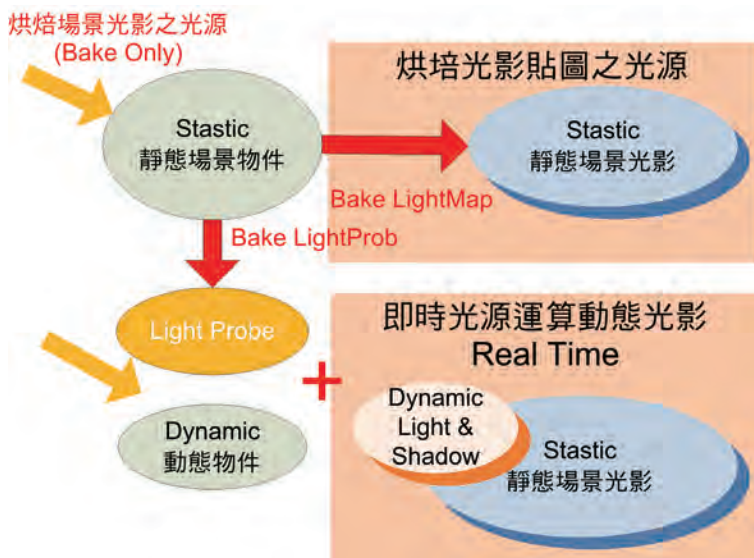


5.2.3 光偵測子

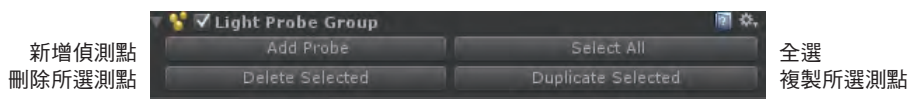
Unity 的光影設定概念可分為烘焙用光源及動態光源，分別運算場景物件的光影，其相互關係可參考如下圖：



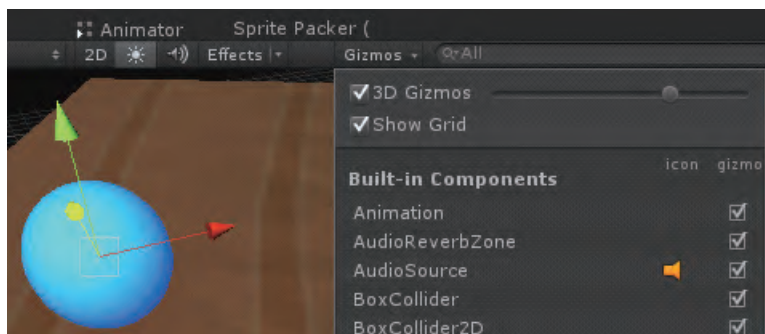
烘焙用光源可預先運算出靜態場景的光影貼圖，而即時運算時僅計算動態光源對動態物件的光影，此時烘焙用光源已關閉，動態物件無法取得烘焙時場景的光線資訊。為此 Unity 設置光偵測子 (Light Probe) 散佈在場景中，當運算靜態光影貼圖時記錄場景光影資訊，提供即時運算時動態物件所處位置之光影訊息，使物件呈現出與靜態場景光影融合的細緻畫面。

光偵測子設定步驟：

1. 建立空物件更名為 Light Probes。
2. 在功能表 Component → Rendering → 加入 Light Probe Group 元件。
3. 按下 Add Probe 在場景上建立第一個 Light Probe。



Light Probe 是記錄該空間光線資訊，場景上以藍色圓點圖示 (Gizmo) 代表，若圖示太小可調整 Gizmo 上的滑桿。



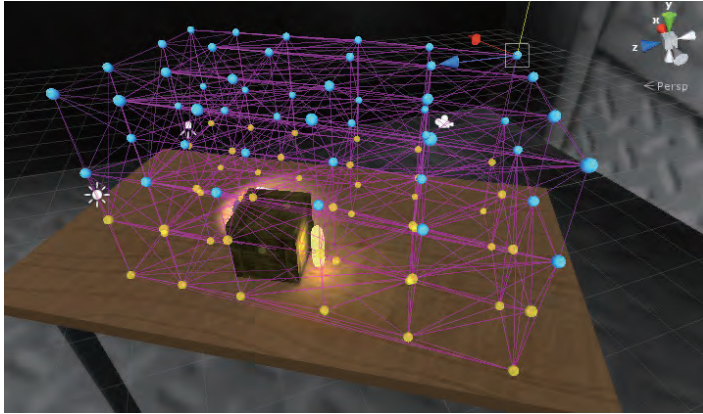
4. 點選第一個偵測點後按下 **Select All**，再以 **Duplicate Selected** 複製該點，移動第二點到鄰近適當位置。



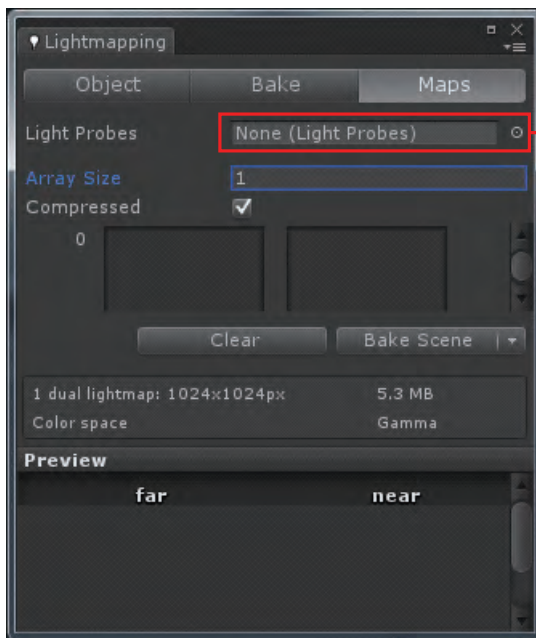
再次重複 **Select All** 與 **Duplicate Selected** 構成線狀排列。平移複製點後再次重複選點及複製的功能，將偵測點佈滿動態物體所需要的平面。



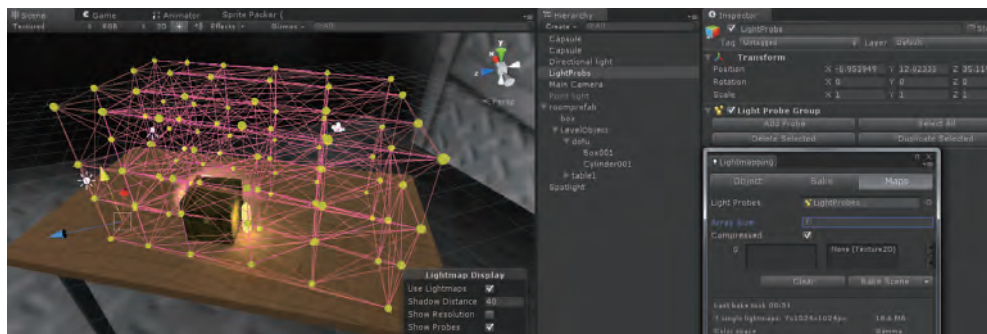
Light Probes 必須形成立體空間，因此佈滿底面後全選偵測點，再複製位移向高度方向延伸，構成 3D 立體網格，如下圖。Light probes 網狀配置可配合場景燈光變化狀態調整疏密度。燈光變化大的區域較密，無光影變化處可稀疏配置。



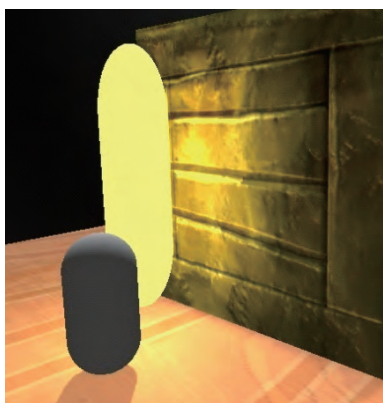
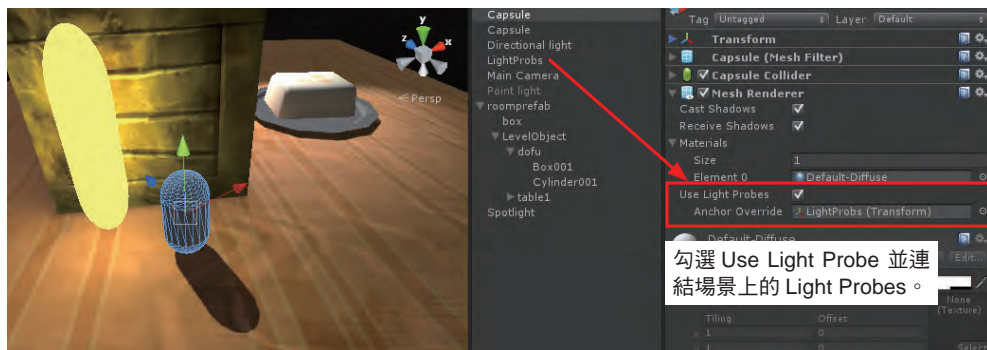
5. 重算場景光影貼圖，在 Light Probes 欄位可暫不填入，運算後 Unity 將自動連結。



Bake Scene 時，Unity 會自動將光資訊記錄到場景上的 Light Prob。



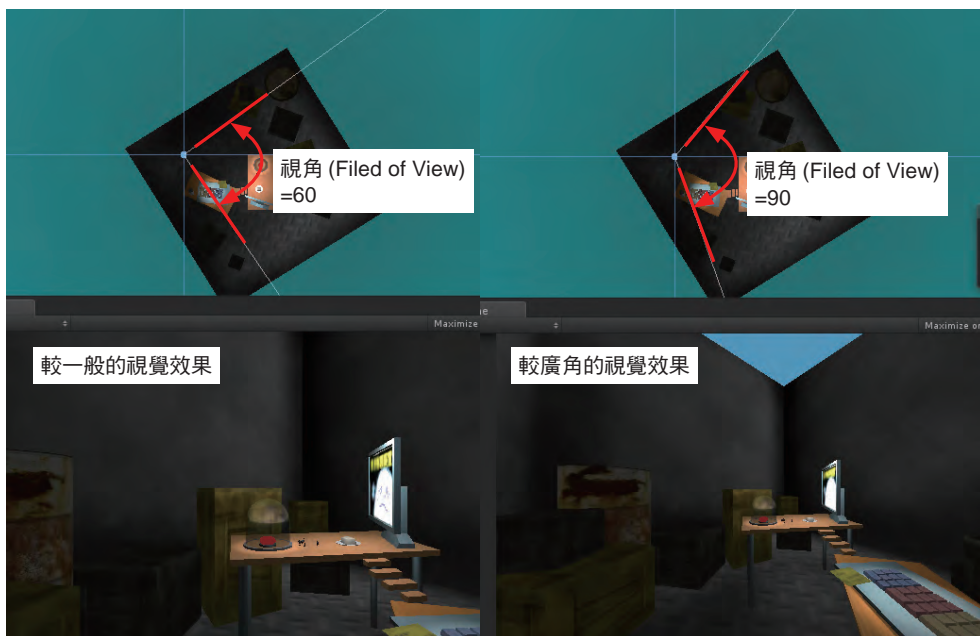
6. 設定場景上動態物件的 Mesh Renderer 使用 Light Probe 光影資訊，播放測試可觀察物件表面已將靜態烘焙的環境光渲染在表面。



5.3 相機介紹

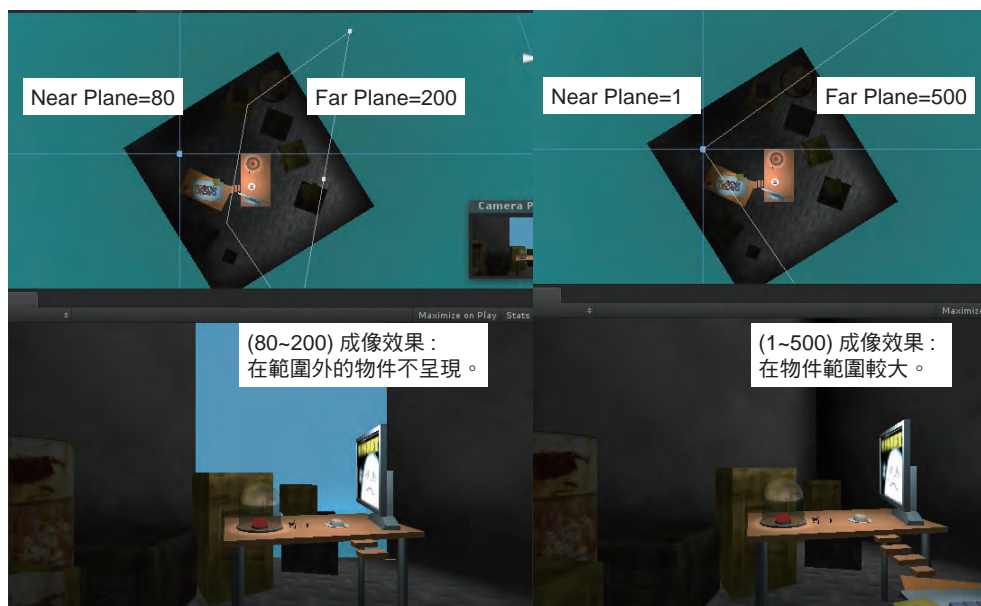
5.3.1 相機視角 (Field of View)

Field of View 是相機水平視角的參數，角度愈大愈具有超廣角效果。



5.3.2 相機可視截面

相機可視截面 (Clipping Planes)：決定相機框選物件的最近與最遠界線。範圍外的物件無法呈現，如下圖：



5.3.3 螢幕成像

螢幕成像 (Normalized View Port Rect) 是相機影像投影到螢幕成像的對應參數，如下圖所示。



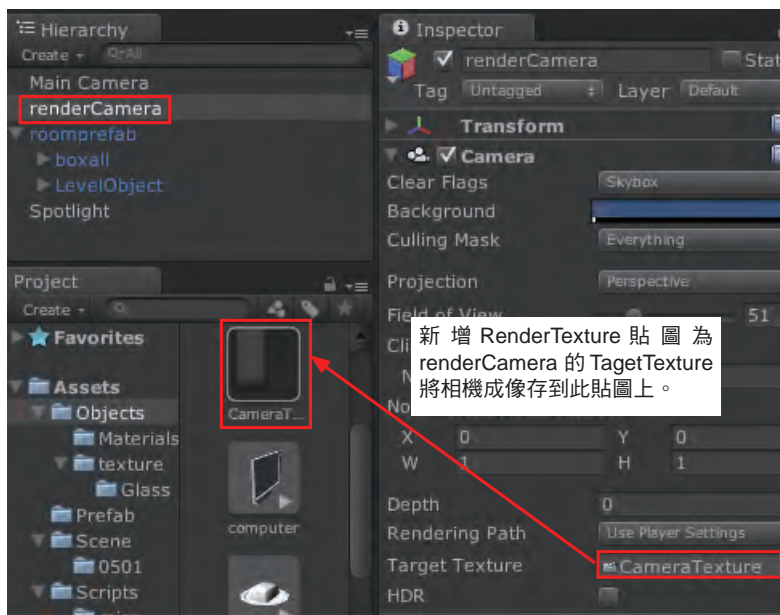
5.3.4 成像演算方式

成像演算方式 (Rendering Path) 設定是相機成像演算法，左右了畫面的質感與成像速度，目前 Unity 有幾種演算模式：

- **Deferred Lighting**：這是計算光影最真實的成像模式，但相對需要優良的硬體支援，此模式不支援行動裝置。
- **Forward Rendering**：是以 GPU 渲染程式主導的即時成像模式，可即時計算陰影，但光源數受限，僅有少數的主光源作為光線計算的參考。
- **Vertex Lit**：是最精簡的成像演算法，無陰影計算。

5.3.5 影像貼圖

影像貼圖 (Target Texture) 是設定相機影像輸出到指定的相機貼圖 (RenderTexture)。例如在場景新增一個相機 (renderCamera)，目標為鍵盤與滑鼠區。在 Project 視窗增加一張 Render Texture 更名為 CameraTexture，此張影像貼圖指定為 renderCamera 的影像輸出目標。



將 CameraTexture 的相機影像顯示在 PC 螢幕，可將 screen 物件的材質貼圖換成 CameraTexture 結果如下圖，PC 螢幕成了滑鼠區的監視畫面：



5.3.6 3DSpySeal_04 建立子母畫面

續 3DSpySeal_03room(spotlight) 遊戲場景，建立一個可以觀察電腦螢幕顯示訊息的子畫面，小畫面縮小寬高 25%，置於螢幕的右上方。

步驟：

1. 在場景新增一個相機 (nearCamera) 目標為螢幕區。調整相機參數：
2. Normalized View Port Rect 為 x:0.75 ; y:0.75; W =0.25; H=0.25 。
3. Clipping planes: Near = 1,Far =200;
4. 場景存檔為 3DSpySeal_04 Camera 。



佔全螢幕寬的 25%

W=0.25

螢幕座標 (1,1)

佔全螢幕高的 25%

H=0.25

子畫面起點 (0.75,0.75)

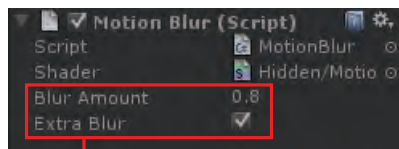
全螢幕原點 (0,0)

5.4 影像特效

影像特效是 Unity 專業版才支援的功能，透過相機成像演算所加入的特殊視覺效果。請在功能選單 Assets → Import Package → Image Effects 匯入影像特效的素材。

5.4.1 動態模糊 (Motion Blur image effect)

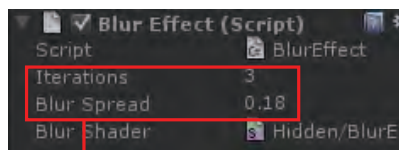
在相機物件上加入元件 Image Effect → Motion Blur，動態模糊僅在相機位移時產生模糊效果，且需要 GeForce2 或 Radeon 7000 以上的顯示卡有效果。



Blur 數值愈大運動愈模糊，或 Extra Blur 可強化模糊效果。

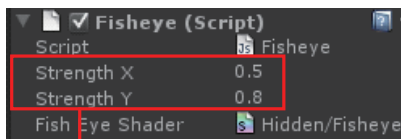
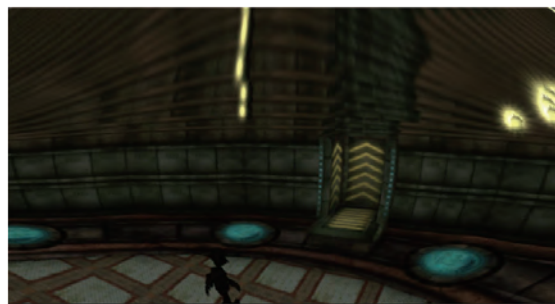
5.4.2 模糊 (Blur effect)

模糊 (Blur effect) 是相機焦距不對影像模糊的效果，與動態模糊產生效果的時機不同如下圖。模糊計算仍需要顯示卡的硬體支援。



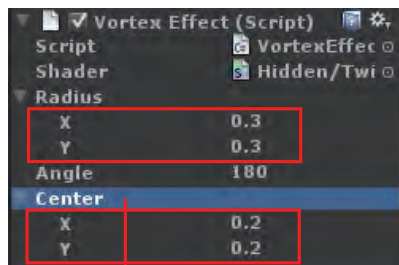
Iterations 模糊運算的迴圈迭代次數，數值太大將浪費運算時間。
Blur spread 建議 0.6~0.7。

5.4.3 魚眼效果 (Fisheye image effect)



Strength X, Strength Y
XY 軸向魚眼拉扯強度。

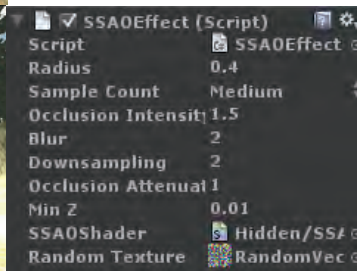
5.4.4 扭轉特效 (Vertex Effect)



設定扭曲範圍：中心點 (Center) 及半徑 (Radius)，設定扭曲角度 (Angle)。

5.4.5 環境遮蔽效果 (SSAO)

環境遮蔽效果 (Screen Space Ambient Occlusion 簡稱 SSAO) 可對場景植栽成像增加光影深度效果使畫面更為自然逼真，如下圖比較植栽。環境遮蔽運算還修正相機成像錐體區域物件影響陰影運算的問題。當然此效果是以 shader 程式運算的結果，需要高階的顯示卡。

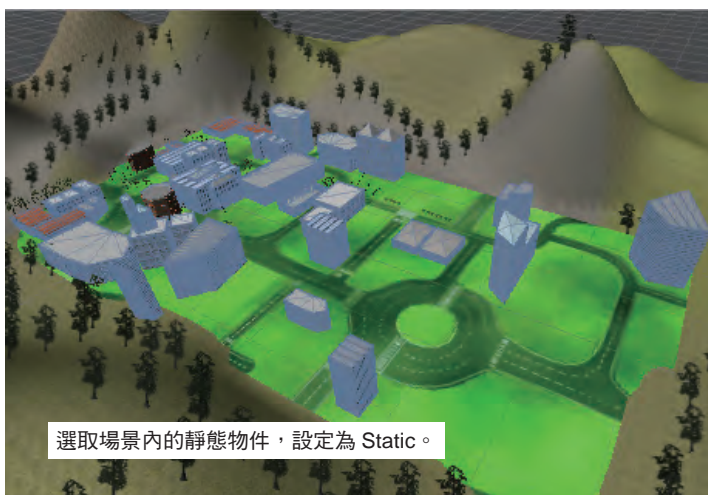


5.5 遮蔽剔除演算

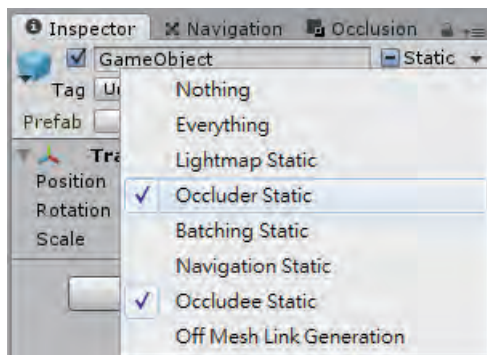
遮蔽剔除 (Occlusion Culling) 演算是為解決場景物件繁多時，透過切割空間為小單元，計算相機視角範圍內成像的小單元的技術。

5.5.1 遮蔽剔除設定

1. 選取場景內的靜態物件如建築物及 Terrain，設定為 Static。



計算遮蔽剔除的靜態物體分遮檔器 (Occluder) 及被遮檔物 (Occludee)。可兩項均勻勾選，但體積太小的物件不會再遮擋到其他物體時，可僅勾選 Occludee。



Occluder Static 遮擋器

Occludee Static 被擋物