

# DESARROLLO DE VIDEOJUEGOS Y REALIDAD VIRTUAL CON UNITY 3D

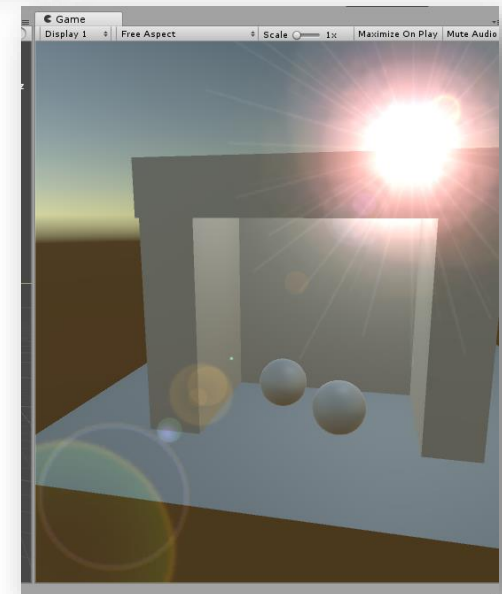
Tema 8: Iluminación y efectos gráficos

1



# ILUMINACIÓN Y EFECTOS GRÁFICOS

- Tipos de fuentes de luz
  - Panel Window-Lighting.
  - La luz ambiental (mínima).
    - Provocada por el skybox.
    - Provocada por el Environment Lighting.
  - Tipos de luces:
    - Directional Light (sólo afecta rotación).
    - Point Light (sólo afecta posición).
    - Spot Light (afecta posición y rotación).
    - Area Light (sólo baked).
  - Cookies. Troquel sobre la sombra.
  - Draw Halo. Dibuja un halo.
  - Flare. Reflejo de la luz en la lente.
  - Render Mode. Prioridad de la iluminación.
  - Culling Mask. Indicar qué capas iluminar.
- NOTA: En la cámara se determina si se ve o no el Skybox, pero no determina si ilumina o no.

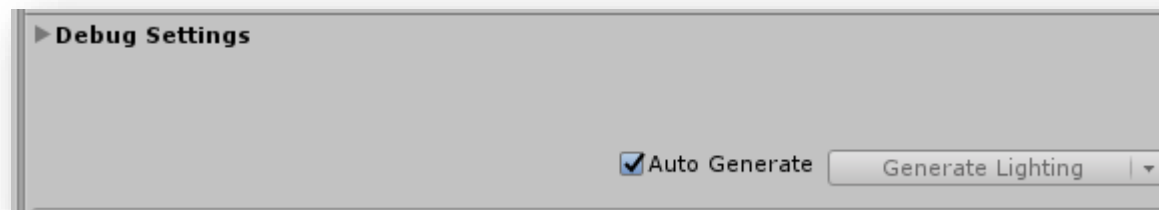
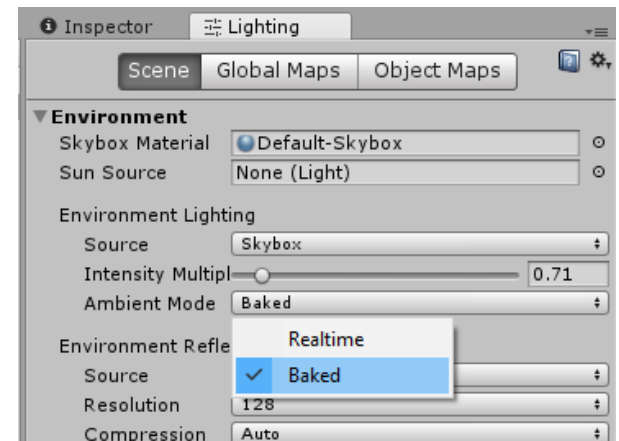
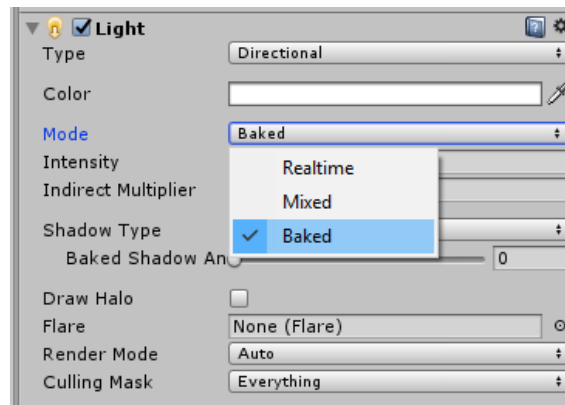
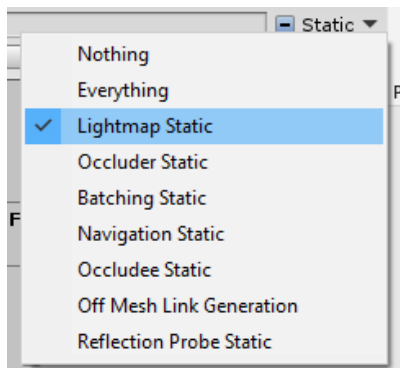


# ILUMINACIÓN Y EFECTOS GRÁFICOS

- Sombreado.
  - Se configura desde la fuente de luz y desde los objetos que lo reciban.
  - Desde la fuente de luz:
    - Soft Shadows. Más calidad, menos eficiencia.
    - Hard Shadows. Menos calidad, más eficiencia.
    - No Shadows.
    - Strength: intensidad.
    - Resolution: determina la calidad. La opción **“Use Quality Settings”** obtiene la configuración de “Edit->Project Settings->Quality”.
    - Bias, Normal Bias y Near Plane. Ajustes de la sombra. Permite eliminar ruido generado por el sombreado.
  - Desde los objetos.
    - En el componente Mesh Renderer:
      - Cast Shadows (proyectar sombras).
      - Recieve Shadows (recibir sombras)

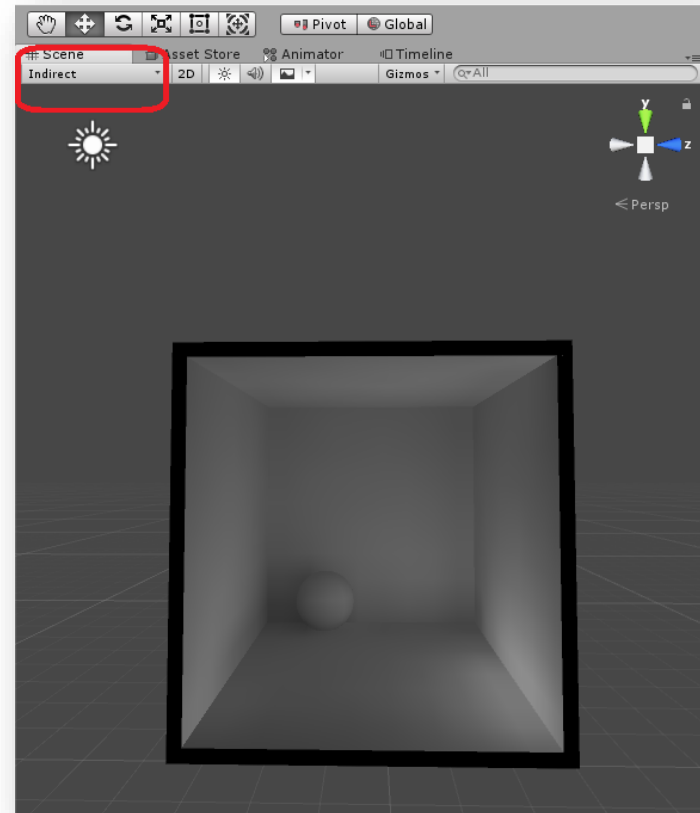
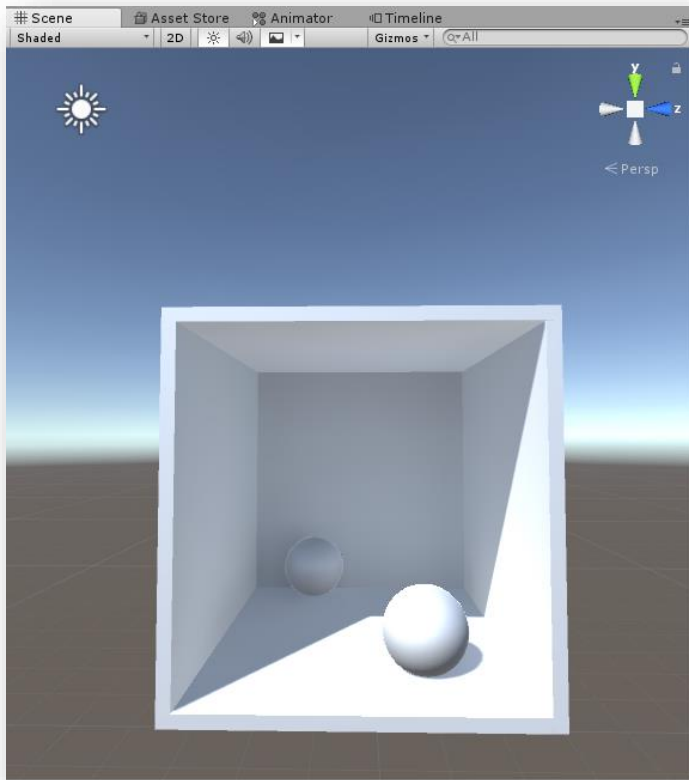
# ILUMINACIÓN Y EFECTOS GRÁFICOS

- Luces y modos de renderizado
  - Realtime (luz y escena)
  - Mixed (luz)
  - Baked (luz y escena)



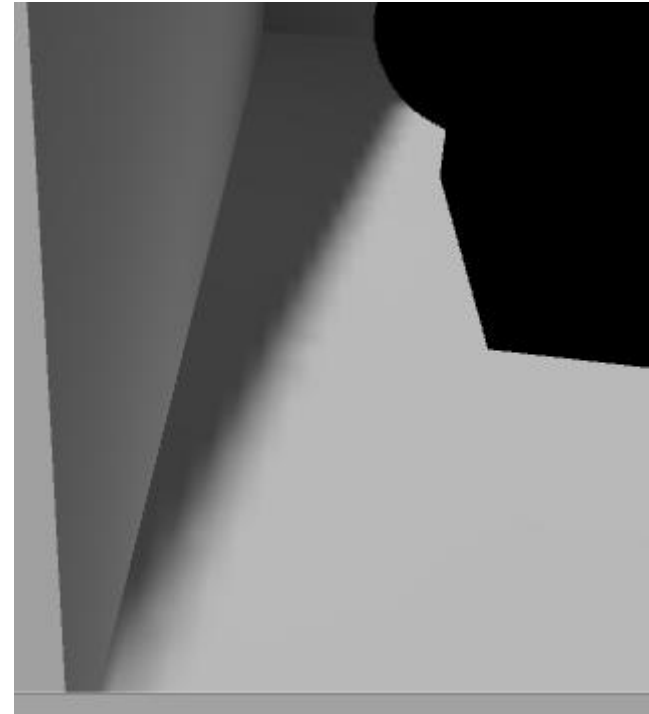
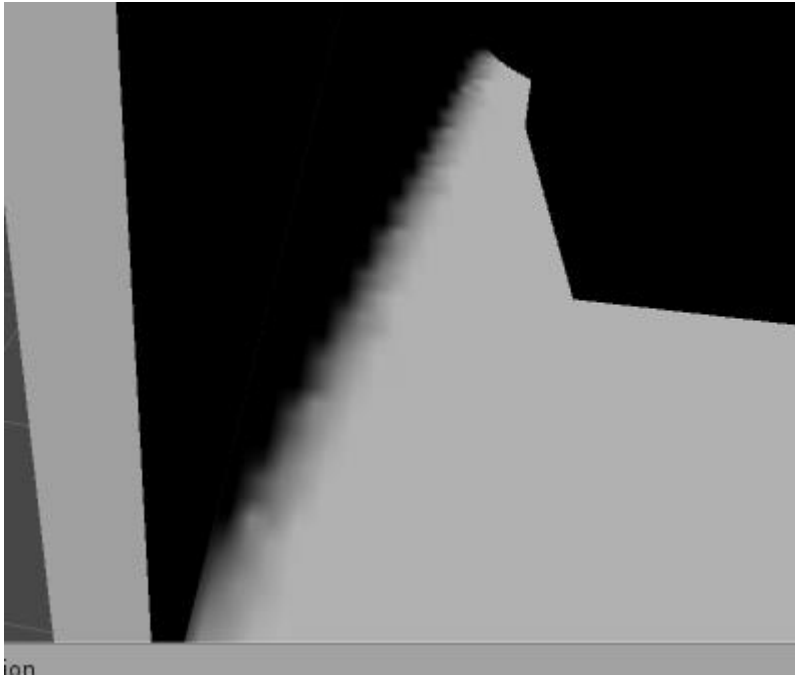
# ILUMINACIÓN Y EFECTOS GRÁFICOS

- Luces y modos de renderizado: luz indirecta (sólo se muestran los objetos estáticos)



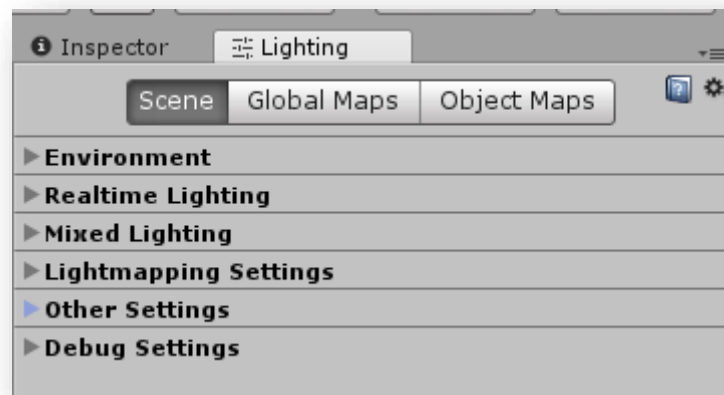
# ILUMINACIÓN Y EFECTOS GRÁFICOS

- Luces y modos de renderizado: luz indirecta.
  - Intensity multiplier (0 vs 1 en Directional Light)



# ILUMINACIÓN Y EFECTOS GRÁFICOS

- Ventana Lighting. Global Illumination.
  - Permite configurar la iluminación global (GI)
    - Pestaña Scene. Efectos de iluminación generales de la escena.
    - Pestaña Global Maps. Muestra los archivos de los mapas de luz.
    - Pestaña Object Maps. Muestra los mapas de luz y las máscaras de sombra para el gameobject seleccionado.



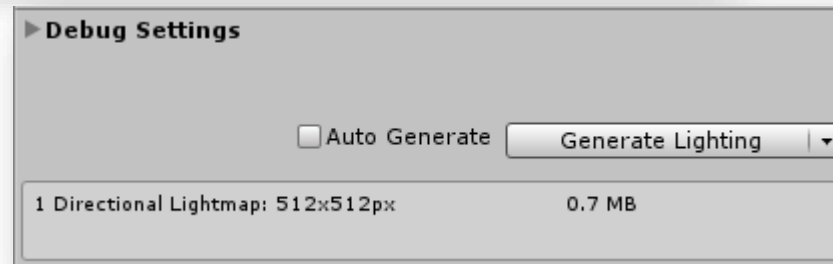
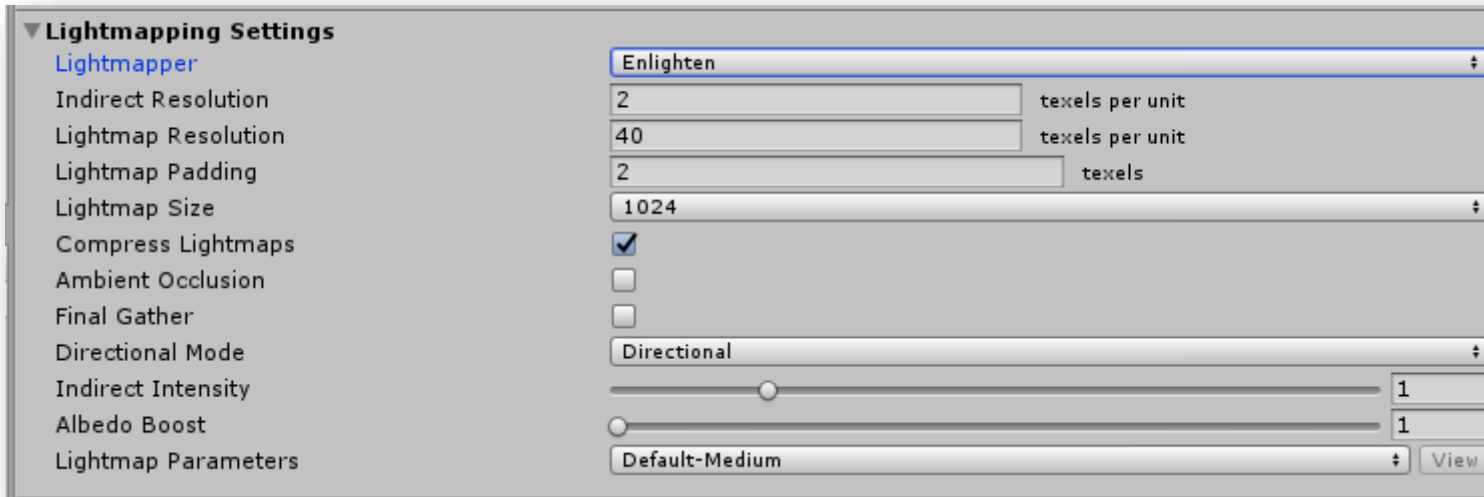
# ILUMINACIÓN Y EFECTOS GRÁFICOS

- Ventana Lighting. Pestaña Scene.
  - Environment (luz ambiental o luz difusa). Permite configurar el skybox, la iluminación ambiental y los reflejos.
  - Realtime Lighting.
    - Calcula la iluminación global en tiempo real (afecta a las fuentes de luz identificadas como **tiempo real** y a su efecto sobre objetos estáticos). Muy realista pero implica mucho consumo.
  - Mixed Lighting.
    - Precalcula la iluminación en tiempo de ejecución. La opción “Baked indirect” almacena la luz indirecta.
- Lightmapping Settings.
  - Configuración de la iluminación.
    - Bajar “Lightmap resolution” y “Lightmap size” para agilizar los cálculos.
    - Directional Mode: “Directional” más realista (tienen en cuenta los mapas de normales) y “Non-directional” más económicos (considera que todo la iluminación es plana y difusa).
    - Lightmap parameter: agilizar los cálculos utilizando una resolución “low” en desarrollo”.
- Other settings. Niebla, halos, cookies...



# ILUMINACIÓN Y EFECTOS GRÁFICOS

- Lightmap Settings:
  - Permite configurar el tamaño y la resolución del Lightmap.

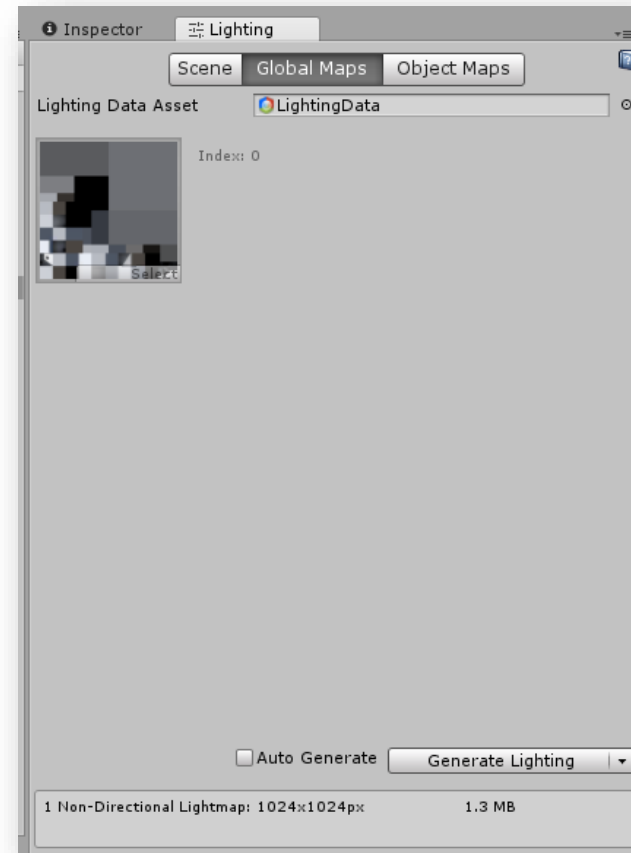
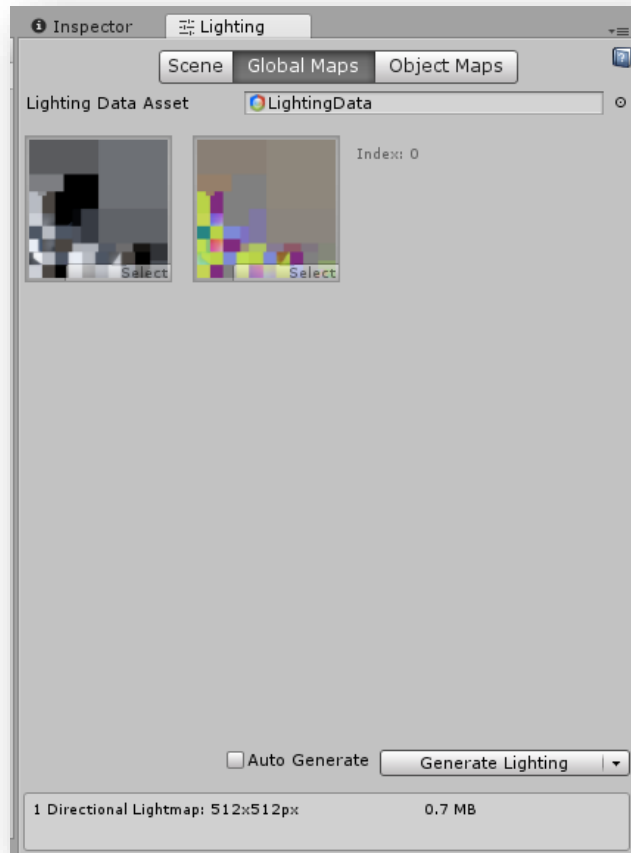


# ILUMINACIÓN Y EFECTOS GRÁFICOS

- Lightmap Settings (ver Global maps después de generar):
  - Indirect Resolution (indirect).
  - Lightmap Resolution (baked lightmap).
  - Lightmap Padding. Puede provocar líneas en las sombras.
  - Lightmap Size. Tamaño del lightmap.
  - Compress Lightmaps. Tamaño del mapa.
  - Ambient Occlusion. Definición mayor de la sombra.
  - Final Gather. Multiplica los rebotes de luz. Utilizar al final del proceso de iluminación. Elimina ruidos, limpia sombras...
  - Directional Mode.
    - Non-Directional. Un solo mapa con dos texturas.
    - Directional. Genera dos mapas. Consume más recursos.
  - Indirect Intensity. Intensidad de luz indirecta (incrementar la luz).

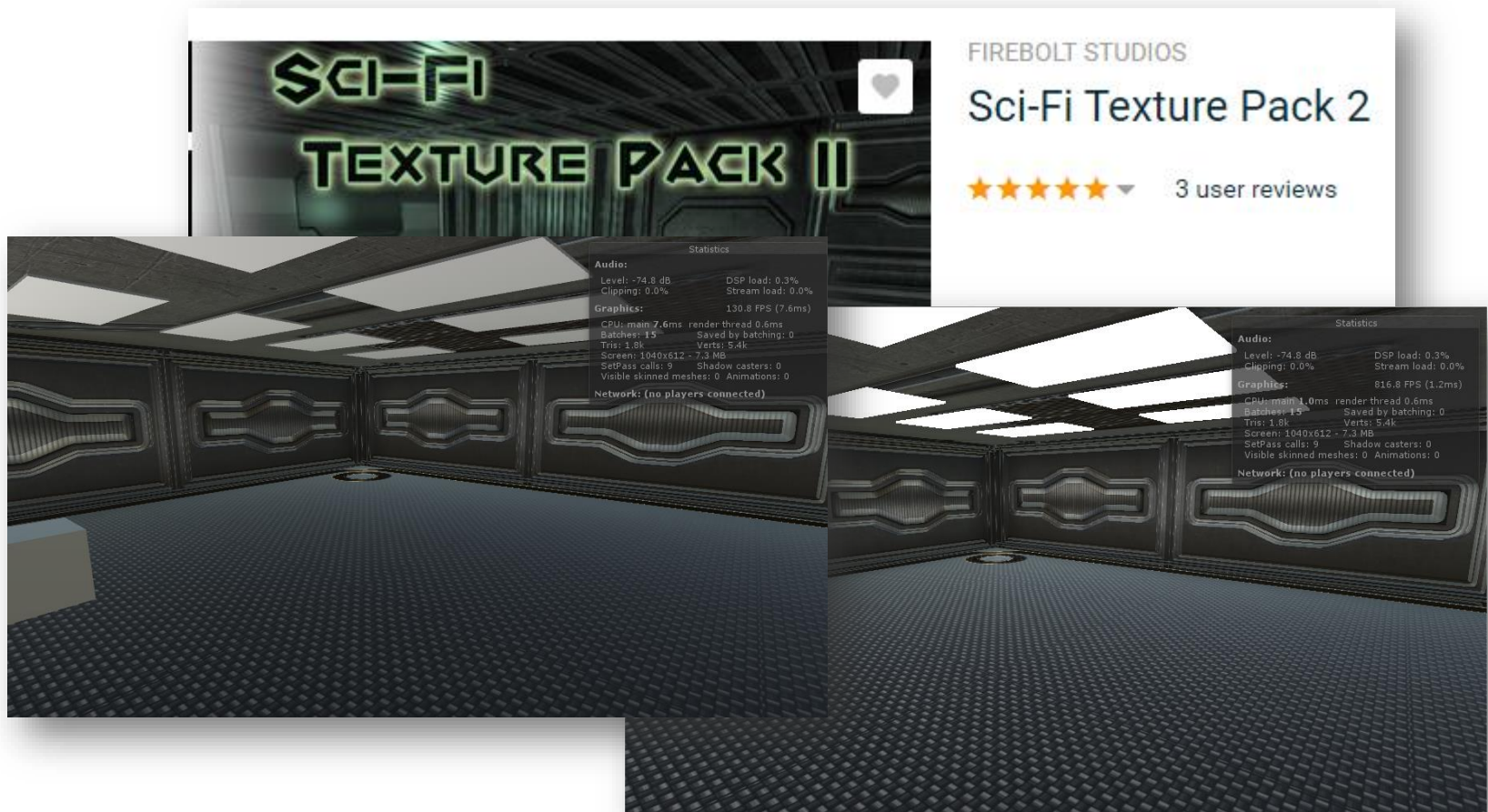
# ILUMINACIÓN Y EFECTOS GRÁFICOS

- Lightmap:



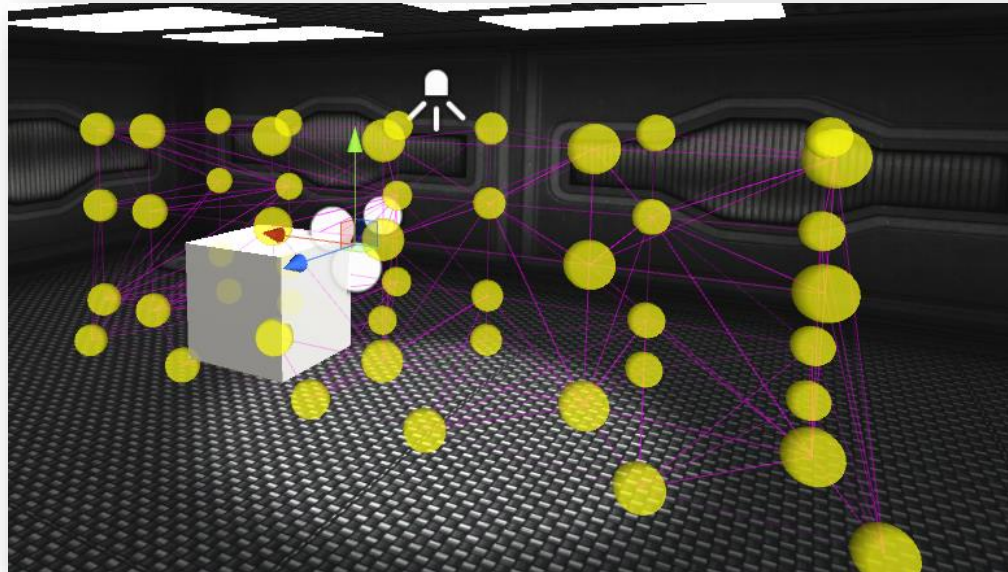
# ILUMINACIÓN Y EFECTOS GRÁFICOS

- Ejercicio: crear una sala sci-fi e iluminar.



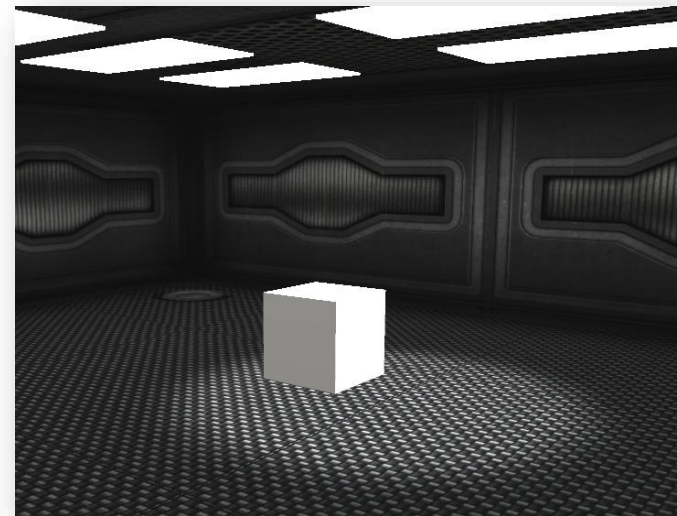
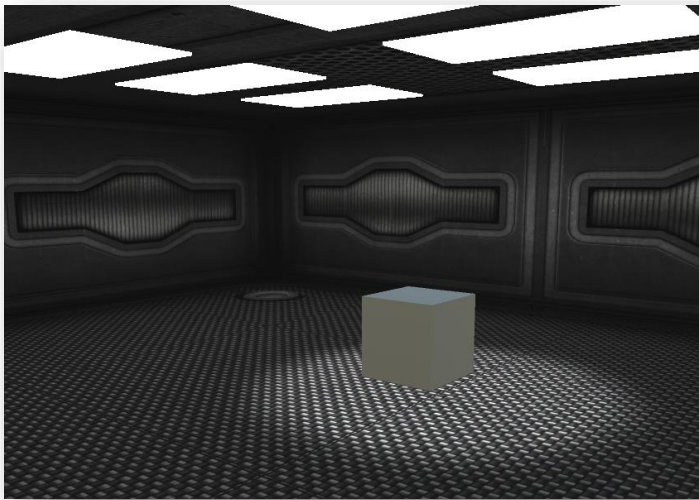
# ILUMINACIÓN Y EFECTOS GRÁFICOS

- Sondas de luz
  - Permiten iluminar de forma económica objetos dinámicos con luces estáticas.
  - Muestran puntos de luz y aplican la información en tiempo de ejecución.



# ILUMINACIÓN Y EFECTOS GRÁFICOS

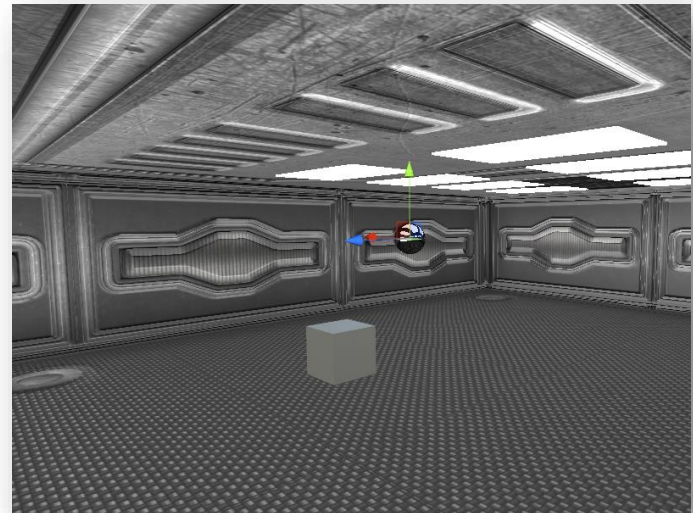
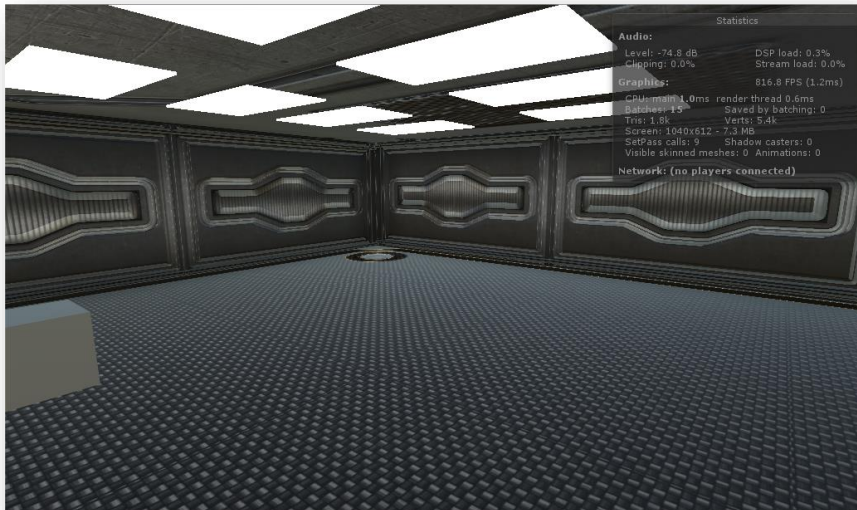
- Sondas de luz
  - Permiten iluminar de forma económica objetos dinámicos con luces estáticas.





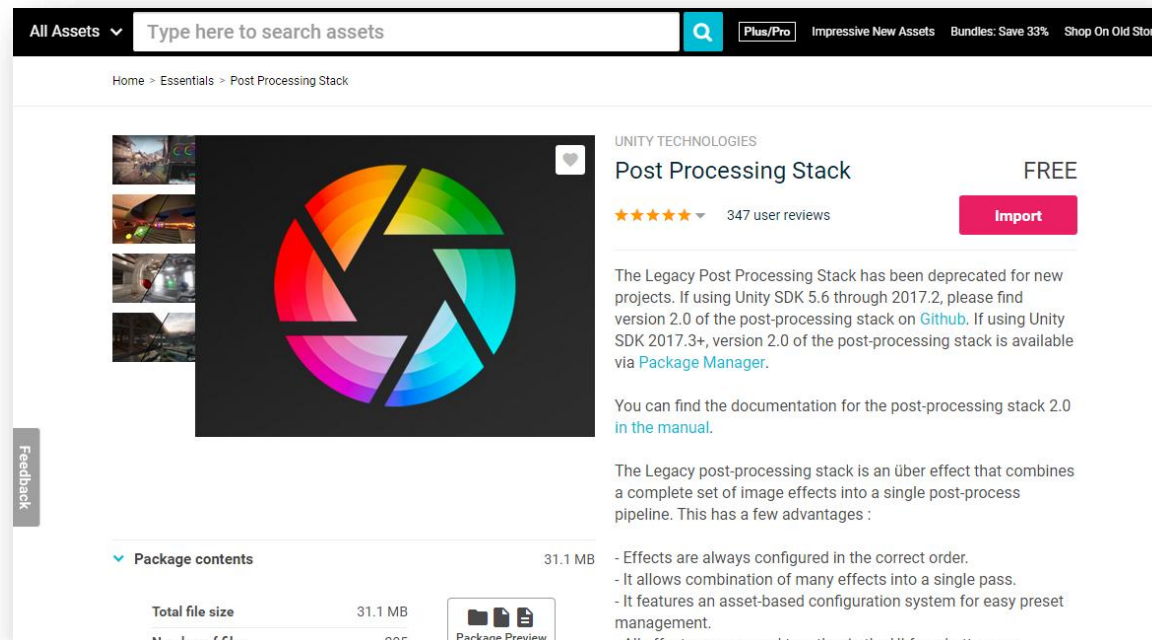
# ILUMINACIÓN Y EFECTOS GRÁFICOS

- Sondas de reflejos (reflection probes)
  - Permite generar las reflexiones de la imagen en los objetos.
  - Refleja lo que está delimitado por caja de reflexión.
  - Permite iluminar la escena mediante reflexiones.
  - El mapa generado por el probe son utilizados por el mapa de reflexión de los objetos (debe ser un objeto reflectivo) al pasar cerca.



# ILUMINACIÓN Y EFECTOS GRÁFICOS

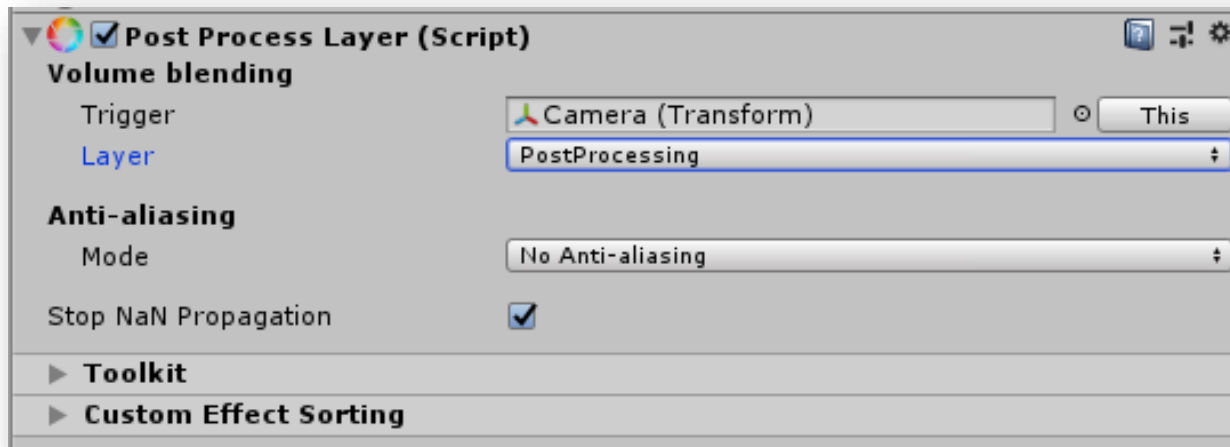
- Efectos de postprocesado.
  - Consiste en aplicar filtros al buffer de la imagen de la cámara antes de proyectar la imagen.
  - Proporciona efectos de calidad.
  - (Unity 2018+) Instalar desde el Package Manager.





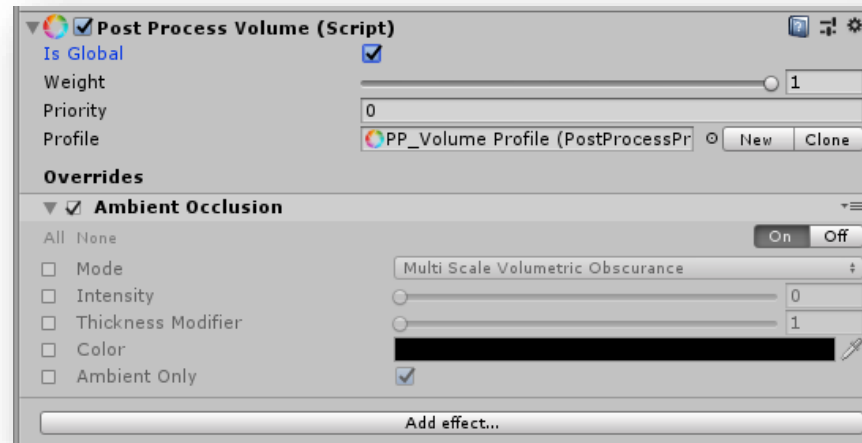
# ILUMINACIÓN Y EFECTOS GRÁFICOS

- Efectos de postprocesado.
- Agregar a la cámara el componente Post Process Layer, marcando “PostProcessing” como Layer.
- Configurar el modo de Anti-aliasing.



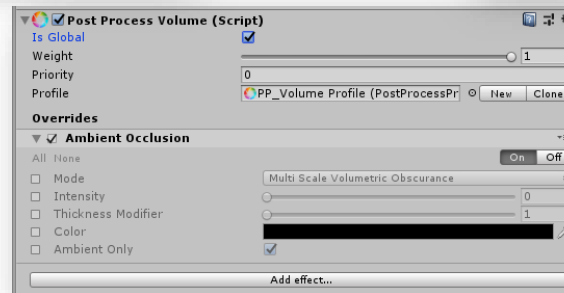
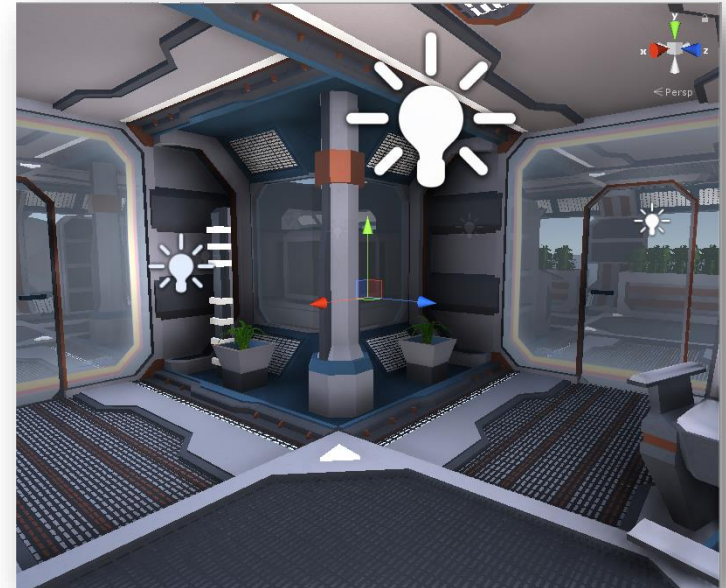
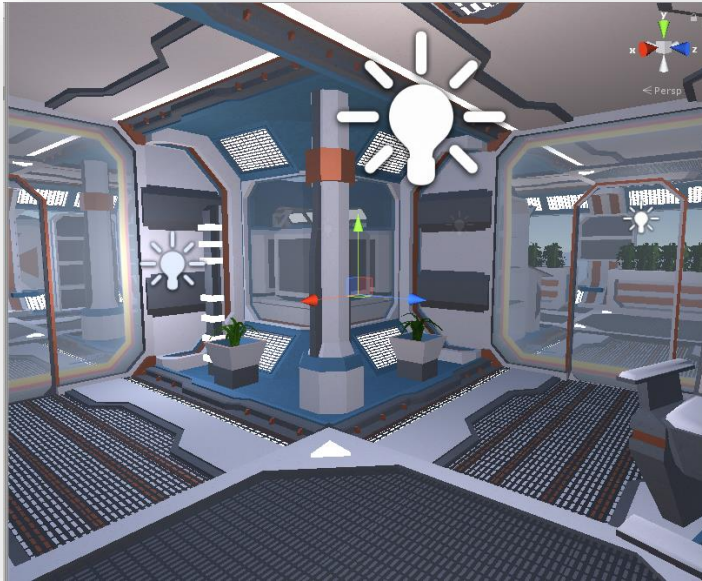
# ILUMINACIÓN Y EFECTOS GRÁFICOS

- Efectos de postprocesado.
- Agregar a un GameObject el componente “Post Process Volume” y seleccionar “PostProcessing” como Layer.
- Marcar el parámetro “Is Global”.
- Asignar o crear un “Profile”.
- Configurar los efectos.



# ILUMINACIÓN Y EFECTOS GRÁFICOS

- Efectos de postprocesado.



# ILUMINACIÓN Y EFECTOS GRÁFICOS

- Efectos de postprocesado.
  - Configurar el profile.
    - Ambient Occlusion. Sombreado de la escena.
    - Auto Exposure. Cálculo de la luminancia y configuración de la adaptación del ojo a los cambios en la iluminación.
    - Bloom. Provoca que las fuentes de luz se muestren más brillantes.
    - Chromatic aberration. Simula un defecto de las lentes que provoca que los colores no converjan correctamente.
    - Color Grading. Permite alterar el color de la escena.
    - Depth of view. Profundidad de campo.
    - Grain. Granularidad de la cámara.
    - Lens distortion. Distorsiones de la lente.
    - Motion blur. Desenfoque de movimiento.
    - Screen Space Reflection. Reflexiones de la escena (por ejemplo para crear efectos de charcos).
    - Vignette. Viñeta.
  - Post Processing Layer de la cámara.
    - Antialiasing. Reduce el aliasing.

# ILUMINACIÓN Y EFECTOS GRÁFICOS

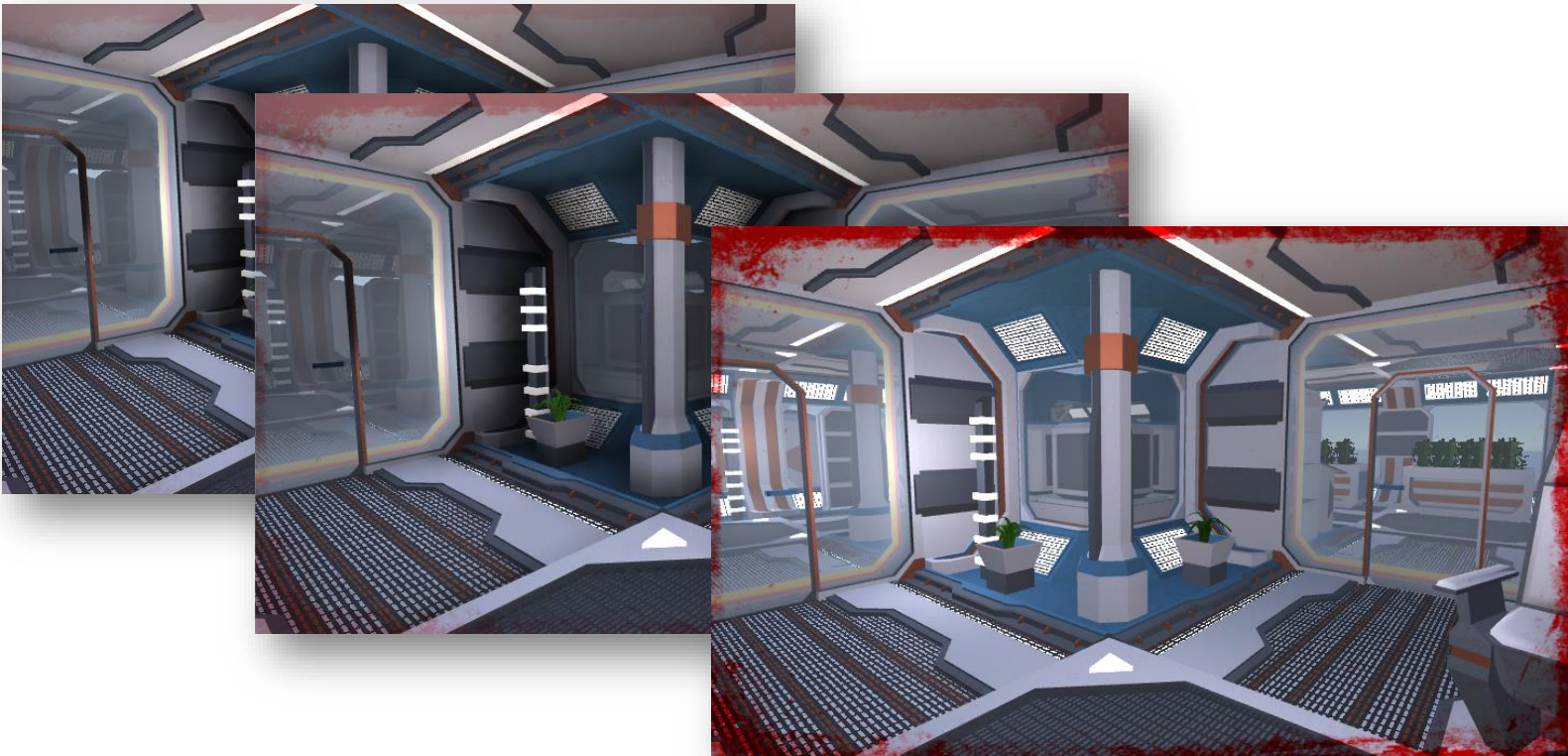
- Efectos de postprocesado.
  - Activación o desactivación desde código.
  - Ejemplo: Desactivación del efecto AmbientOcclusion

```
using UnityEngine;
using UnityEngine.Rendering.PostProcessing;

public class Vignete : MonoBehaviour
{
    PostProcessVolume ppVolume;
    AmbientOcclusion ambientOcclusionLayer;
    void Start()
    {
        ppVolume = gameObject.GetComponent<PostProcessVolume>();
        ppVolume.profile.TryGetSettings(out ambientOcclusionLayer);
        ambientOcclusionLayer.enabled.value = false;
    }
}
```

# ILUMINACIÓN Y EFECTOS GRÁFICOS

- Efectos de postprocesado.
  - Modificación de las propiedades de los efectos
  - Ejemplo: viñeta de daño





# ILUMINACIÓN Y EFECTOS GRÁFICOS

- Efectos de postprocesado.
  - Modificación de las propiedades de los efectos
  - Ejemplo: viñeta de daño



# ILUMINACIÓN Y EFECTOS GRÁFICOS

- Efectos de postprocesado.
  - Modificación de las propiedades de los efectos
  - Ejemplo: viñeta de daño

```
using UnityEngine;
using UnityEngine.Rendering.PostProcessing;

public class Vignette : MonoBehaviour
{
    PostProcessVolume ppVolume;
    Vignette vignetteLayer;
    void Start()
    {
        ppVolume = gameObject.GetComponent<PostProcessVolume>();
        ppVolume.profile.TryGetSettings(out vignetteLayer);
    }
    void Update()
    {
        vignetteLayer.opacity.value = vignetteLayer.opacity.value + Time.deltaTime * 0.1f;
    }
}
```