

## MODUL 10 – Lanjutan Lighting dan Effect

---

### A. TUJUAN

- Mahasiswa dapat menambahkan Reflection Map secara Custom pada Scene
- Mahasiswa dapat Membuat Sinar Laser Pointer menggunakan Projector dan Line Renderer
- Mahasiswa dapat membuat Lighting pada scene sederhana menggunakan Lightmaps dan LightProbes
- Mahasiswa dapat membuat Spot Light Cookies di Unity

### B. PETUNJUK

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

### C. ALOKASI WAKTU: 6 jam pelajaran

### D. DASAR TEORI

Modul 10 adalah lanjutan dari Lighting dan Effect yang telah dibahas pada modul sebelumnya.

### E. LATIHAN PRAKTIKUM

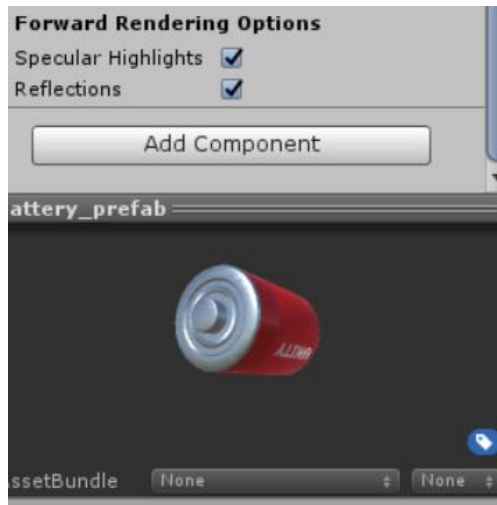
#### 1. Menambahkan Reflection Map secara Custom pada Scene

Untuk melakukan percobaan ini, anda akan membutuhkan file yang sudah disediakan pada folder 1362\_06\_02.

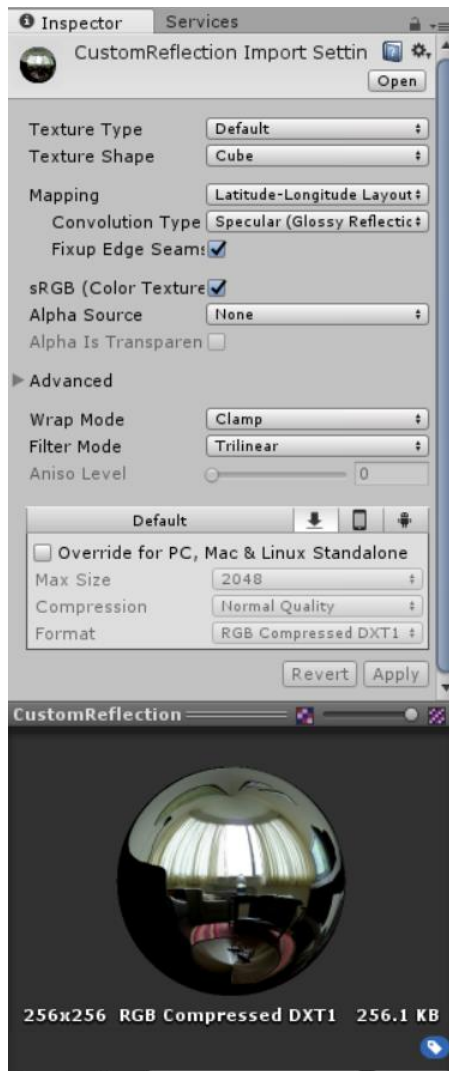
Berikut adalah beberapa langkah untuk mempersiapkan environment game:

1. Import batteryPrefab.unitypackage kedalam project baru. Pilih object battery\_prefab dari folder Asset pada Project View

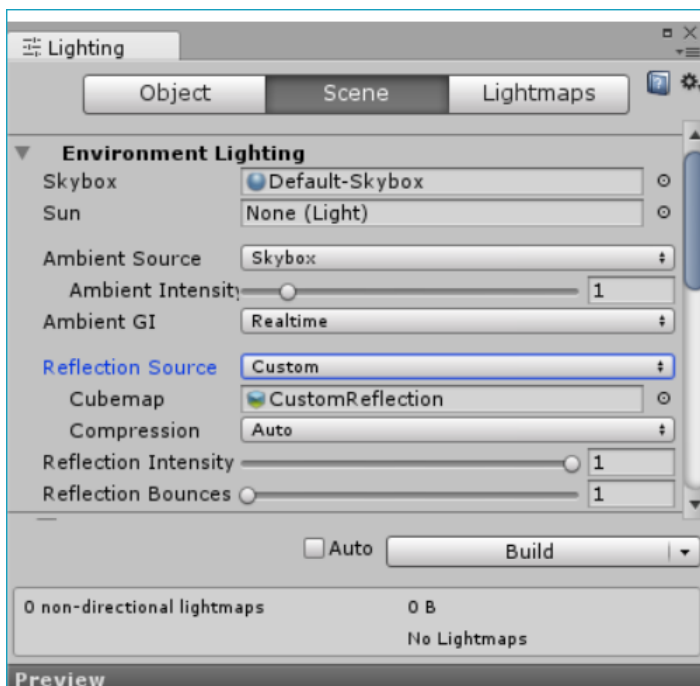
2. Perhatikan pada Inspector, expand komponen Material. Karena Map Specular, material telah dapat menampilkan kesan reflective.



3. Import image CustomReflection.jpg. Dari Inspector View, pastikan texture typenya ke Default, Texture Shape = Cube, Mappingnya ke Latitude – Longitude Layout (Cylindrical), Convolution Type = Specular (Glossy Reflection), centang Fixup Edge Seams. Kemudian ubah Filter Modenya ke Trilinear dan klik tombol Apply.



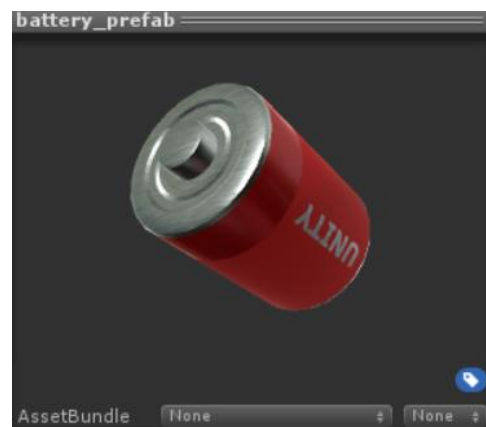
4. Ubah Reflection Skybox dengan reflection map yang baru saja anda buat. Buka jendela Lighting dari Window | Lighting. Pilih Section Scene dan ubah Reflection Source menjadi Custom. Selanjutnya Drag atau pilih CustomReflection yang anda buat kedalam CubeMap.



Perhatikan perbedaan lighting saat menggunakan skybox dan menggunakan reflection map yang baru anda buat.



Skybox



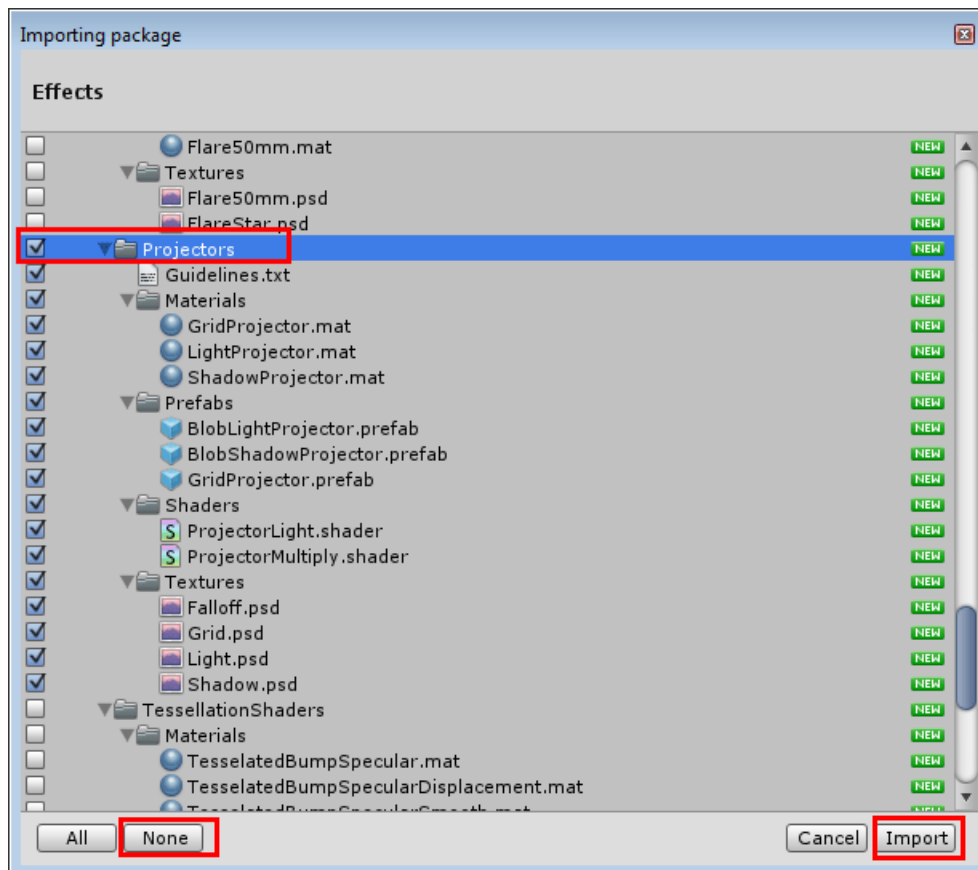
Custom Reflection map

## 2. Membuat Laser pointer menggunakan Projector dan Line Renderer

Umumnya pada game FPS (First person Shooter), cross-hair digunakan sebagai penunjuk arah tembakan, menggantinya dengan menggunakan laser juga dimungkinkan untuk tujuan khusus lain.

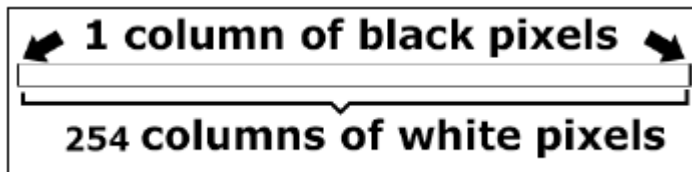
Untuk melakukan project ini, telah disediakan package berisi scene sederhana berisi character yang sedang memegang laser pointer, dan juga texture map yang dinamakan LineTexture. Seluruh file tersedia pada folder 1362\_06\_03. Anda juga akan menggunakan package Asset Effects yang disediakan oleh Unity (dan sudah diinstall saat awal menginstall Unity)

1. Import BasicScene.unitypackage kedalam project baru. Buka scene yang bernama BasicScene. Ini adalah scene dasar berisi character yang targetnya dikendalikan menggunakan mouse.
2. Import package Effects dengan menu navigasi Assets | Import Package | Effects. Jika ingin mengimport yang dibutuhkan untuk project ini saja dan tidak semua effect, hilangkan semua centang pada jendela Importing packages dengan meng-klik tombol None, kemudian centang hanya folder Projectors saja. Kemudian klik Import :

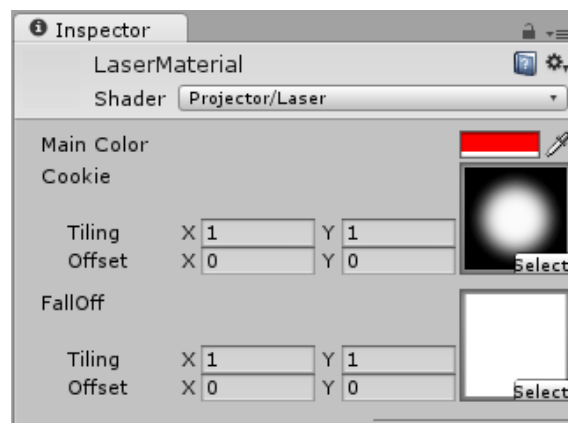


3. Dari View Inspector, cari shader ProjectorLight (didalam folder Assets | Standard Assets | Effects | Projectors | Shaders). Duplicate file shader tersebut dan ubah namanya menjadi ProjectorLaser.
4. Buka code dari ProjectorLaser, pada baris pertama, ubah Shader "Projector/Light" menjadi Shader "Projector/Laser". Kemudian cari baris kode berisi – Blend DstColor One dan ubah menjadi Blend One One, Simpan dan tutup filenya.  
(Meng-edit shader menjadi Laser dilakukan untuk mengubah tipe Blend menjadi Additive.)

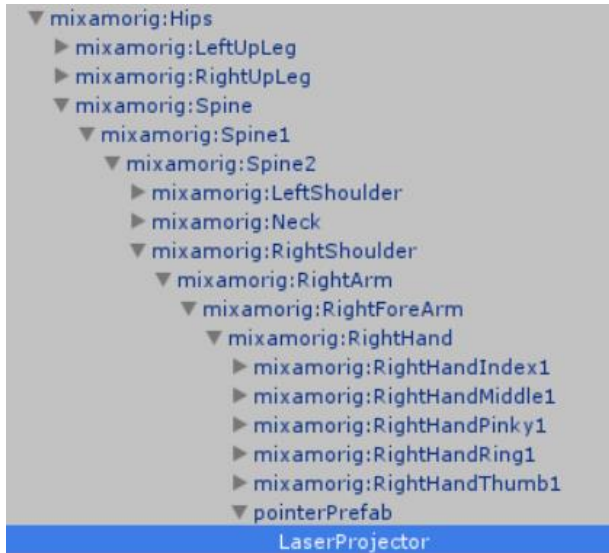
5. Selanjutnya untuk memperbaiki shader, dibutuhkan material. Dari Project View, gunakan menu dropdown Create untuk membuat Material baru. Berikan nama LaserMaterial. Pilih LaserMaterial dari Project view, dari Inspector view, ubah shadernya menjadi Projector/Laser.
6. Dari Project view, cari texture Falloff, buka ke dalam image editor, pixel kolom pertama dan kolom terakhir buat sebagai warna hitam, kolom yang lain dibuat sebagai warna putih. Simpan filenya dan kembali lagi ke Unity.



7. Ubah Main Color dari LaserMaterial menjadi red (RGB: 255,0,0). Kemudian dari slot texture, pilih texture Light sebagai menjadi Cookie dan texture Falloff menjadi Falloff.



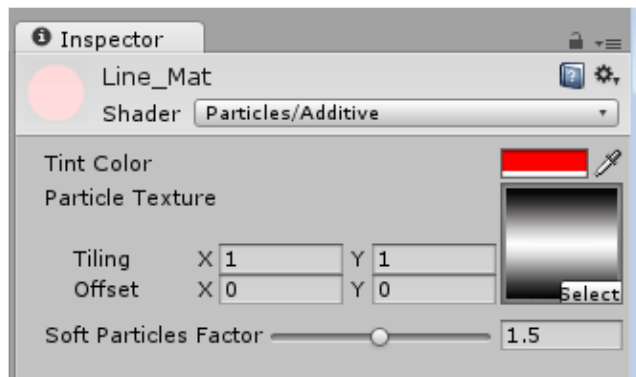
8. Dari Hierarchy view, Cari dan pilih object pointerPrefab (MsLaser | mixamorig:Hips | mixamorig:Spine | mixamorig:Spine1 | mixamorig:Spine2 | mixamorig:RightShoulder | mixamorig:RightArm | mixamorig:RightForeArm | mixamorig:RightHand | pointerPrefab). Kemudian, dari menu drop-down Create, pilih Create Empty Child. Ubah nama new child dari pointerPrefab sebagai LaserProjector.



9. Select object ProjectorLaser. Kemudian dari Inspector View, klik Add Component, kemudian pilih Effect | Projector. Kemudian dari komponen Projector, set option Orthographic sebagai True dan set Orthographic Size menjadi 0.1. Kemudian pilih LaserMaterial dari slot Material.
10. Coba jalankan scene, jika sudah benar maka akan terlihat laser yang berwarna merah menyasar ke target.



11. Selanjutnya akan dibuat material untuk komponen Line Renderer. Dari Project View, gunakan menu drop-down Create untuk menambahkan Material baru. Beri nama Line-Mat.
12. Dari menu Inspector, ubah shader dari Line\_Mat menjadi Particles/Additive. Set Tint Colornya menjadi red (RGB: 255,0,0)
13. Import file image LineTexture. Set sebagai Particle Texture dari Line\_Mat. Set Particles Factor menjadi 1.5.



14. Gunakan menu drop-down Create dari Project view untuk menambahkan script C# yang dinamakan LaserAim. Buka di script editor. Tambahkan script berikut:

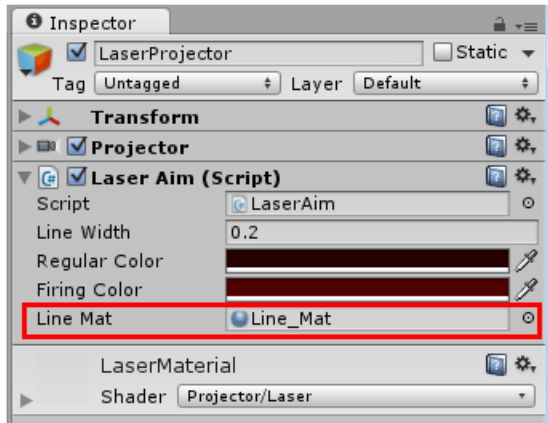


```

using UnityEngine;
using System.Collections;
public class LaserAim : MonoBehaviour
{
    public float lineWidth = 0.2f;
    public Color regularColor = new Color(0.15f, 0, 0, 1);
    public Color firingColor = new Color(0.31f, 0, 0, 1);
    public Material lineMat;
    private Vector3 lineEnd;
    private Projector proj;
    private LineRenderer line;
    void Start()
    {
        line = gameObject.AddComponent<LineRenderer>();
        line.material = lineMat;
        line.material.SetColor("_TintColor", regularColor);
        line.SetVertexCount(2);
        line.SetWidth(lineWidth, lineWidth);
        proj = GetComponent<Projector>();
    }
    void Update()
    {
        RaycastHit hit;
        Vector3 fwd = transform.TransformDirection(Vector3.forward);
        if (Physics.Raycast(transform.position, fwd, out hit))
        {
            lineEnd = hit.point;
            float margin = 0.5f;
            proj.farClipPlane = hit.distance + margin;
        }
        else
        {
            lineEnd = transform.position + fwd * 10f;
        }
        line.SetPosition(0, transform.position);
        line.SetPosition(1, lineEnd);
        if (Input.GetButton("Fire1"))
        {
            float lerpSpeed = Mathf.Sin(Time.time * 10f);
            lerpSpeed = Mathf.Abs(lerpSpeed);
            Color lerpColor = Color.Lerp(regularColor,
            firingColor, lerpSpeed);
            line.material.SetColor("_TintColor", lerpColor);
        }
        if (Input.GetButtonUp("Fire1"))
        {
            line.material.SetColor("_TintColor", regularColor);
        }
    }
}

```

15. Atach script yang baru dibuat ke object LaserProjector.
16. Pilih GameObject LaserProjector, dari Inspector view, pada komponen Laser Aim isikan Line\_Mat pada slot Line Material. Jalankan scene, tekan Ctrl untuk menembak target dan perhatikan perbedaan laser beam saat menembak dan saat tidak menembak.



17. Untuk mengubah ketebalan dari beam (line), dapat diubah dari Line Width pada komponen Laser Aim menjadi lebih kecil, dan Orthographic size dari komponen Projector menjadi lebih kecil.

Efek dari Laser Pointer bias dibuat dengan kombinasi dua efek yang berbeda, Projector dan Line Renderer.

Projector yang digunakan untuk simulasi Light, shadows, dsb adalah komponen yang memproyeksikan material (dan teksturnya) ke game Object lain. Dengan menerapkan projector pada object LaserProjector, akan dijamin menunjuk ke arah yang benar setiap waktu. Untuk mendapatkan kesan tampilan vibrant (getaran), material shader dari projector telah diubah untuk membuat kesan cerah. Script yang dipakai, berisi kode yang mencegah proyeksi menembus object, dengan cara menset Far Clip Plane kesatu level yang sama dari object pertama yang menerima proyeksi. Code yang melakukannya adalah `proj.farClipPlane = hit.distance + margin;`.

Line Renderer dibuat dinamis lewat script. Script yang dibuat juga berguna untuk mensetting tampilan line, update posisi line, dan mengubah tampilan saat tombol menembak ditekan. Penjelasan lebih lengkap tentang script dapat dilihat pada folder 1362\_06\_03 | End.

### 3. Lighting pada scene sederhana menggunakan Lightmaps dan Light Probes

Lightmaps adalah pilihan yang baik untuk membuat Lighting realtime, karena memiliki kelebihan untuk menyediakan tampilan lighting yang realistic tanpa pengaruh berarti dari spesifikasi prosesor. Kelemahannya hanya tidak dapat diterapkan pada Object dinamis, lighting dari element yang penting (seperti lighting untuk character) dapat terlihat tidak realistic, sehingga dapat dipastikan tidak cocok dengan intensitas cahaya lingkungannya. Solusinya adalah dengan menggunakan Light Probes.

Light Probes bekerja dengan melakukan sampling intensitas cahaya pada lokasi dimana cahaya menyorot. Object dinamis, saat dikenai Light Probes, akan menyala menurut interpolasi dari probes terdekat yang ada disekitarnya.

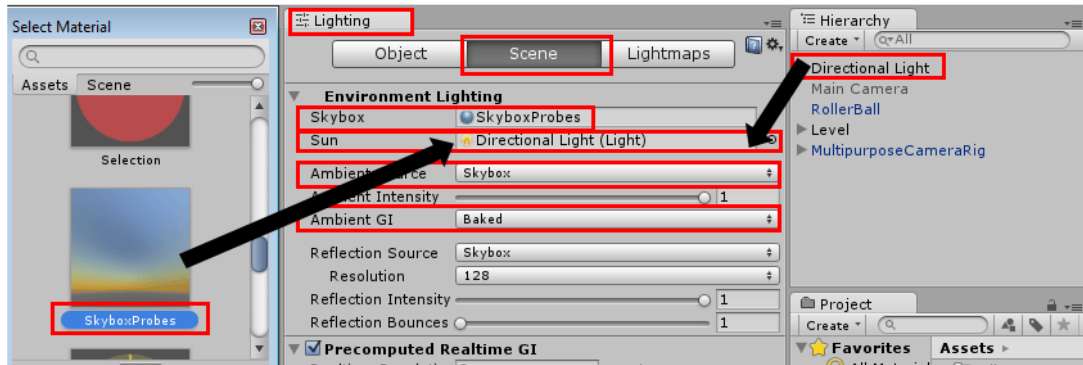
Untuk melakukan project ini, telah disiapkan basic scene berisi environment game sederhana dan instance dari sample asset Unity Rollerball, yang digunakan sebagai character player. Geometry dari scene dibuat menggunakan ProBuilder 2.0, extension yang dibuat oleh ProCore dan dijual di Unity Asset Store dan website ProCore (<http://www.protocolsforunity3d.com>). Probuilder adalah tool design level yang sangat baik yang dapat mempercepat proses desain level baik level sederhana maupun level yang kompleks.

Package LightProbes.unitypackage, berisi scene dan semua file yang dibutuhkan, dapat dibuka pada folder 1362\_06\_06. Untuk membuat refleksi pada object disekitarnya menggunakan Reflection Probes, ikuti langkah berikut:

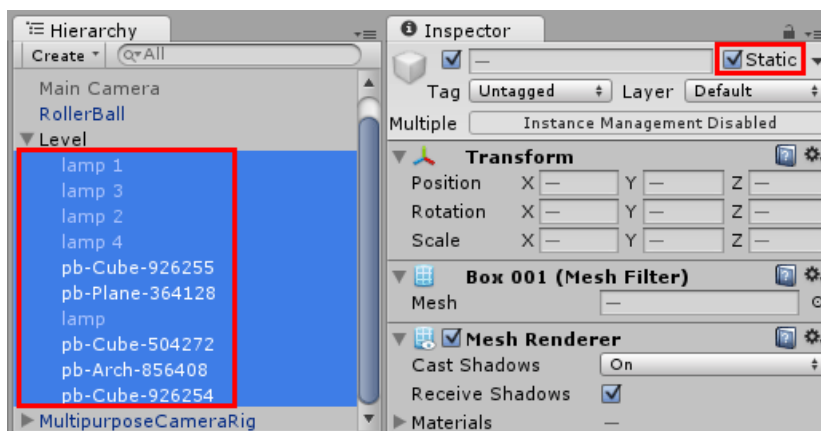
1. Import LightProbes.unitypackage kedalam project baru. Buka scene yang bernama LightProbes. Scene tersebut berisi basic environment dan game Rollerball sederhana.
2. Untuk tahap awal, dimulai dengan set up lighting. Dari hierarchy view, pilih Directional Light. Kemudian, dari Inspector view, set Baking menjadi Baked. Masih di Inspector view, centang static pada bagian kanan atas view, seperti terlihat pada gambar berikut.



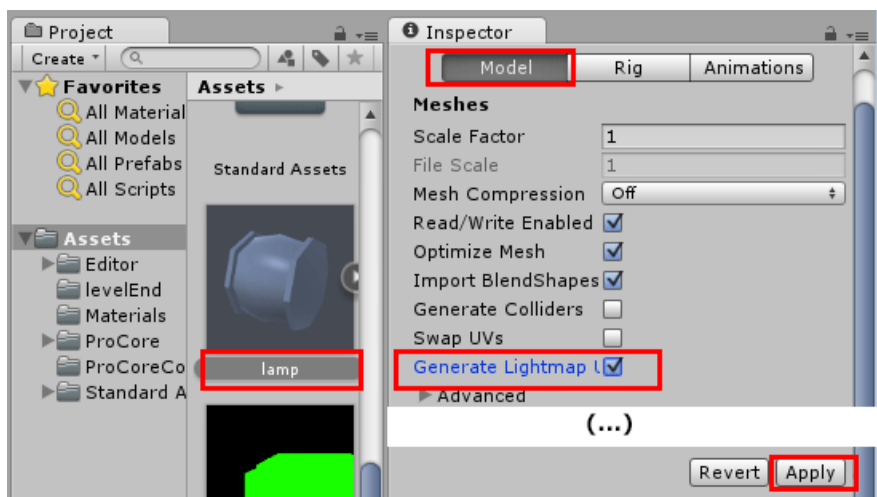
3. Langkah berikutnya dengan men-set Global Illumination untuk scene. Buka jendela Lighting (melalui Window | Lighting) dan pilih section scene. Kemudian dari subsection Environment Lighting, set SkyboxProbes (ada didalam Assets) di skybox, dan Directional Light di Sun. Selanjutnya, ubah opsi Ambient GI dari realtime menjadi Baked, seperti terlihat pada gambar berikut:



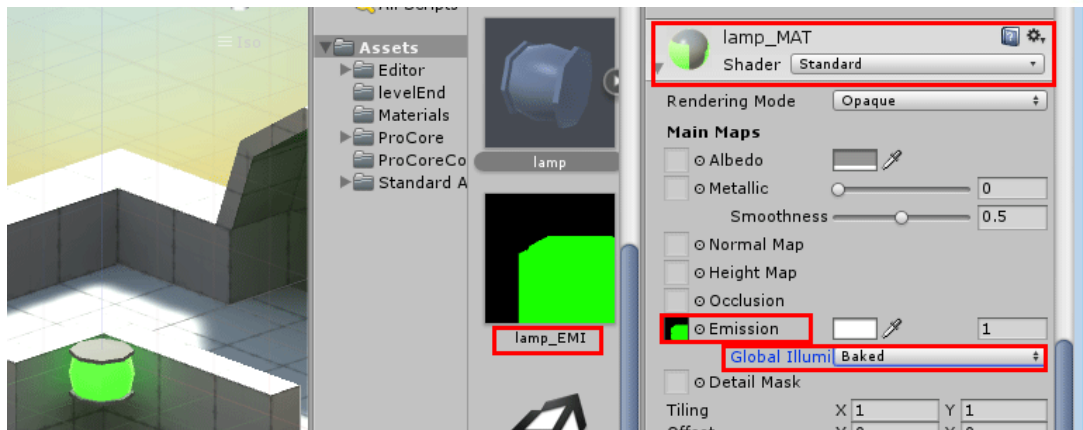
4. Lightmaps hanya dapat digunakan pada object statis saja. Dari Hierarchy view, expand game object Level untuk menampilkan daftar object-object childrenya. Kemudian pilih semua children dan set sebagai static pada inspector view kanan atas.



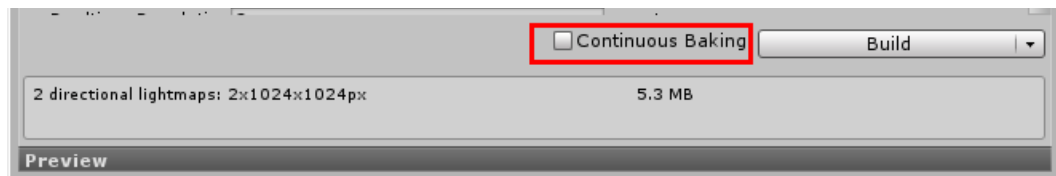
5. 3D Mesh yang diimport harus memiliki fitur Lightmap UV Coordinates. Dari Project view, cari dan pilih mesh lamp. Kemudian dari Inspector view, didalam seksi Model dari Import Setting, periksa opsi Generate Lightmaps UV, klik tombol Apply untuk menyimpan perubahan.



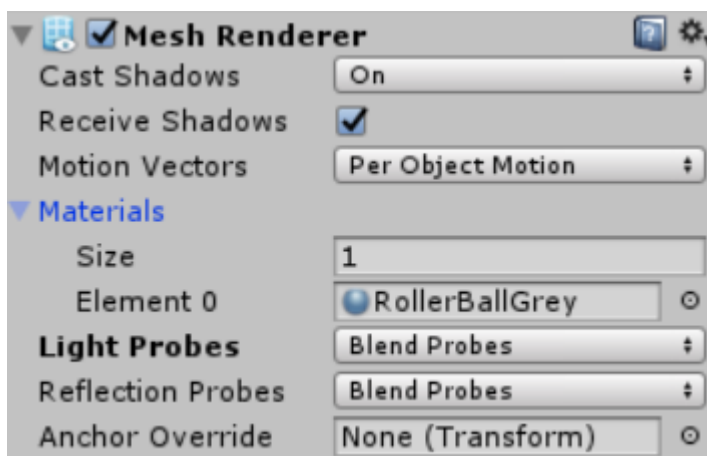
6. Pada bagian bawah (Shader), Ubah Emission menjadi lamp\_EMI, dan Global Illuminationnya menjadi dari realtime menjadi Baked. Perhatikan perubahan yang terjadi dari lamp pada scene.



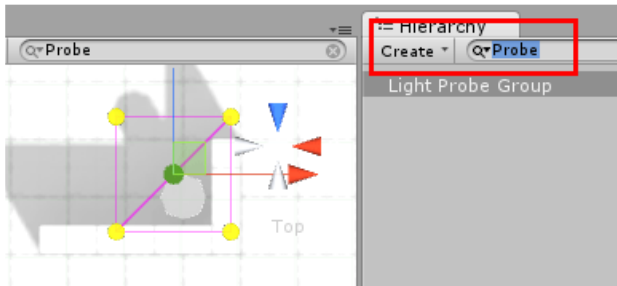
7. Buka Jendela Lighting. Secara default, Opsi Continuous Baking selalu dicentang, hilangkan centangnya agar Lightmaps dapat di-bake sesuai kebutuhan. Klik Build dan tunggu Lightmaps sampai selesai di generate.



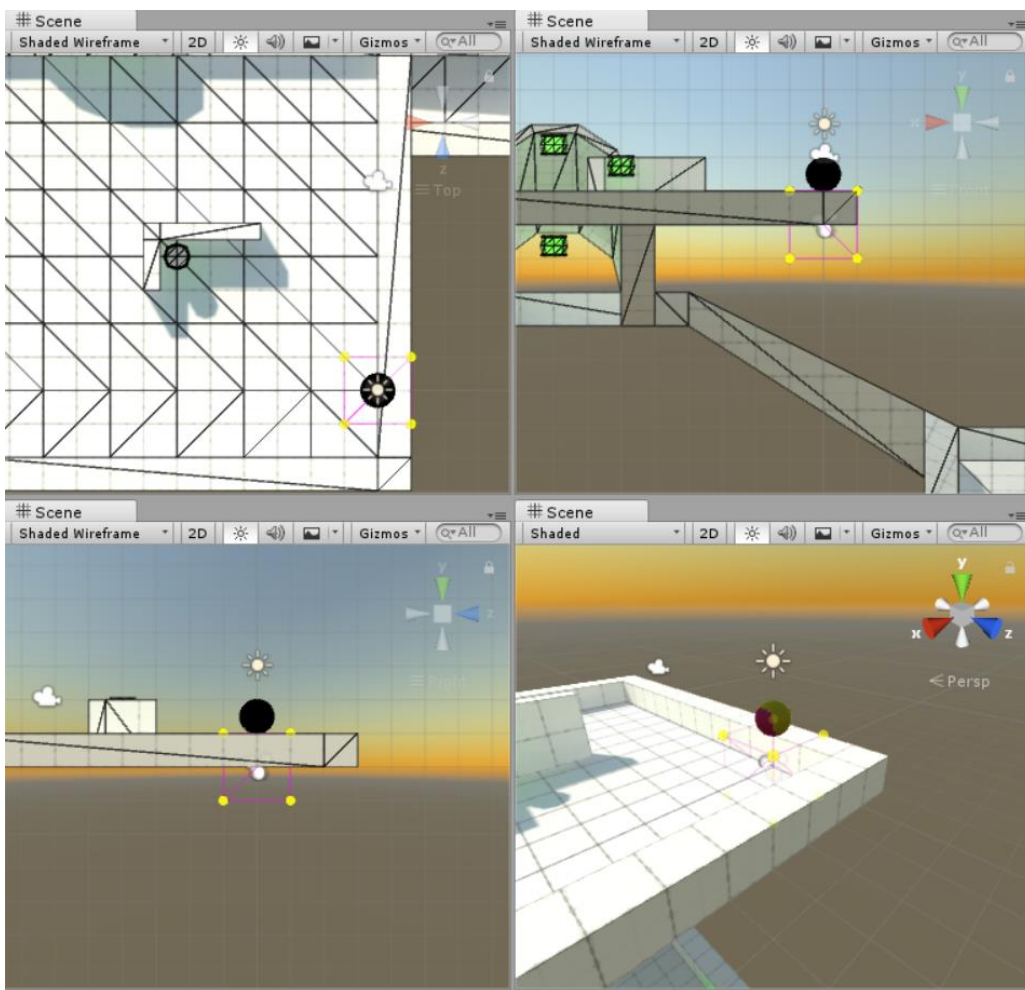
8. Dari Hierarchy view, pilih RollerBall. Kemudian dari Inspector view, cari komponen Mesh Renderer, Pada Light Probes, pastikan Blend Probes sudah terpilih:



9. Selanjutnya akan dibuat Light Probes untuk scene. Dari Hierarchy view, klik menu Dropdown Create dan tambahkan Light Probe Group kedalam scene (Create | Light | Light Probe Group)
10. Untuk memanipulasi probes saja, ketikkan Probe kedalam field pencarian dari hierarchy view. Hal ini akan membuat Light Probe Group sebagai satu-satunya game object yang tampak dan dapat diedit di scene.

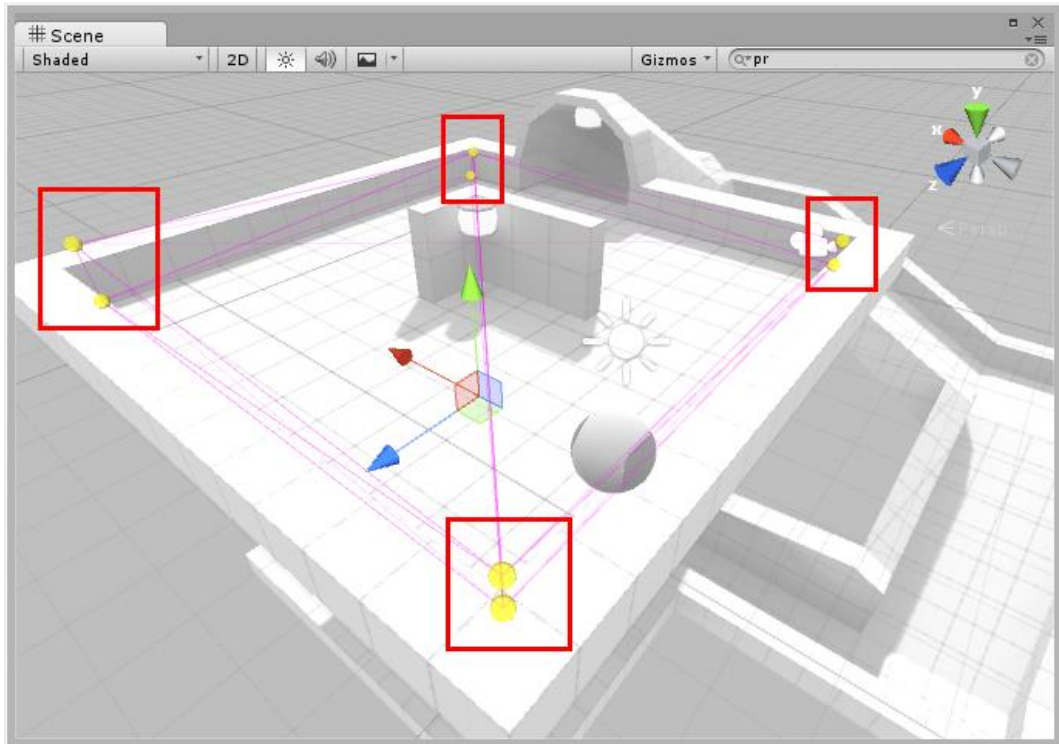


11. Ubah Layout viewport menjadi 4 bagian dengan menggunakan menu Window | Layouts | 4 Split. Set viewport sebagai Top, Front, Right, dan Persp. Pada bagian Top, Front, dan Right tampilannya diubah ke Wireframe. Hal ini dilakukan untuk mempermudah kita menempatkan Light Probes.

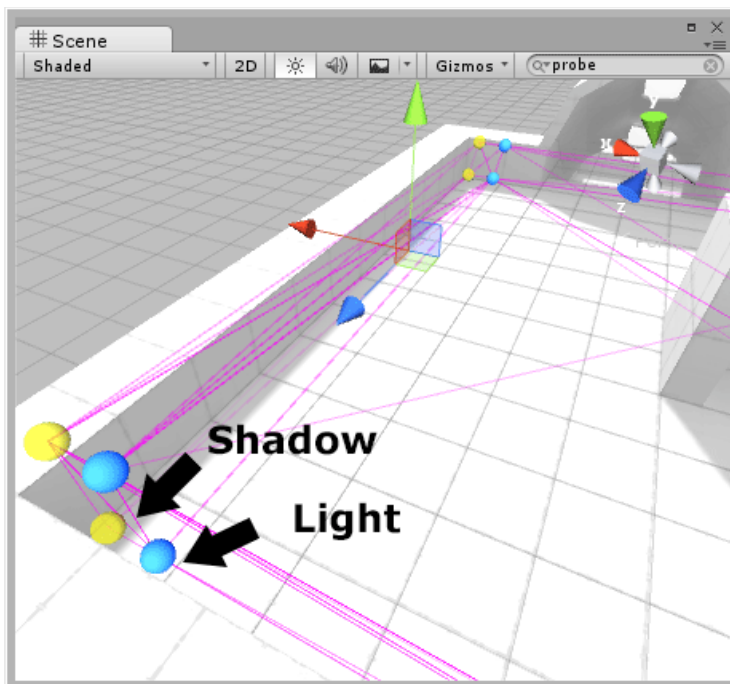


12. Tempatkan Light Probes yang sudah dibuat ke tiap sudut dari level top room. Dengan drag tiap Probe nya.

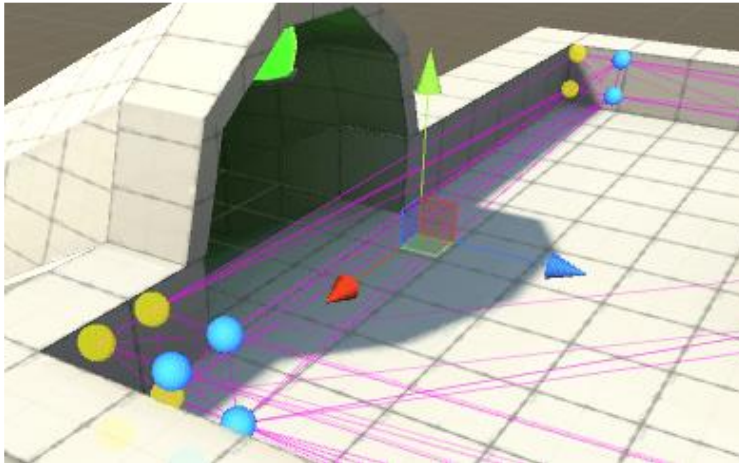




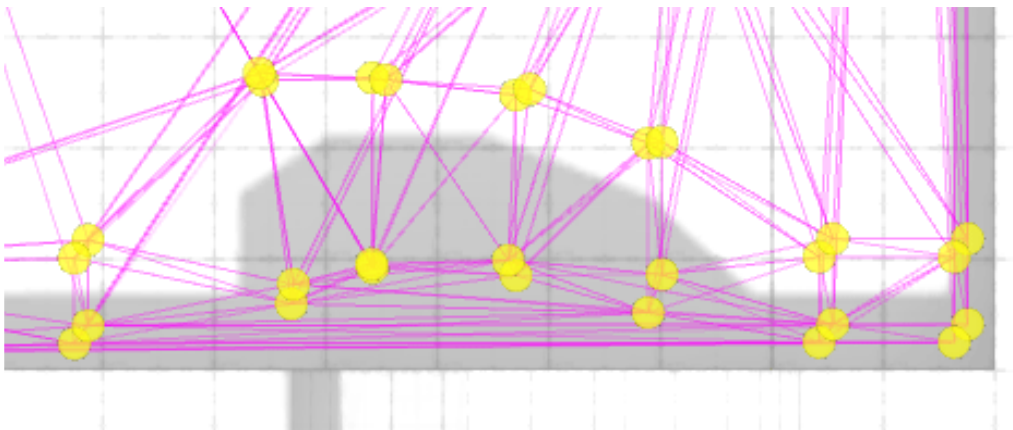
13. Pilih empat probes pada sisi kiri dari tunnel / terowongan. Duplikat keempat probe tersebut Dengan Ctrl + D, lalu geser probenya sedikit kesebelah kanan dari probe asli, setidaknya hingga tidak terkena shadow dari wall / dinding.



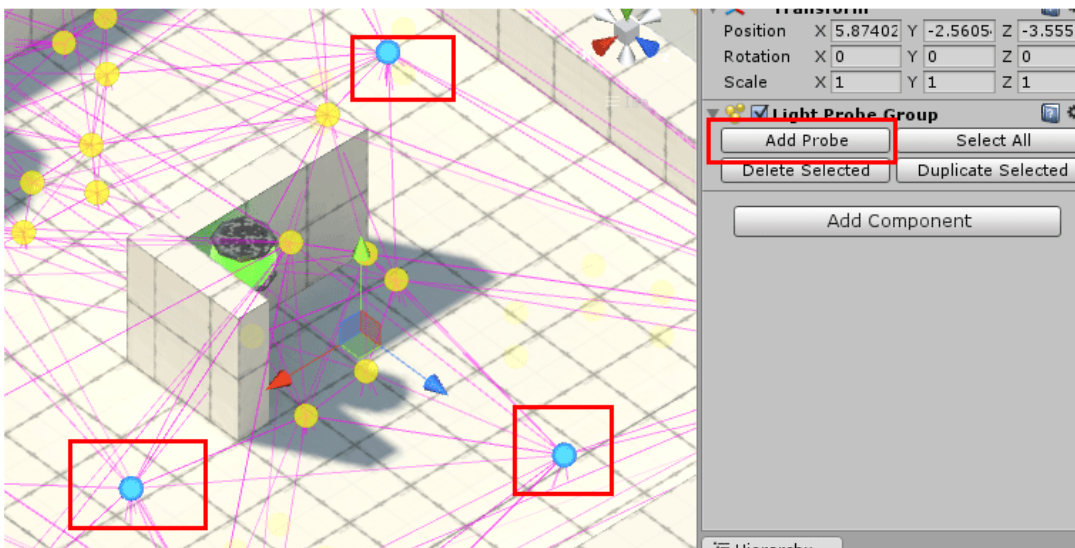
14. Ulangi langkah diatas, kali ini pada probe yang berada di tepi tunnel / terowongan. Jika ingin menghapus probe, dapat digunakan Ctrl + Backspace.



15. Duplikat keempat probes yang terdekat dengan tunnel dan posisikan sehingga seperti gambar berikut.

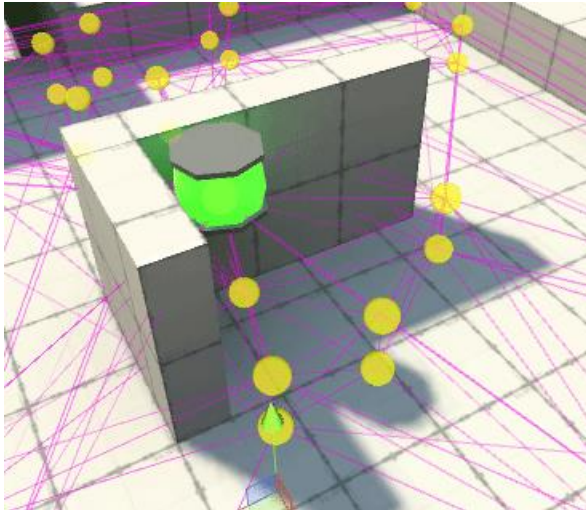


16. Gunakan tombol Add Probe untuk menambah tiga probe lagi pada sekeliling dinding tengah.

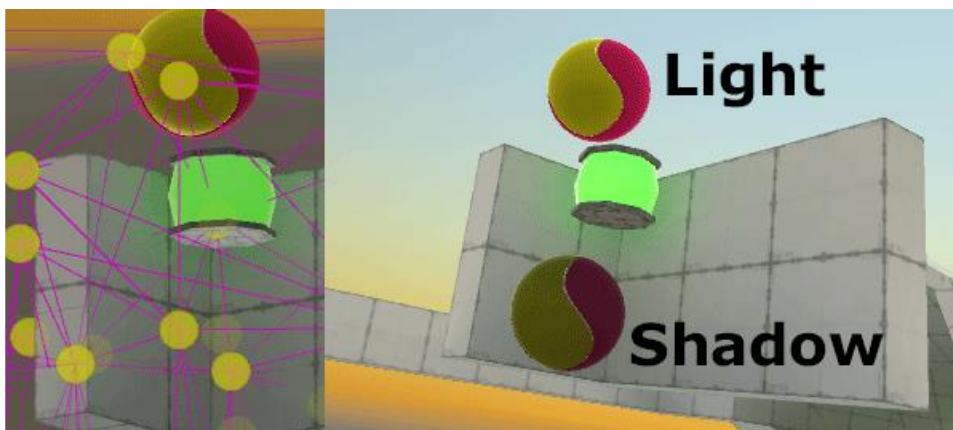


17. Berikutnya, tambahkan Light Probes kedalam shadow dari dinding L-Shaped.

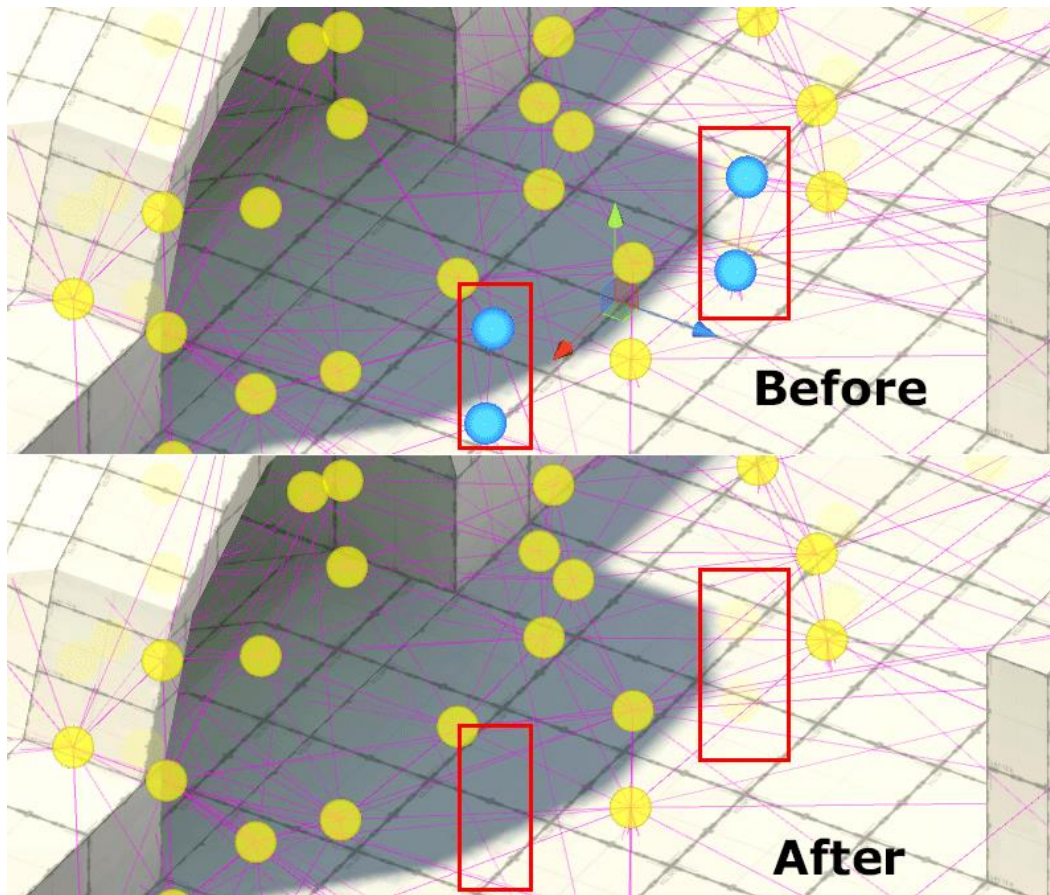




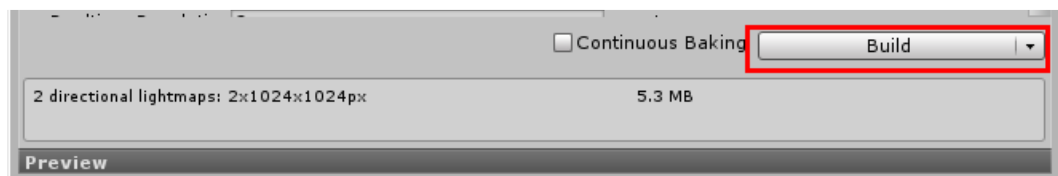
18. Karena Rollerball dapat meloncat, tempatkan probes ditempat yang lebih tinggi daripada loncatan bola.



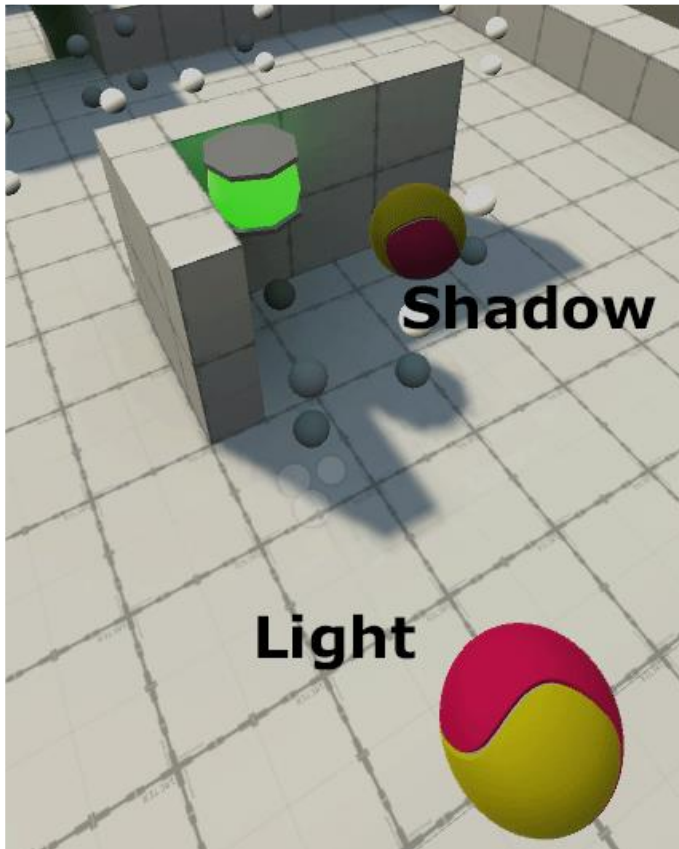
19. Menempatkan banyak Light Probes pada scene dapat memakan resource memory yang besar. Lakukan optimasi Light Probes dengan menghapus probe probe dimana player tidak mungkin untuk melihatnya. Hapus juga probes yang terlalu berdekatan satu sama lain pada satu kondisi lighting yang sama.



20. Pada bagian bawah dari jendela Lighting, klik tombol Build dan tunggu hingga baking Lightmaps selesai dilakukan.



21. Test scenenya. Rollerball akan dikenai cahaya sesuai dengan Light Probes yang telah dibuat.



22. Tetap tambahkan probes hingga seluruh level terisi probes.

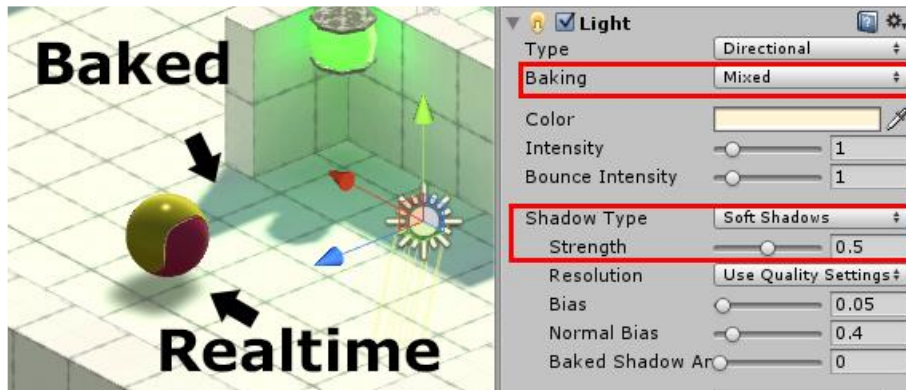
Lightmaps pada dasarnya adalah map tekstur yang memiliki fitur lighting/shadowing scene, global illumination, indirect illumination, dan fitur object material Emissive. Dapat degenerate secara otomatis ataupun sesuai kebutuhan oleh Lighting dari Unity Engine. Tentu saja beberapa hal harus diperhatikan, seperti:

1. Menset semua object yang tidak bergerak dan Light dengan mem-bake nya sebagai Statis.
2. Menset lighting game sebagai Baked.
3. Menset Ambient GI dari scene sebagai Baked.
4. Menset opsi Global Illumination dari material emissive sebagai Baked
5. Untuk seluruh mesh 3D (khususnya yg diimport) dilakukan Generate Light UV
6. Build Lightmaps secara manual dari jendela Lighting, atau centang opsi Continues Baking

Light Probes bekerja dengan mensampling iluminasi scene pada titik dimana mereka ditempatkan.

Jika ingin lebih mengefisienkan komputasi processor, kita dapat mengubah penggunaan murni Light Probes dengan lighting Mixed. Hapus Light Probe Group dari scene, Pilih Directional Light, dari komponen Light, ubah Baking menjadi Mixed. Kemudian Set Shadow Type menjadi Soft Shadows dan

Strength menjadi 0.5. Terakhir klik tombol Build dan tunggu hingga Lightmaps selesai di Baked. Lighting dan Shadowing realtime akan dikenakan dari dan ke objek-objek dinamis, seperti Rollerball.



#### A. TUGAS PRAKTIKUM

- a. Kerjakan seluruh project yang ada dimodul ini dan kumpulkan. Khusus untuk project Light Probes, tambahkan Light Probe ke semua lantai dan bukan hanya di lantai atas saja.

--- SELAMAT BELAJAR ---