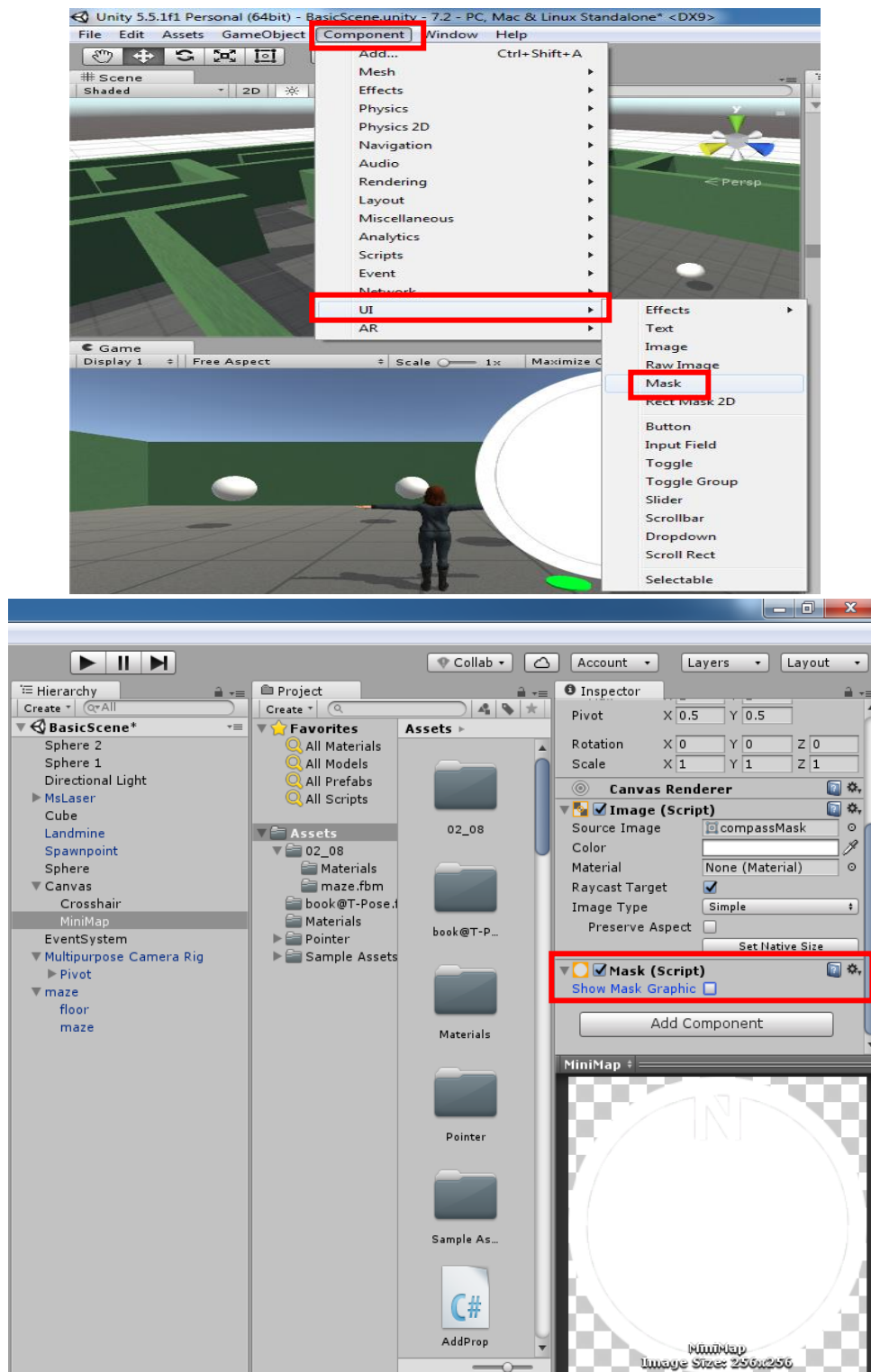
3. Dari tampilan **Assets**, pilih file tekstur **Compass**, **CompassMarker**, dan **compassMask**. Kemudian, di **Inspector** rubahlah Jenis **Tekstur** untuk **Sprite (2D dan UI)**, **Mode Sprite** sebagai **Single** dan **Pivot** dirubah **Center**. Klik **Apply** untuk mengkonfirmasi perubahan, seperti yang ditunjukkan pada gambar berikut:



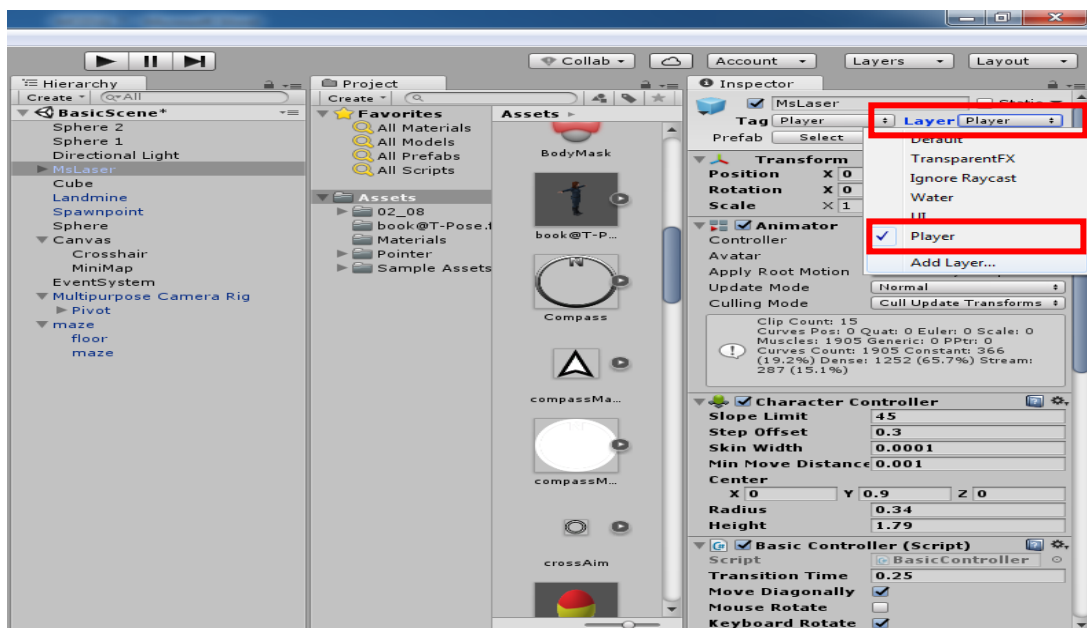
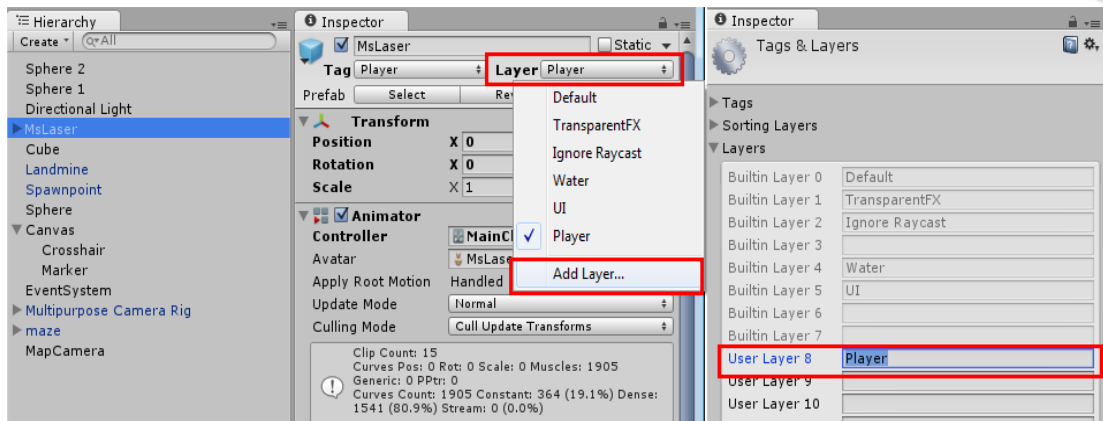
4. Dari tampilan **Hierarchy**, membuat objek **UI Panel** baru (**Create**→**UI**→**Panel**). Ini akan dibuat sebagai bagian dari **UI Canvas** **GameObject**. Gantilah namanya dengan **MiniMap**. Kemudian, dari tampilan **Inspector**, mengatur penyalarsan ke **Top / Right**, mengubah baik **Width** dan **Height** ke 256, dan yang bidang **Pos X** dan **Pos Y** ke -128. Juga, mengisi bidang **Source Image**, dalam komponen **Image**, dengan sprite **compassMask**, menyesuaikan bidang **Color** dengan membawa **Alpha** hingga 255, seperti yang ditunjukkan pada gambar berikut:



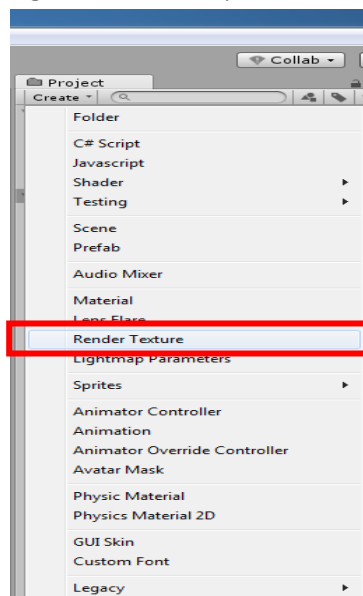
5. Menambahkan komponen **Mask** untuk **MiniMap** (dari menu utama, pilih **Component** → **UI** → **Masker**). Kemudian, dari tampilan **Inspector**, ditemukan komponen **Mask** dan hapus centang **Show Masker Graphic** (dimaksudkan menjadi tidak terlihat dan melayani sebagai mask untuk mini-peta).

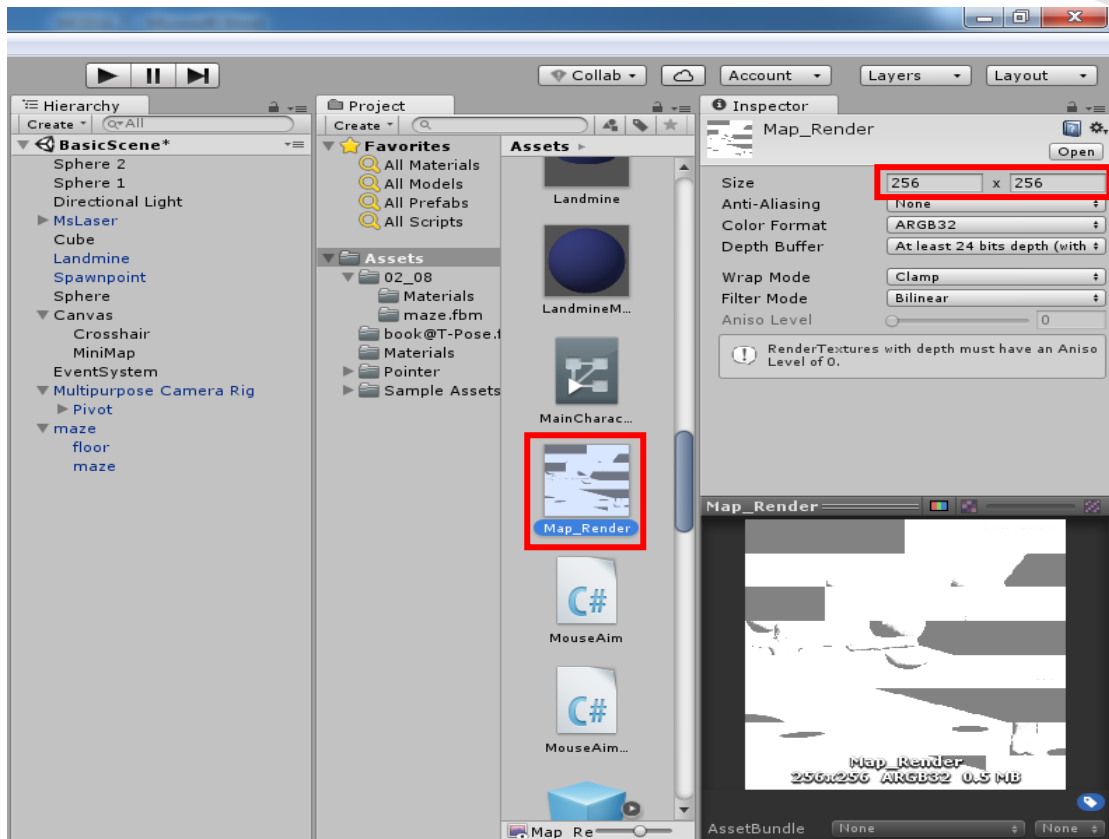


6. Pilih **MsLaser** GameObject (merupakan karakter pemain), bukalah tampilan **Inspector** lalu mengakses menu drop-down **Layer**. Pilih **Add Layer...** dan kemudian **User Layer** beri nama **Player** selanjutnya Pilih karakter **MsLaser** lagi dan **Layer** menu drop-down, pilih **Player**, seperti yang ditunjukkan pada gambar berikut:

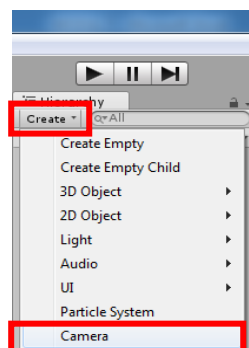


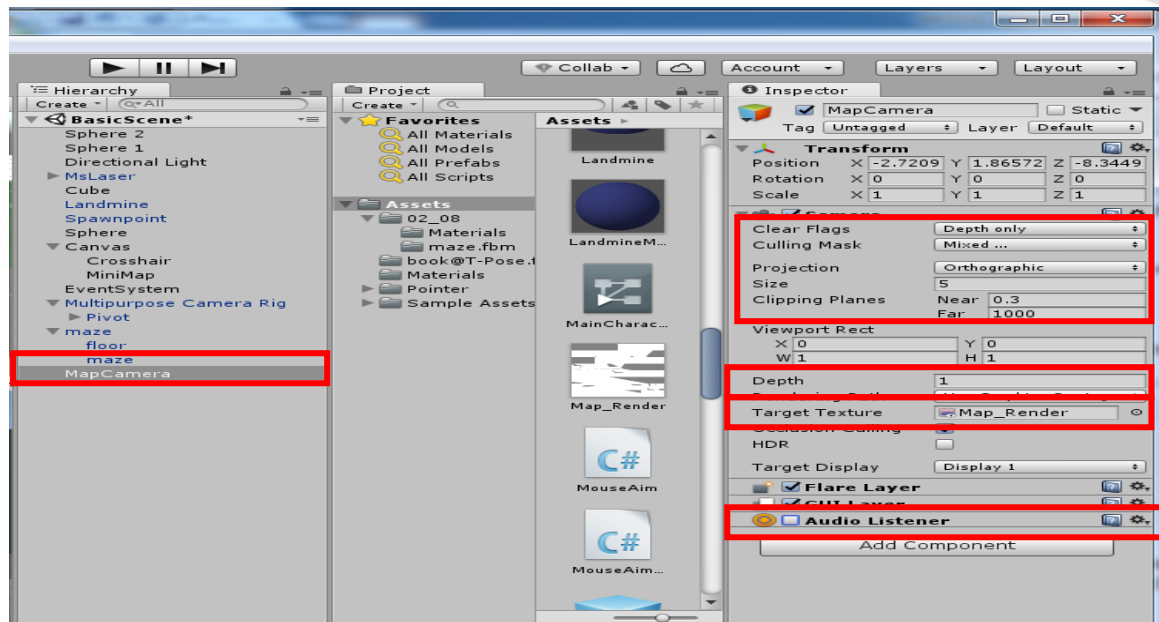
7. Dari tampilan **Project**, buatlah baru Render Texture dan nama itu **Map_Render**. Kemudian, dari **Inspector**, mengubah ukurannya ke **256 x 256**.



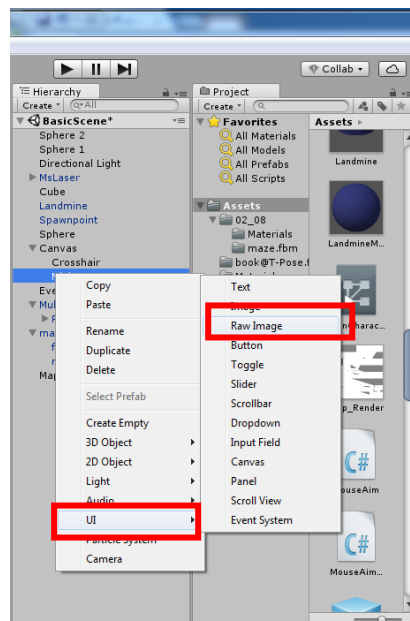


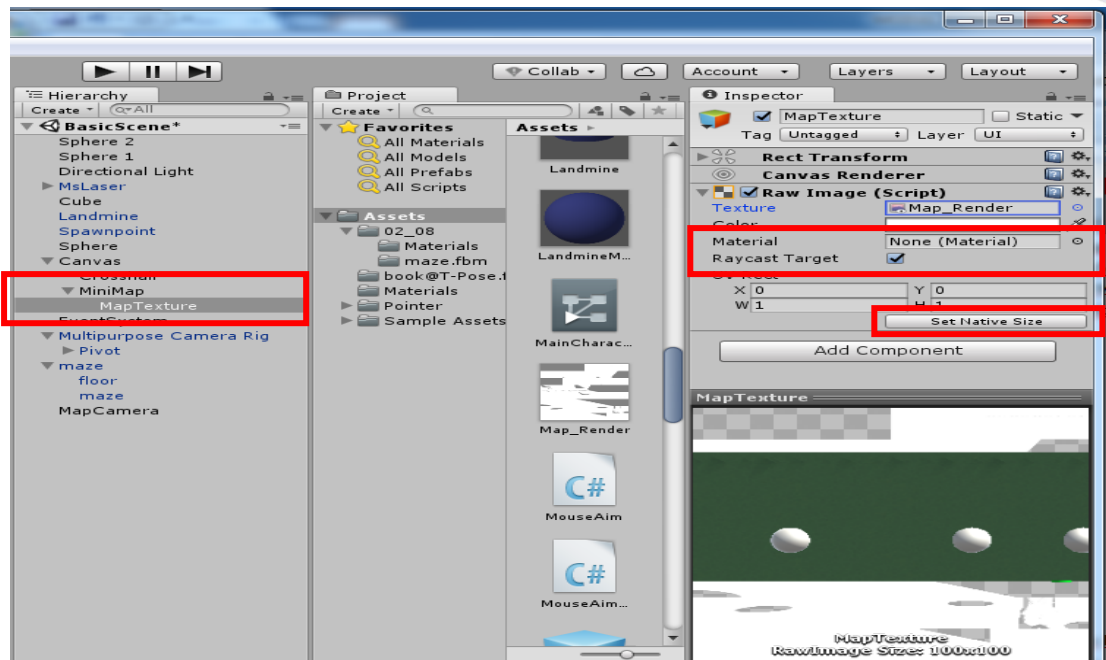
8. Dari tampilan **Hierarchy**, membuat kamera baru (**Create → Camera**) dan ganti nama dengan **MapCamera**. Dari tampilan Inspector, mengubah parameter sebagai berikut (ditunjukkan pada gambar yang akan mengikuti):
 - a. **Clear Flags**: Depth Only
 - b. **Culling Mask**: Mixed ... (unselected **Player**)
 - c. **Projection**: Orthographic
 - d. **Depth**: 1 (or higher)
 - e. **Target Texture**: Map_Render
 - f. Juga, hapus centang pada kamera komponen **Audio Listener**



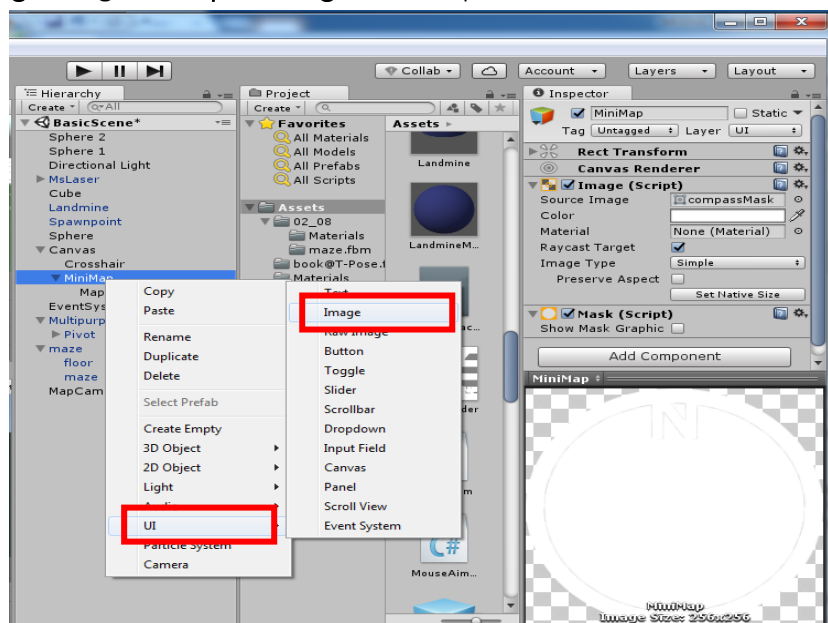


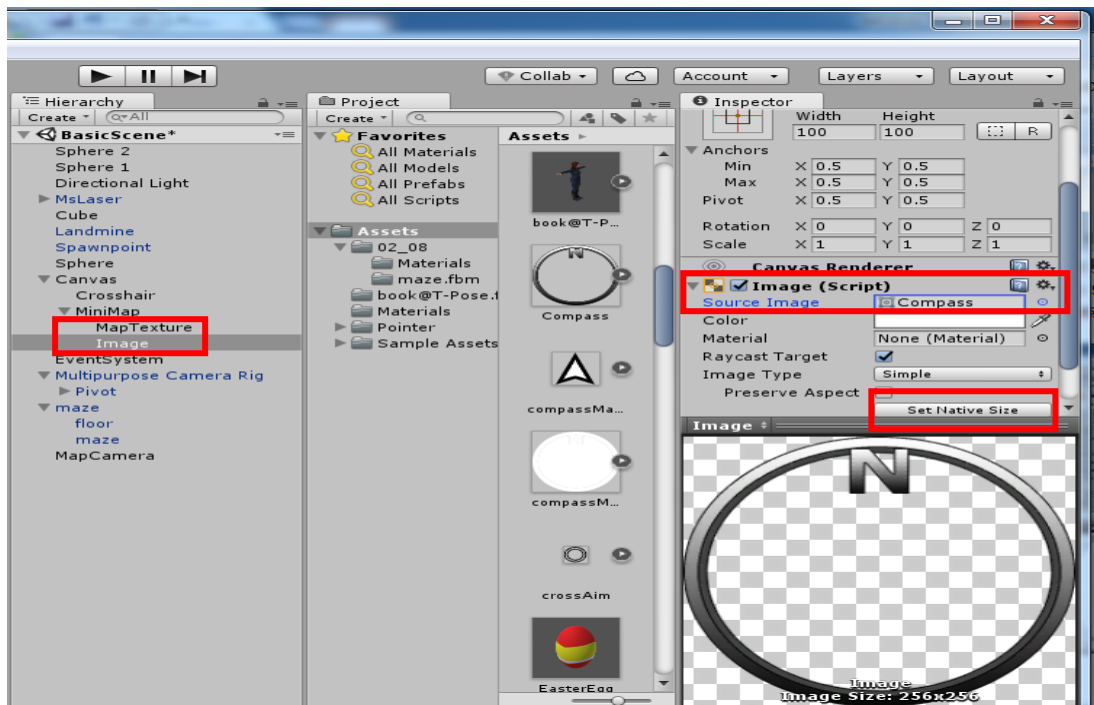
9. Di tampilan **Hierarchy**, klik kanan pada **MiniMap** dan arahkan ke **UI** → **Raw Image** untuk membuat elemen UI kecil. Ganti nama menjadi **MapTexture**. Kemudian, dari tampilan **Inspector**, isilah bidang **Texture** dengan tekstur **Map_Render** dan klik pada tombol **Set Native Size**, seperti yang ditunjukkan pada gambar berikut:



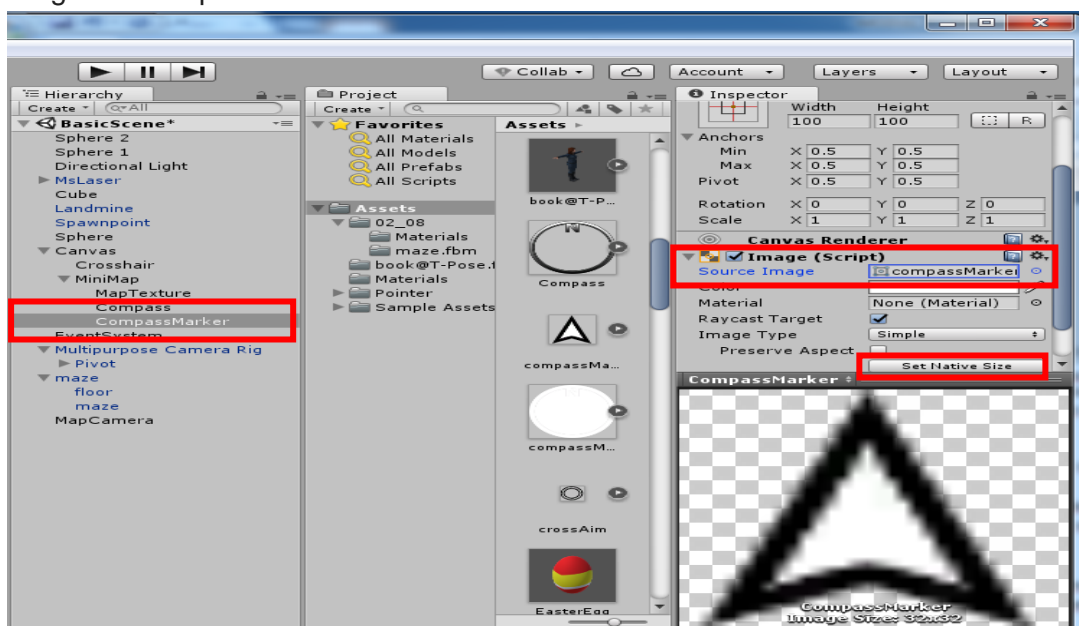


10. Selanjutnya klik-kanan pada **MiniMap** dan arahkan ke UI → Image untuk membuat elemen anak lain. Nama itu Compass. Kemudian, dari tampilan **Inspector**, mengisi **Source Image** dengan **Compass Image** dan klik pada tombol **Set Native Size**.





11. selanjutnya klik kanan pada **MiniMap** dan arahkan ke **UI → Image** untuk menambahkan elemen kecil lain. Gantilah namanya menjadi **Marker**. Kemudian, dari tampilan **Inspector**, isilah sumber **Source Image** dengan **compassMarker** image dan klik pada tombol **Set Native Size**.

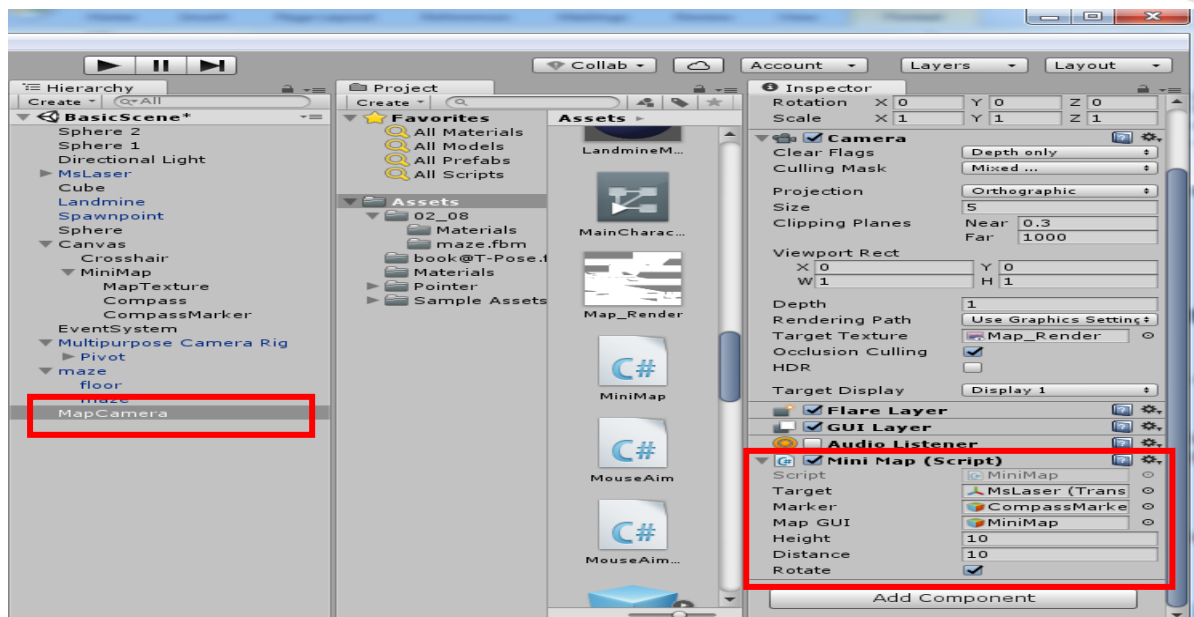


12. Dari tampilan Project, membuat baru C # Script dan nama itu MiniMap. Buka dan mengganti semuanya dengan kode berikut:

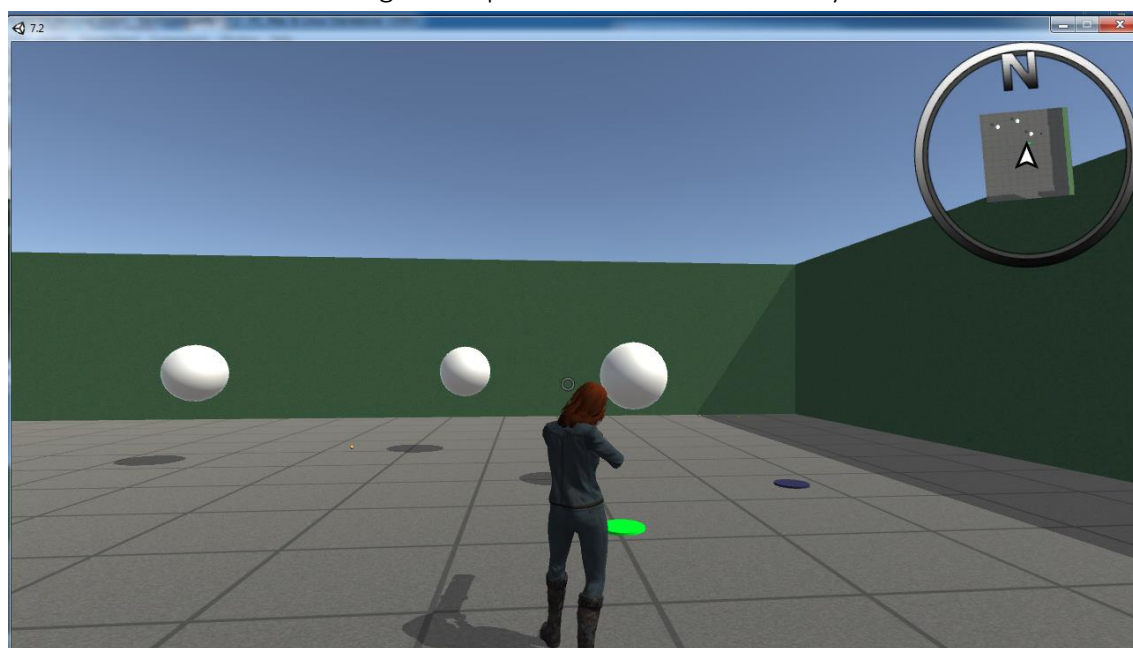

```
using UnityEngine;
using UnityEngine.UI;
using System.Collections;
public class MiniMap : MonoBehaviour
{
    public Transform target;
    public GameObject marker;
    public GameObject mapGUI;
    public float height = 10.0f;
    public float distance = 10.0f;
    public bool rotate = true;
    private Vector3 camAngle;
    private Vector3 camPos;
    private Vector3 targetAngle;
    private Vector3 targetPos;
    private Camera cam;
    void Start(){
        cam = GetComponent<Camera> ();
        camAngle = transform.eulerAngles;
        targetAngle = target.transform.eulerAngles;
        camAngle.x = 90;
        camAngle.y = targetAngle.y;
        transform.eulerAngles = camAngle;
    }
    void Update(){
        targetPos = target.transform.position;
        camPos = targetPos;
        camPos.y += height;
        transform.position = camPos;
        cam.orthographicSize = distance;
        Vector3 compassAngle = new Vector3();
        compassAngle.z = target.transform.eulerAngles.y;
        if (rotate)
        {
            mapGUI.transform.eulerAngles = compassAngle;
            marker.transform.eulerAngles = new Vector3();
        }
        else
        {
            marker.transform.eulerAngles = -compassAngle;
        }
    }
}
```

13. Simpan script dan drag ke **MapCamera**. Kemudian dari tampilan **Inspector**, mengubah parameter dari komponen **Mini Map** sebagai berikut (ditunjukkan pada screenshot yang akan mengikuti):

- a. **Target:** MsLaser
- b. **Marker:** Marker/CompassMarker
- c. **Map GUI:** MiniMap (panel UI dibuat sebelumnya)
- d. **Height:** 10
- e. **Distance:** 10
- f. **Rotate:** Diperiksa

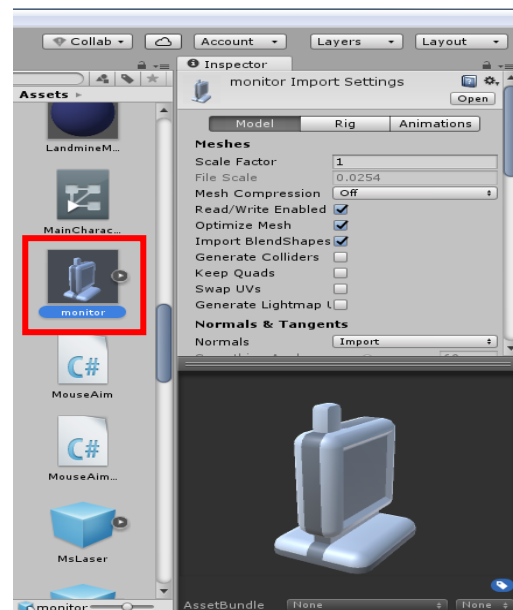
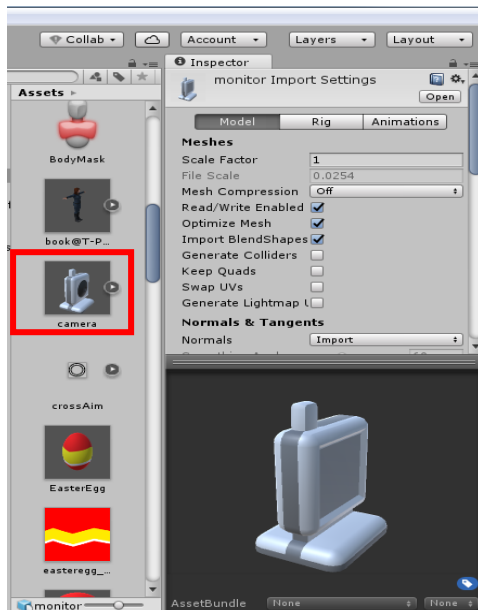


14. Jalankan dan lihat fungsi mini-peta di sudut kanan atas layar:



G. Creating an in-game surveillance camera

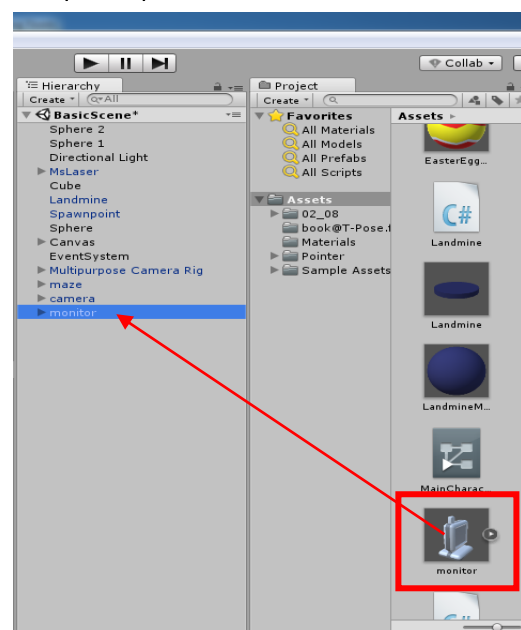
1. Untuk proyek ini, telah disiapkan paket **BasicScene** Unity dan juga dua model **FBX 3D** untuk **monitor** dan **kamera** benda. Paket ini di folder **1362_05_codes**, dan model 3D dalam **1362_05_06** folder.

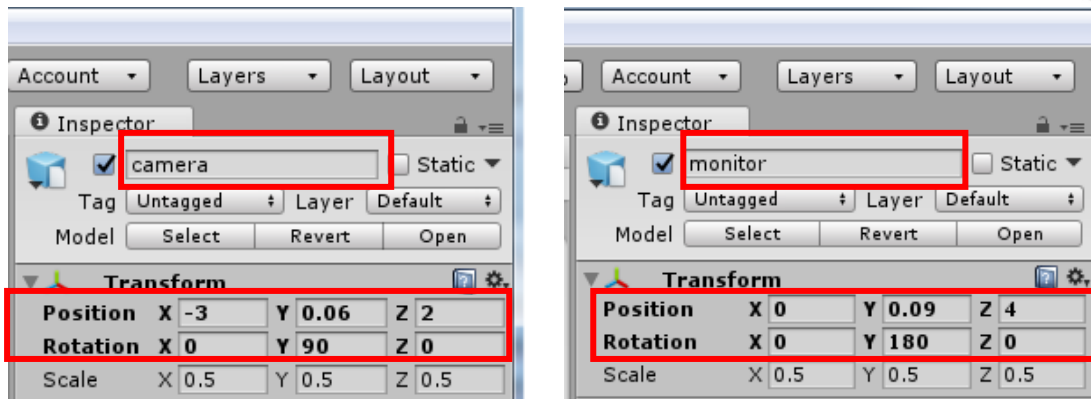


2. Dari tampilan Proyek bukalah **BasicScene** dengan klik 2 kali. Ini adalah adegan dasar menampilkan karakter animasi dan beberapa geometri tambahan.
3. Dari tampilan **Proyek**, mengimpor **Monitor** dan **Camera** ke Hierarchy dengan mendrag ke tempat Hierarchy. Pengaturan **Transform** mereka harus (ditunjukkan pada gambar berikut):

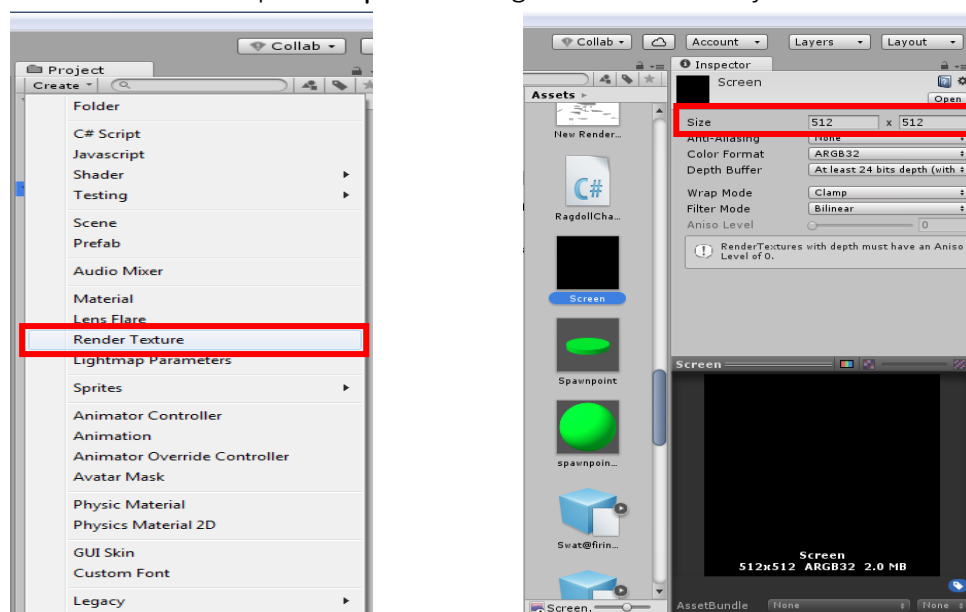
Monitor: Position: X: 0; Y: 0,09; Z: 4. Rotation: X: 0; Y: 180; Z: 0.

Camera: Position: X: -3; Y: 0,06; Z: 4. Rotation: X: 0; Y: 90; Z: 0.





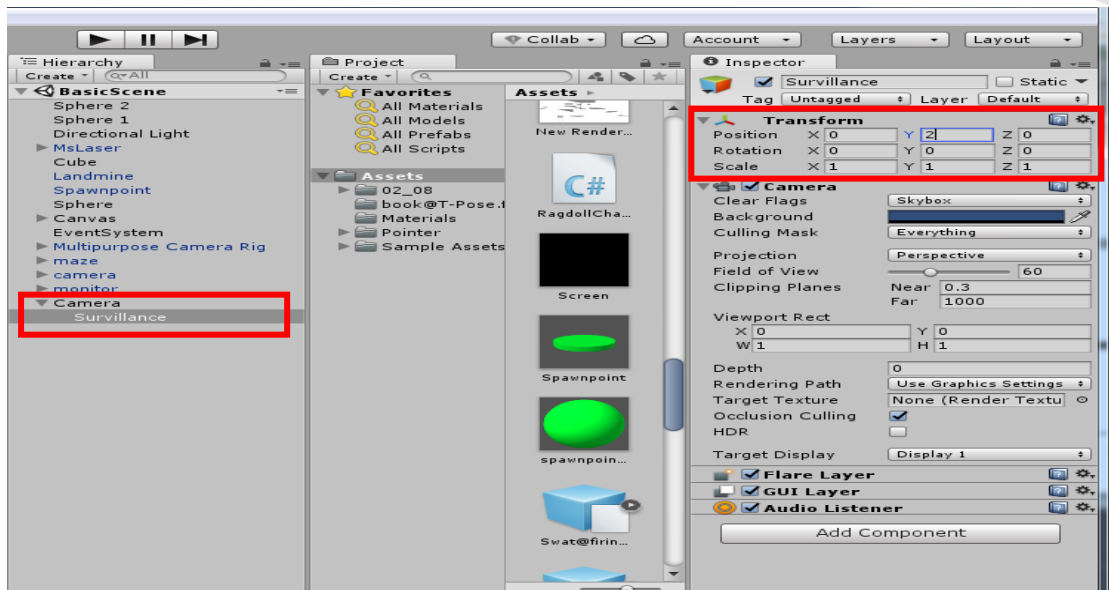
4. Buatlah didalam **Project** dengan memilih **Render Textur** dan rubah nama itu menjadi **Screen**. Kemudian dari tampilan **Inspector** mengubah ukuran menjadi **512 x 512**.



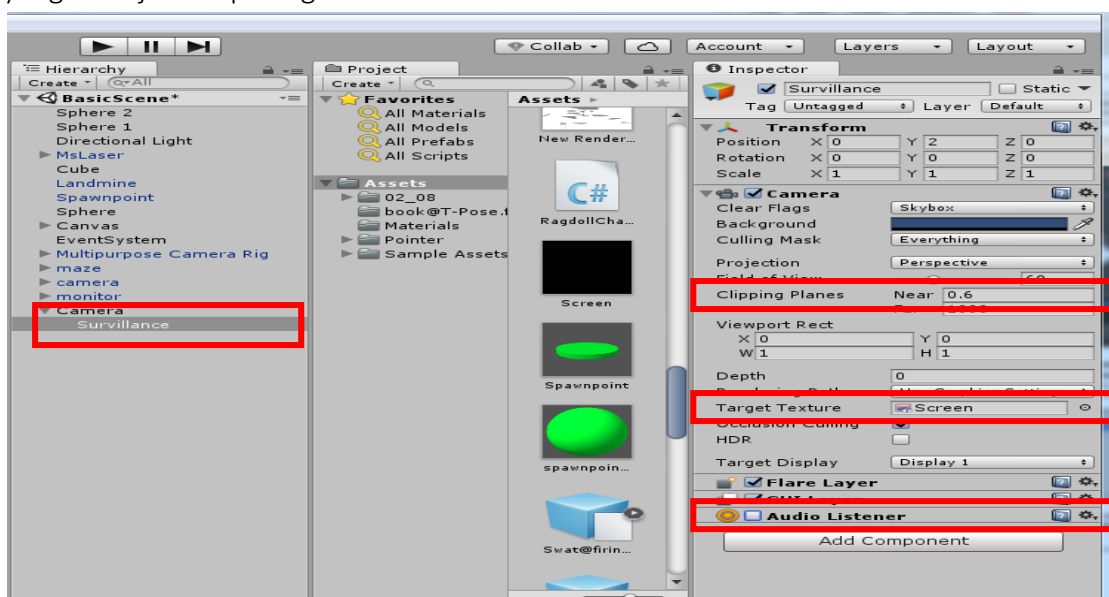
5. Tambahkan **Camera** baru ke **Hierarchy** melalui menu drop-down di atas tampilan **Hierarchy** (**Create** → **Camera**). Kemudian, dari tampilan **Inspector**, nama itu **Surveillance** dan membuatnya menjadi child dari **Camera** GameObject. Kemudian, mengubah nya **Transform** pengaturan sebagai berikut:

Position: X: 0; Y: 2; Z: 0

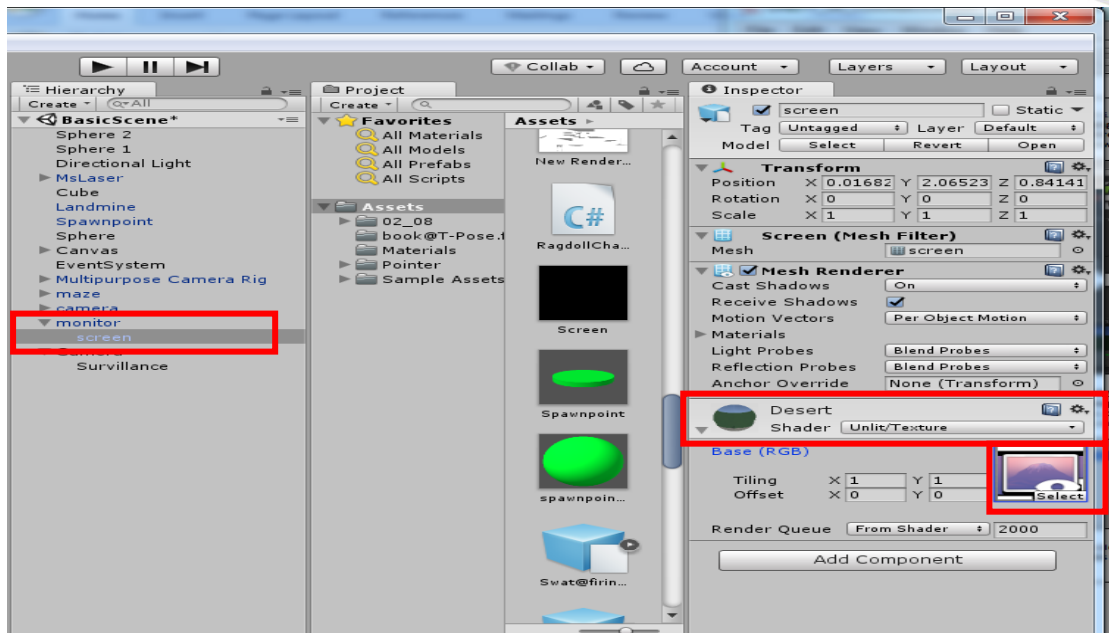
Rotation: X: 0; Y: 0; Z: 0.



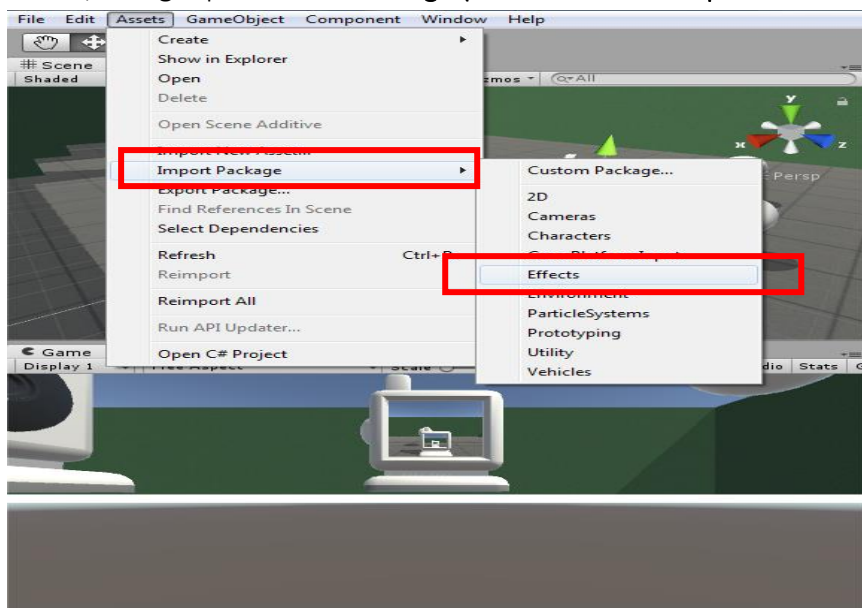
6. Selanjutnya memilih **Camera Surveillance** yang telah Anda buat dan dari tampilan Inspector ubahlah **Clipping Planes** → **Near** ke 0,6. Juga mengisi slot **Target Texture** dengan layar **Render Texture** dan menonaktifkan komponen **Audio Listener**, seperti yang ditunjukkan pada gambar berikut:



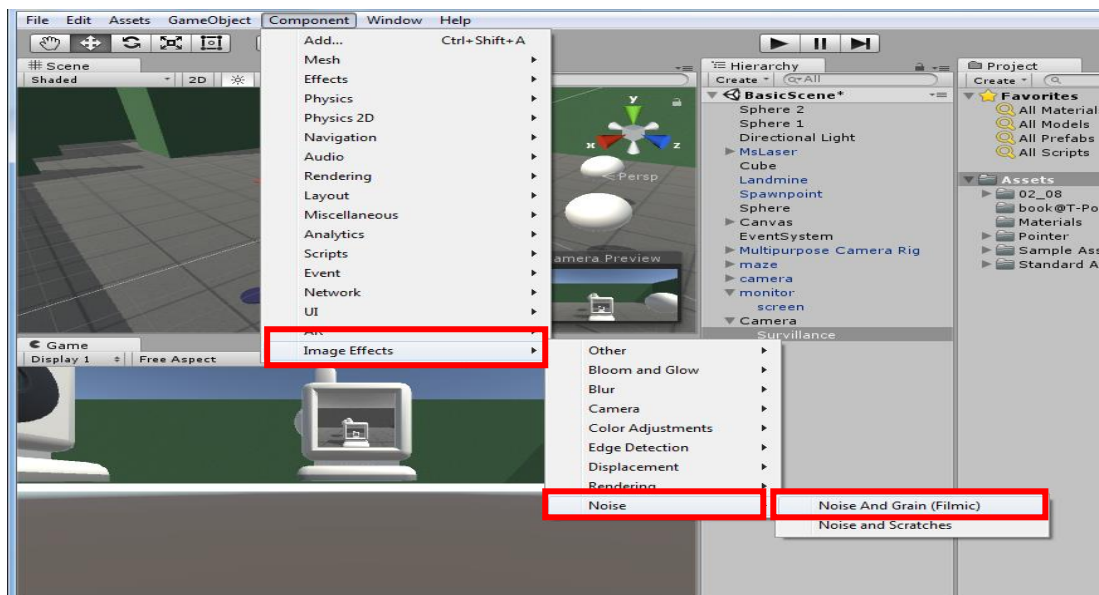
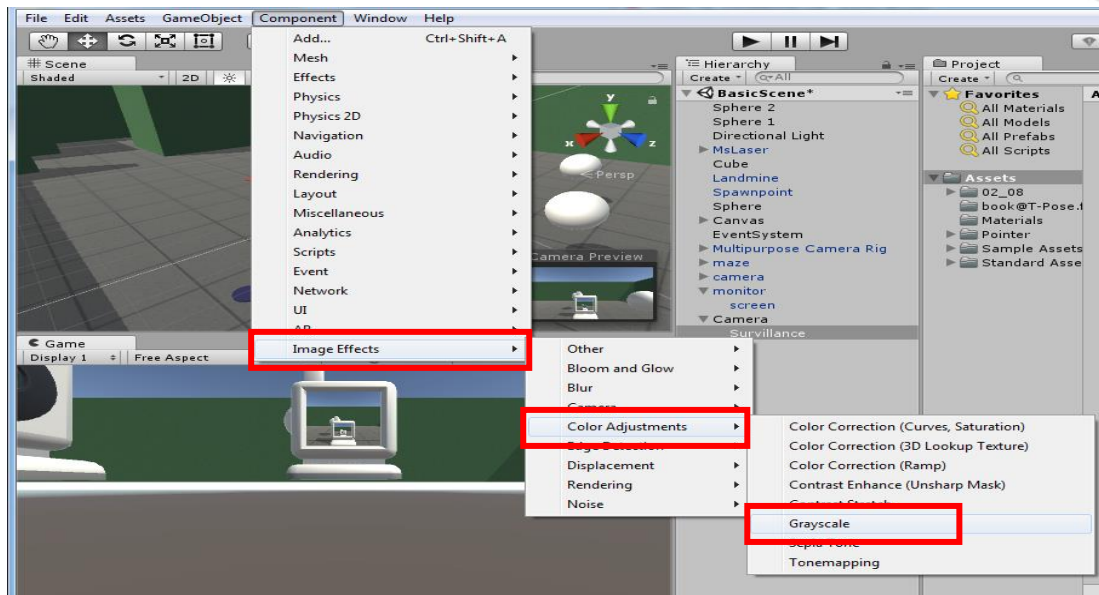
7. Dari tampilan **Hierarchy**, memperluas objek dengan memilih turunan dari **Monitor**. Kemudian dari **Inspector**, menemukan material bernama **Desert** dan menu drop-down **Shader**, mengubah ITTO **Unlit/Texture**. Terakhir, mengatur **screen** texture dasar, seperti yang ditunjukkan pada gambar berikut:



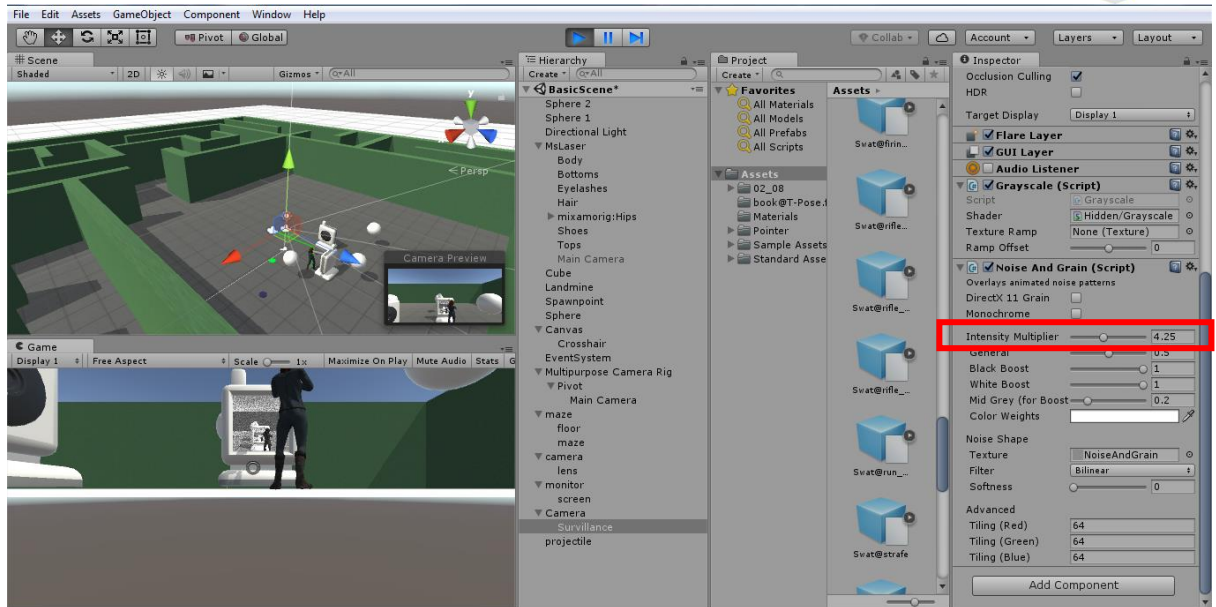
8. Selanjutnya menambahkan beberapa post-processing untuk tekstur. Dari menu utama, mengimpor **Effects Package** (Assets → Paket Impor → Effects)



9. Dari tampilan **Hierarchy**, pilih **Camera Surveillance** kemudian dari menu utama tambahkan **Grayscale** komponen efek gambar (**Component → Image Effects → Color Adjustments → Grayscale**). Tambahkan **Noise and Grain** efek gambar (**Component → Image Effects → Noise → Noise and Grain (Filmic)**). Terakhir tampilan Inspector diatur **Intensity Multiplier** dari **Noise and Grain** ke 4.



10. Jalankan projek anda dan ganti pada tampilan **Inspector** → **Noise and Grain** → **Intensity Multiplier** rubah dengan **4**, juga dapatkan melihat tindakan Anda secara real time pada layar monitor, seperti yang ditunjukkan di sini:



H. TUGAS PRAKTIKUM

1. Implementasikan **PROJEK F** diatas dengan pergerakan Camera 3D bergenre **Camera FPS** dan **Camera RPG**.

--- SELAMAT BELAJAR ---