



# UI, Lighting, Post processing

Instructors: Jingxuan Chen, Dario Quintero, Shangyi Zhu,  
Jeffrey Wang

---

# Table of contents

**01**

## **UI**

How to edit & organize UI

**02**

## **2D Light**

2d light concepts

**03**

## **3D Light**

3d light concepts

**04**

## **Post Processing**

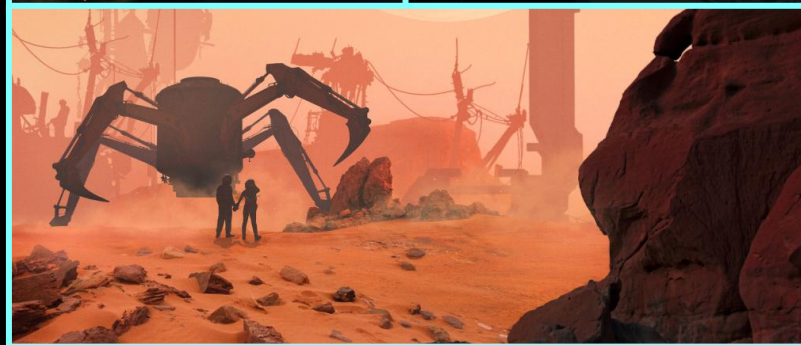
Effects such as bloom or blur



01

# UI in Unity

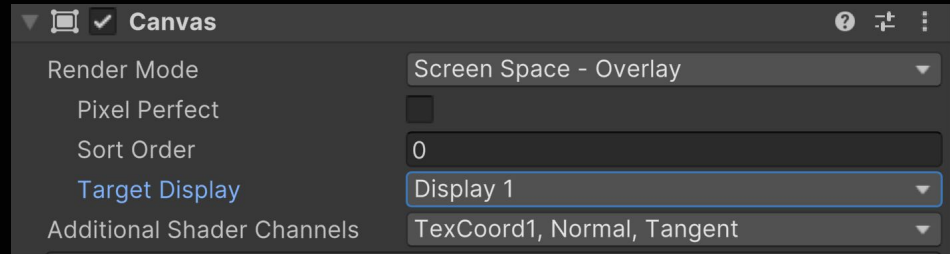
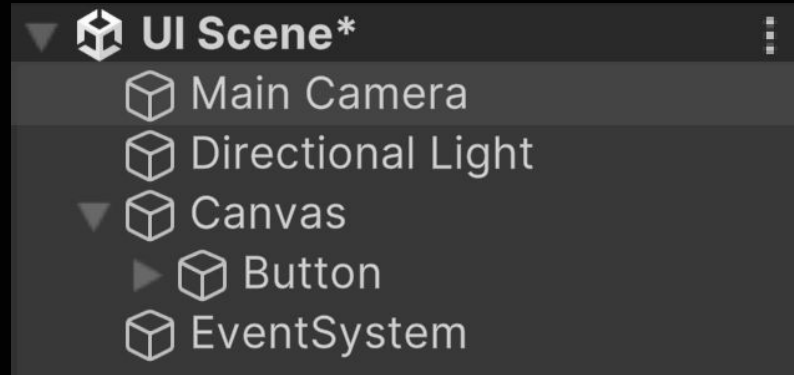
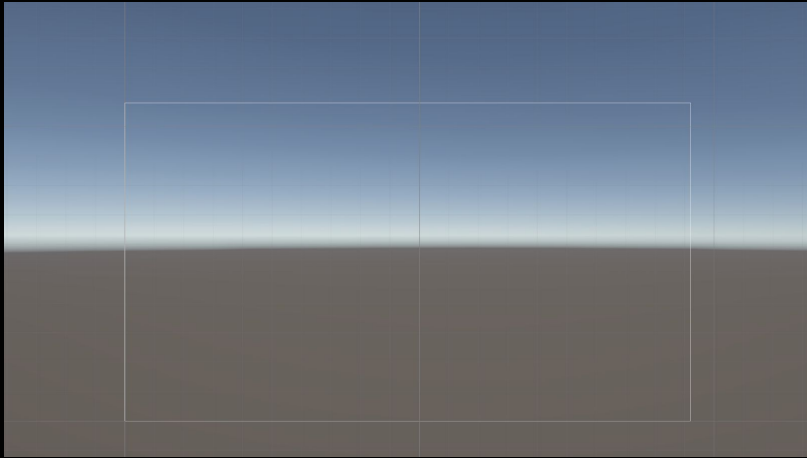
Panels projected into the space



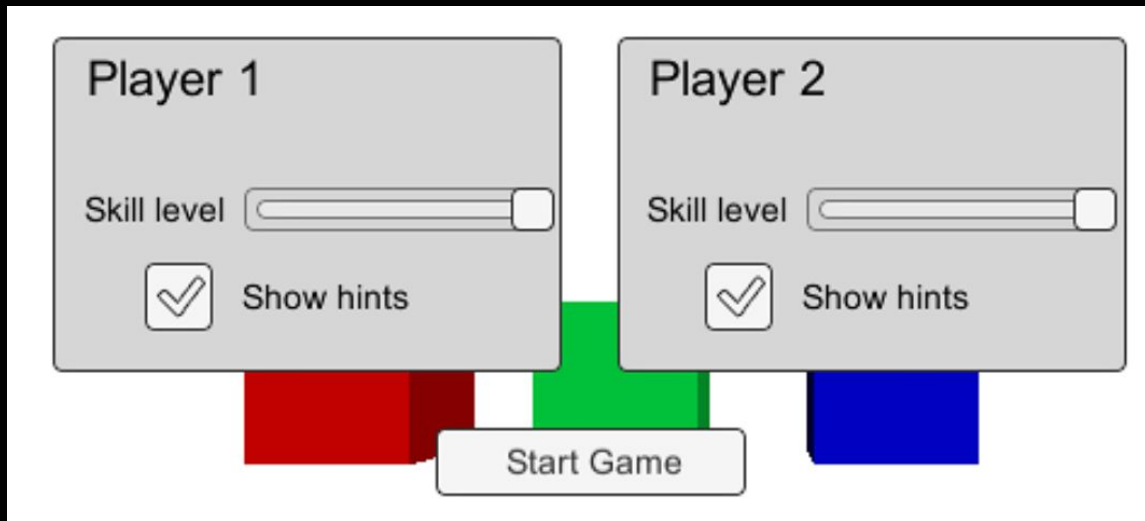


# Canvas

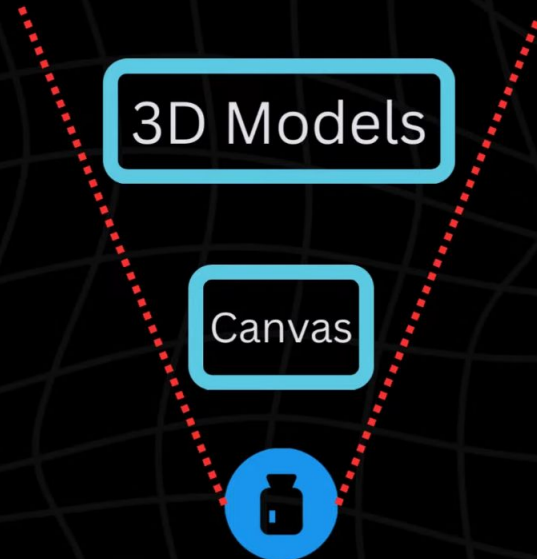
A **User Interface (UI) Canvas** is the base area where you place **all your UI elements**. Think of it as a blank sheet where you can place your buttons, text, images, and other controls.



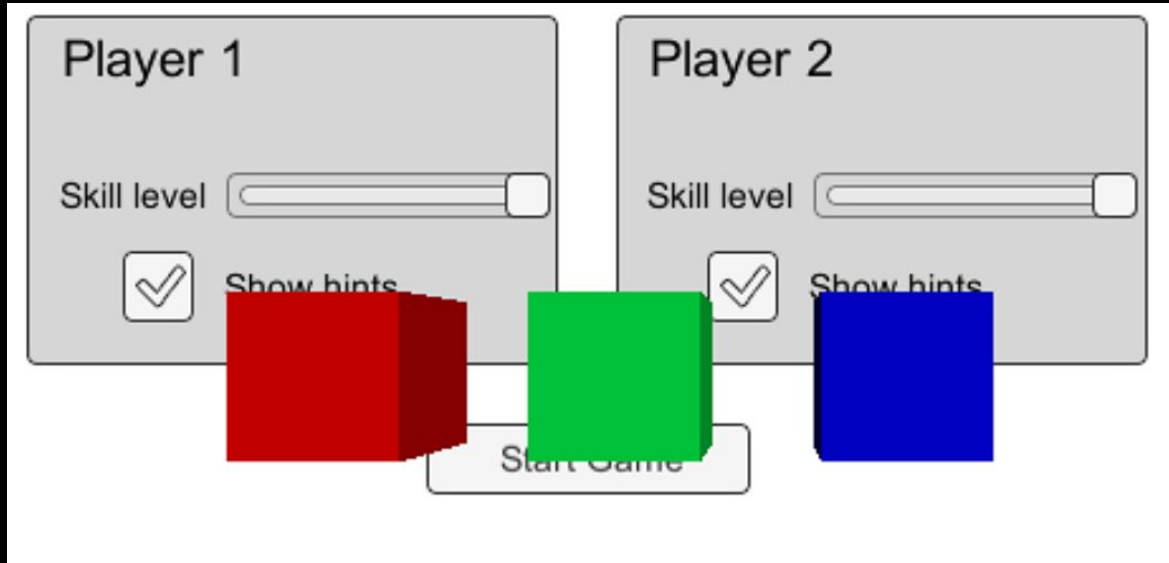
# Screen Space - Overlay



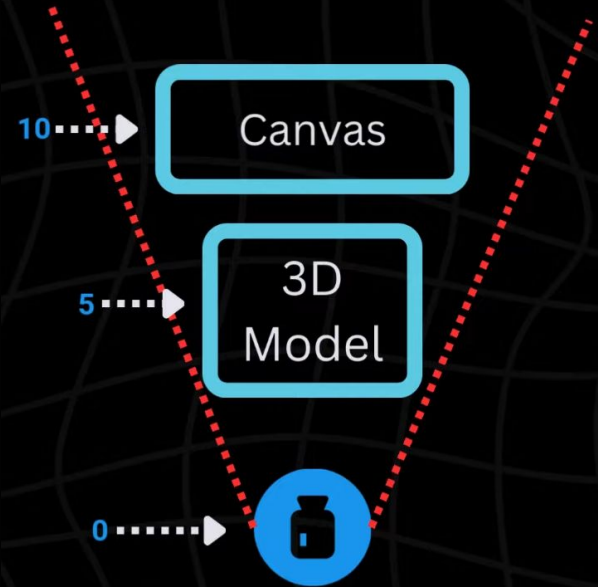
SCREEN SPACE - OVERLAY



# Screen Space - Camera



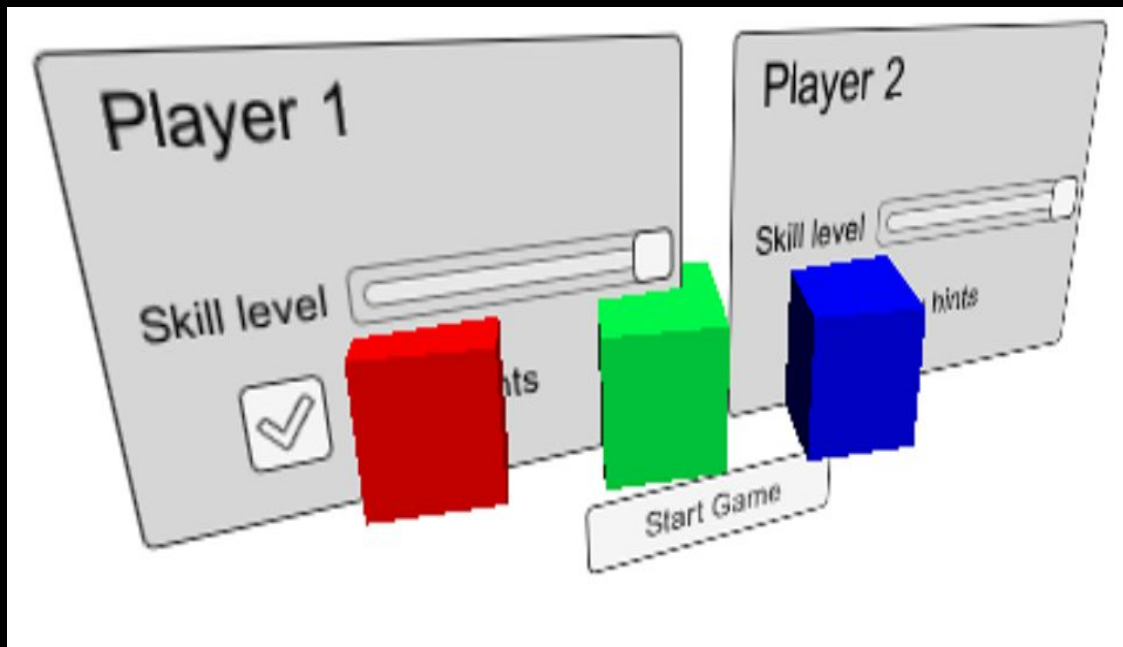
## SCREEN SPACE - CAMERA



# Screen Space - Camera



# World Space



## WORLD SPACE

World Canvas

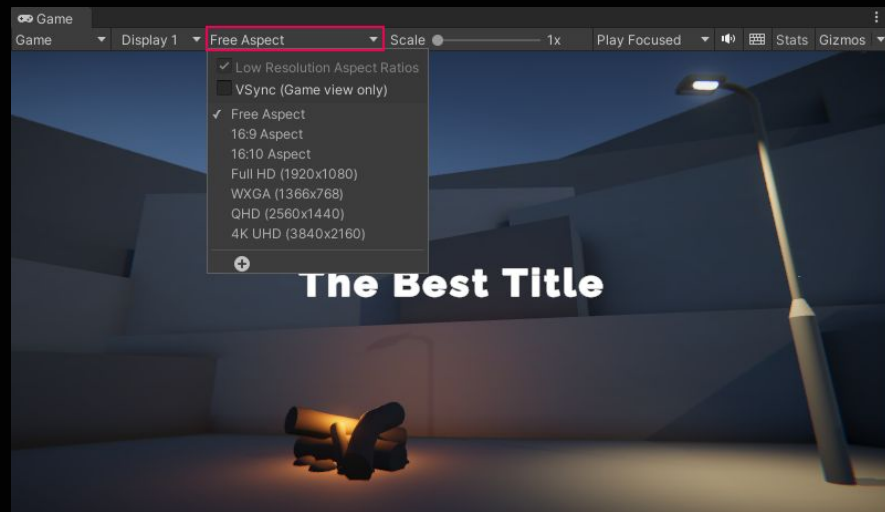
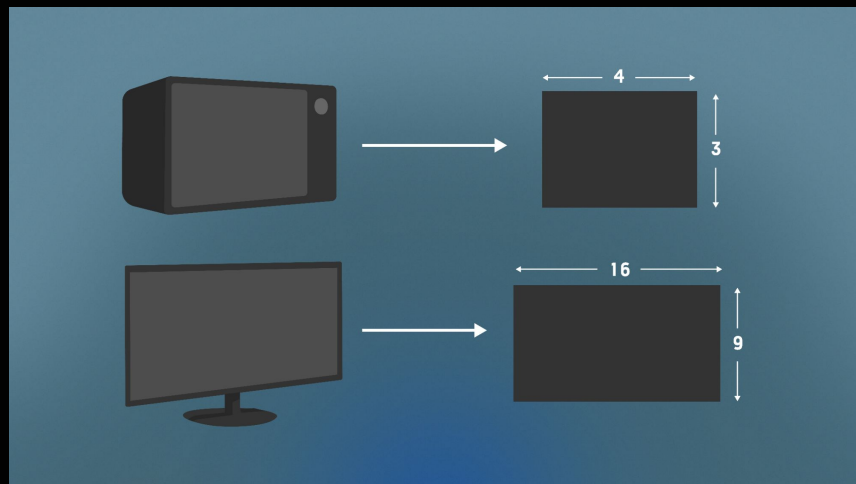
Model

World Canvas



# Aspect Ratio

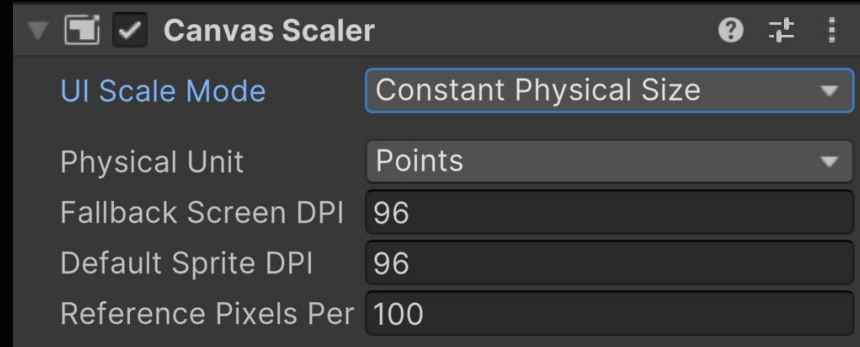
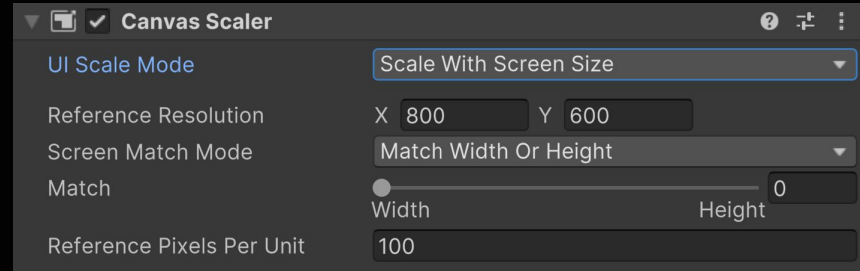
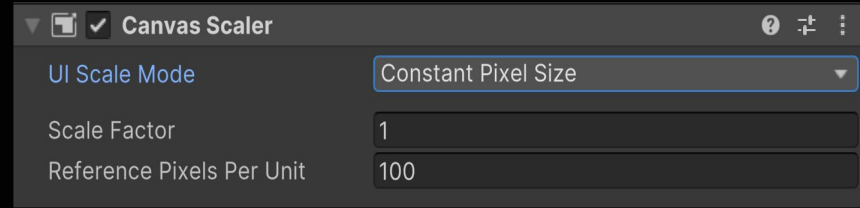
The aspect ratio describes the relationship between the width and the height of a screen.



# Resolution

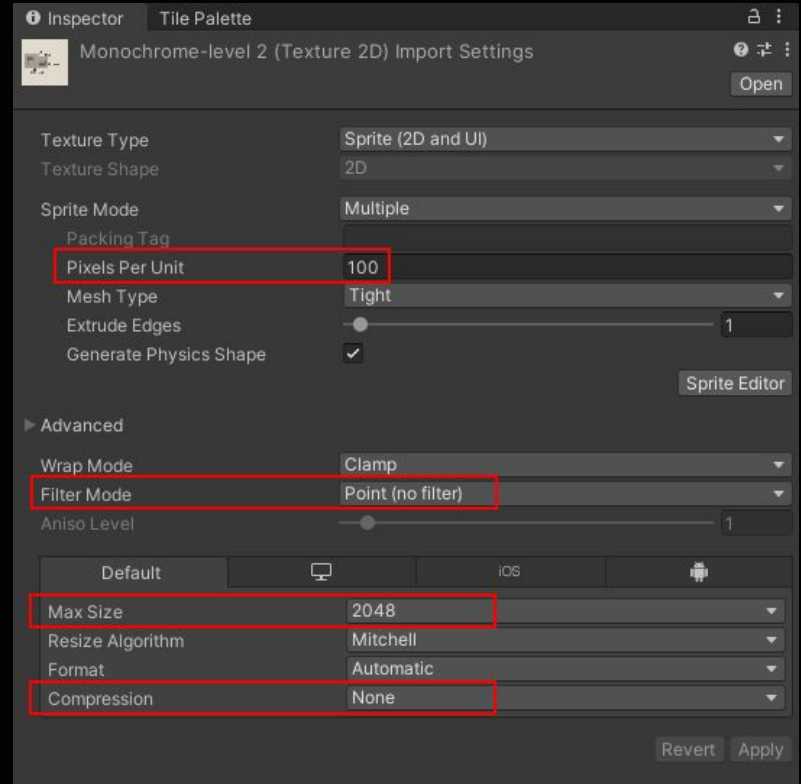
The amount of pixels on the screen

Constant pixel size or screen space scaling



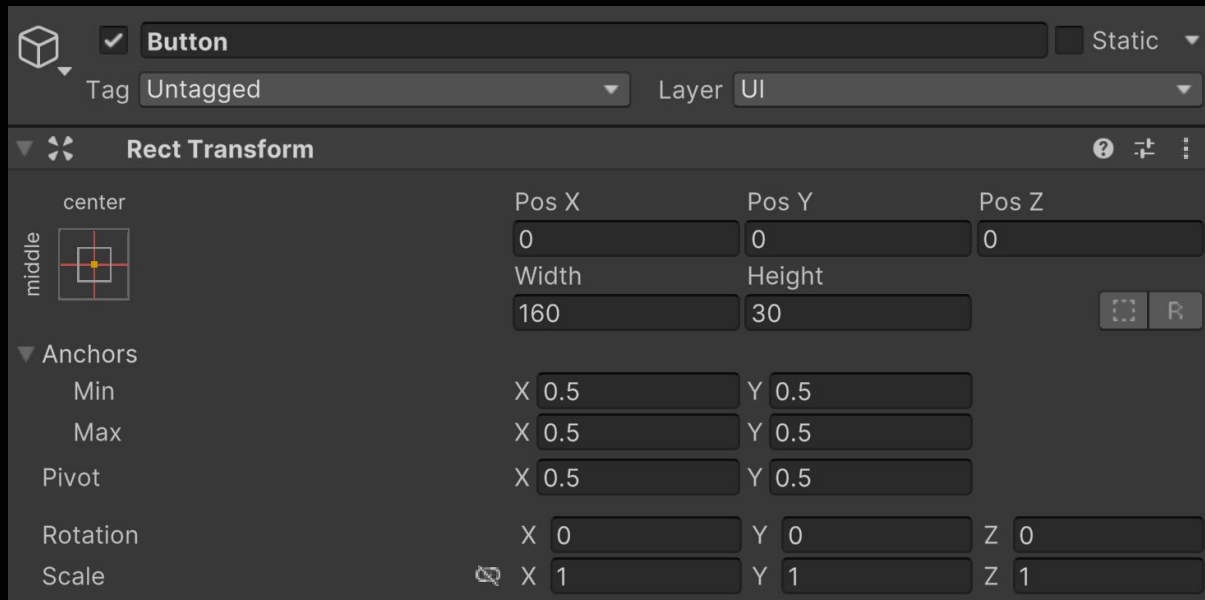
# Resolution

For 2D textures and sprites, set the PPU to determine sampling at different levels



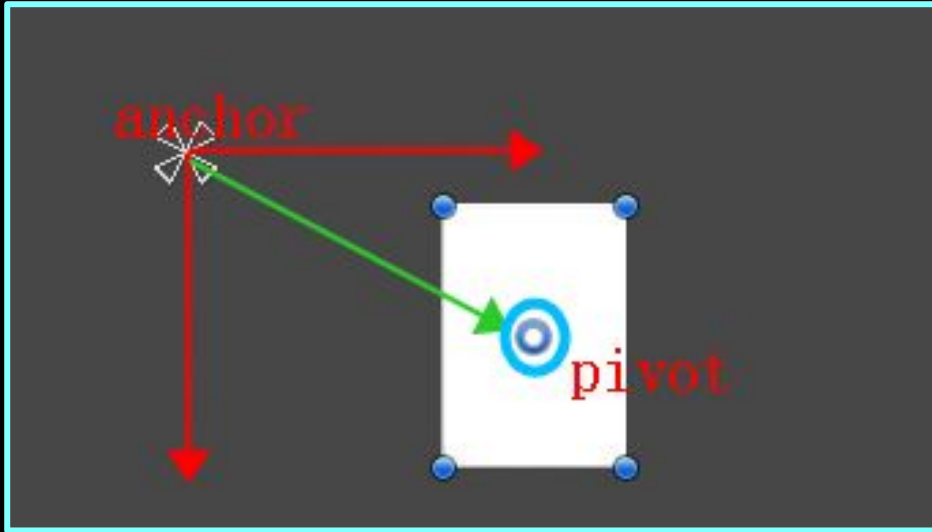
# Rect Transform

Rect Transform is a new transform component that is used for all UI elements instead of the regular Transform component.



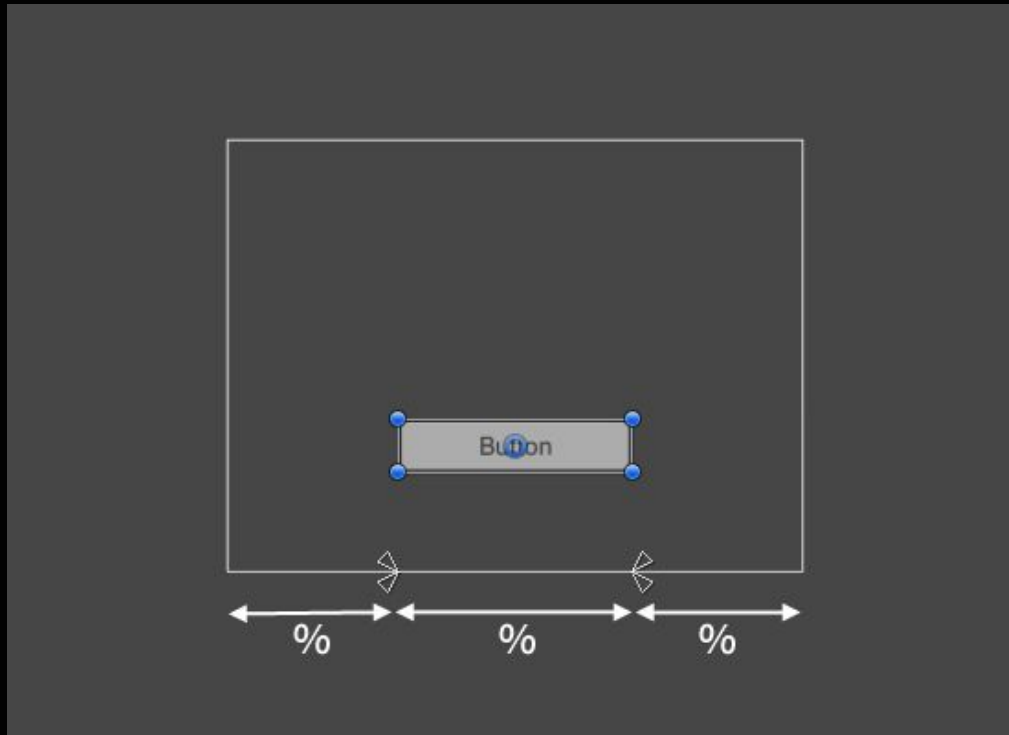
# Anchors

Anchors are percentage based thresholds that determine scaling of objects when aspect ratios change.



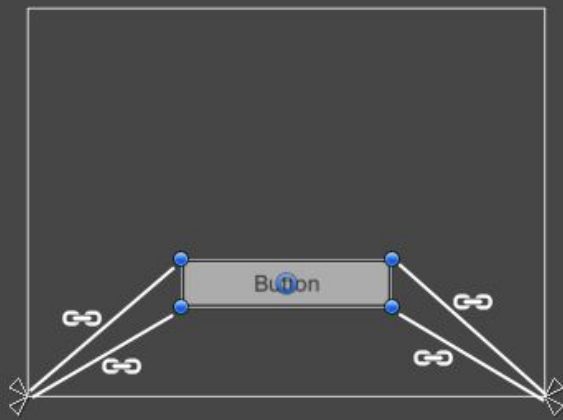
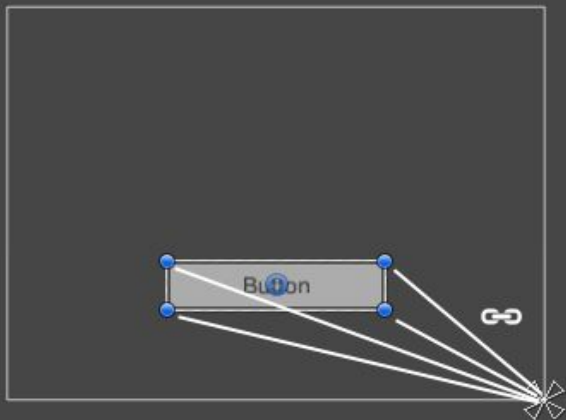
# Anchors

Basic Layout | Unity UI | 1.0.0



# Anchors

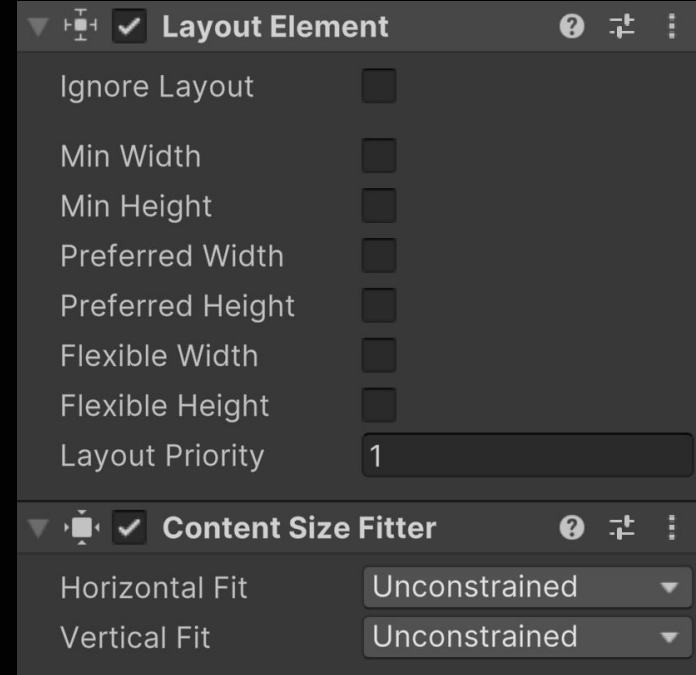
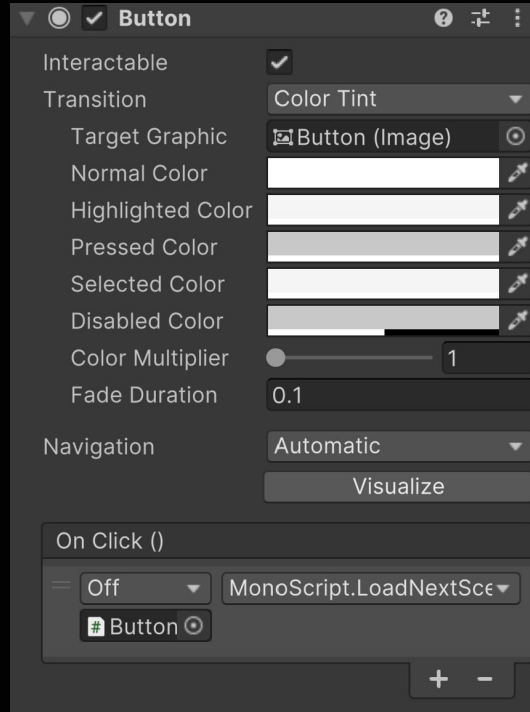
Basic Layout | Unity UI | 1.0.0



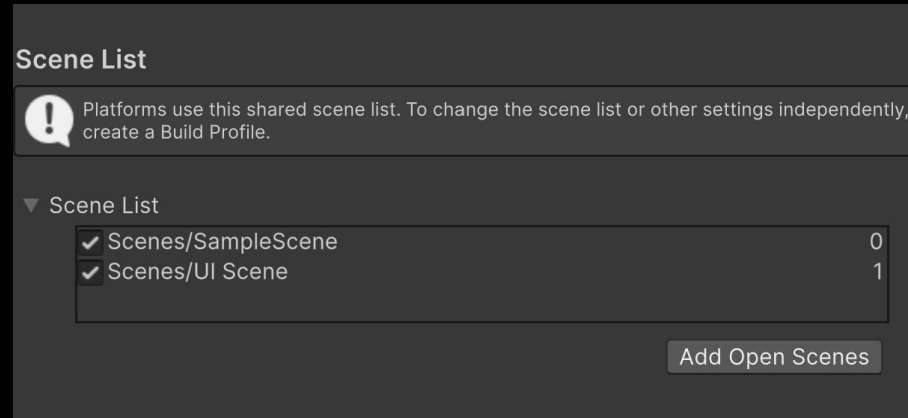
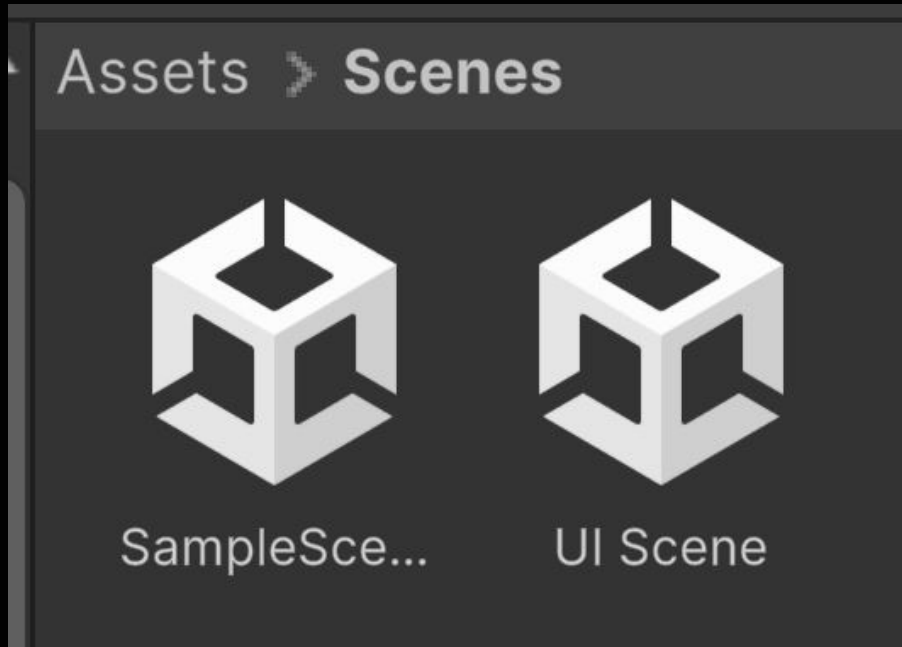
# UI- Component

- UI Button
- UI Images
- UI Text
- UI Event and Event Trigger
- UI Slider
- UI Transitions

[UI Components - Unity Learn](#)



# Scene Loading



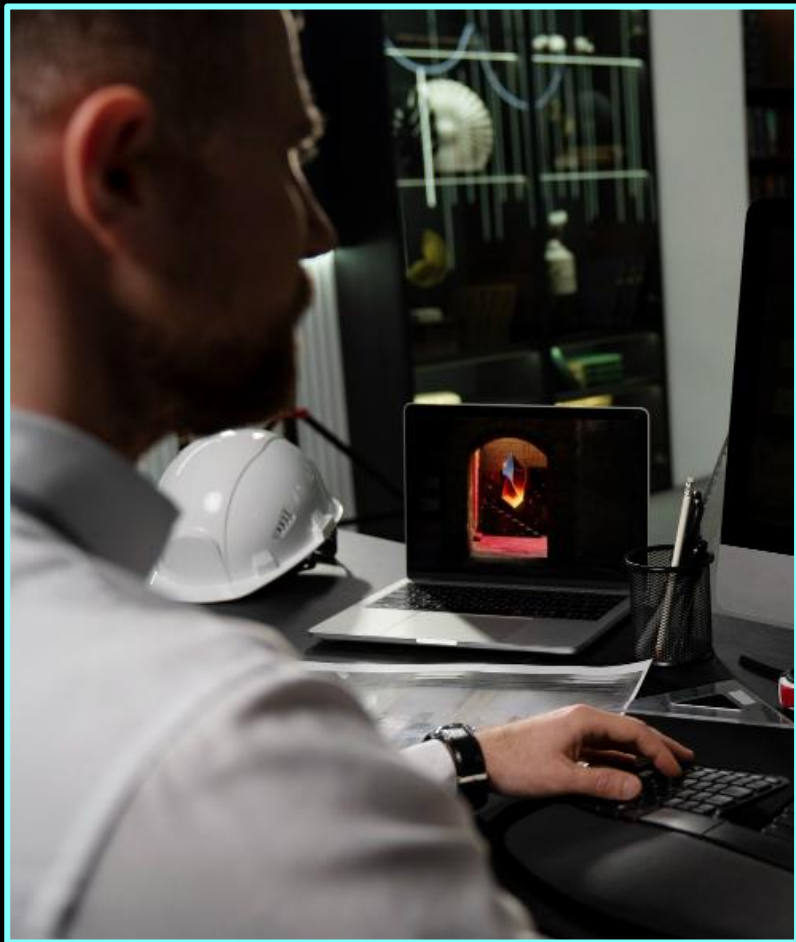
# Scene Loading

```
public void NextScene()
{
    if(SceneManager.GetSceneByBuildIndex(SceneManager.GetActiveScene().buildIndex +1)!= null)
    {
        SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex +1);
    }
}
```

0 references

```
public void PreviousScene()
{
    if (SceneManager.GetSceneByBuildIndex(SceneManager.GetActiveScene().buildIndex - 1) != null)
    {
        SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex - 1);
    }
}
```

**Demo In Unity**



02

## 3D Lighting

# Different Lights in Unity

## Point

located at a point in the Scene and emits light in all directions equally



## Spot

Located at a point in the Scene and emits light in a cone shape



---

## Directional

Infinitely far away and emits light in one direction only

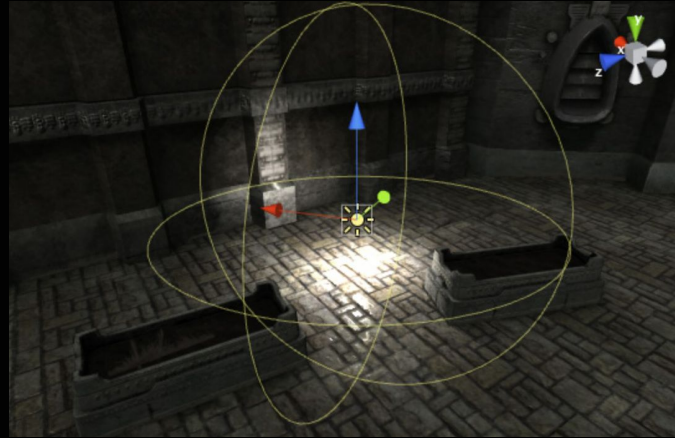
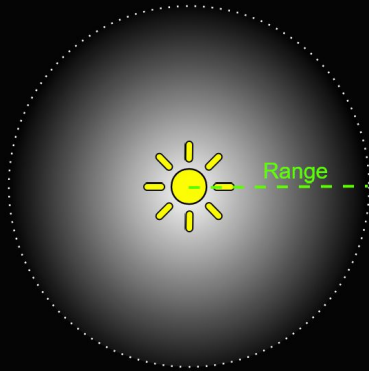


## Area

Defined by a rectangle or disc in the Scene

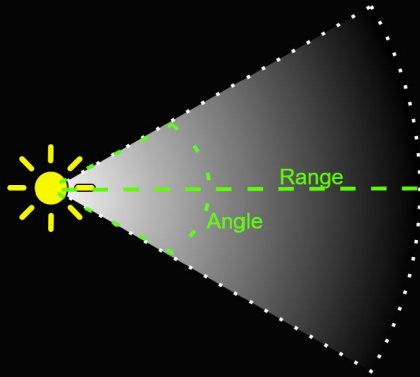


# Point Light



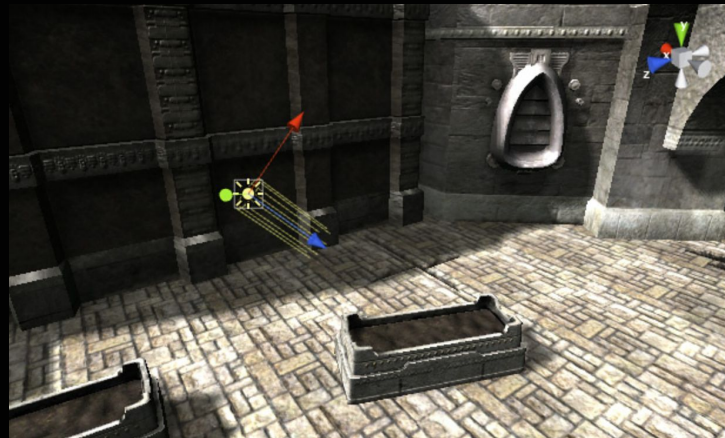
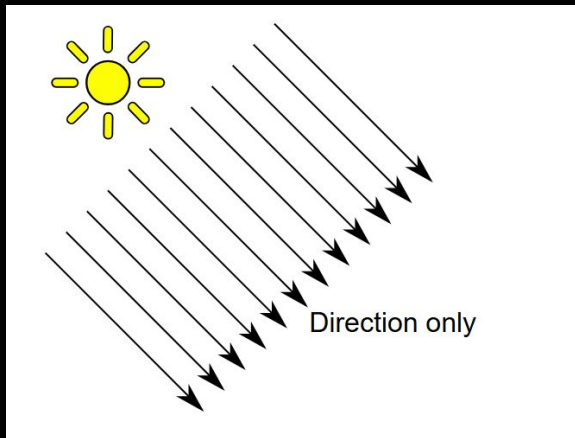
- Point Light located at a point in space and sends light out in all directions equally

# Spot Light



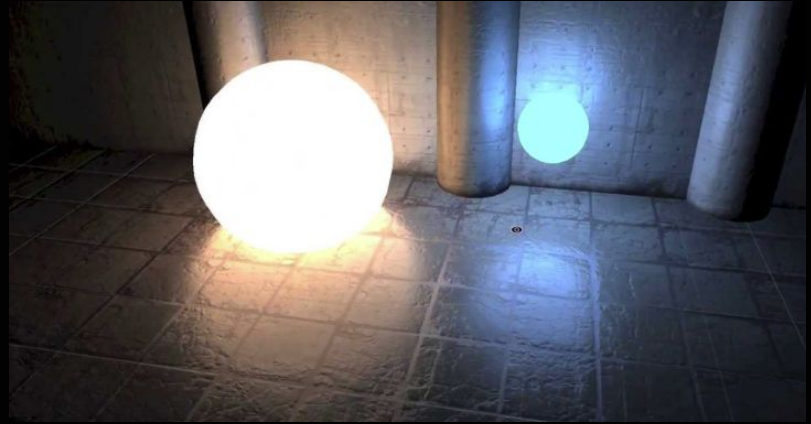
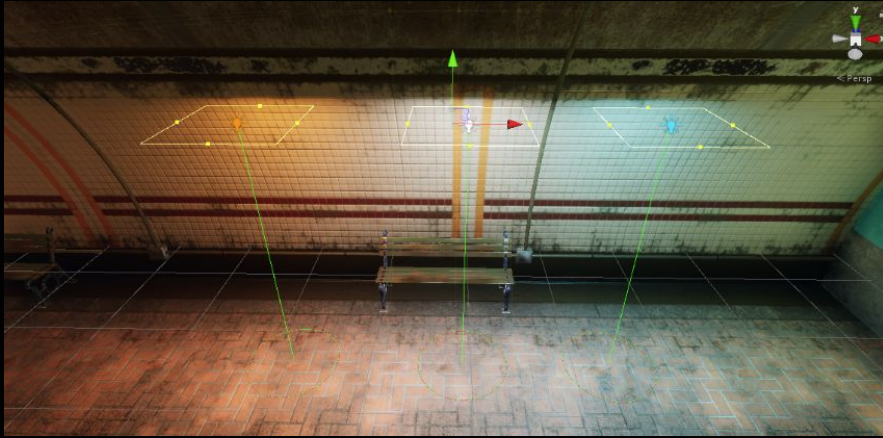
- Spot Light has a specified location and range over which the light falls off
- Can be used as flashlights, car headlights and searchlights

# Directional Light



- Directional lights represent large, distant sources that come from a position outside the range of the game world
- It's usually Sunlight and comes in the default scene

# Area Light



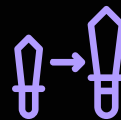
- Point Light located at a point in space and sends light out in all directions equally
- In HRDP mode, it is real time, but in other modes, can only be baked (computationally intensive)

# Lighting Modes



## Realtime

Calculates and updates the lighting of these Lights every frame at run time



## Baked/Mixed

Pre-calculates the illumination from these Lights before run time



# Baked Lighting - Light Maps

## Light Maps:

- The pre-calculating the brightness of surfaces in a **Scene** is stored in a Texture called a **lightmap** for later use.
- It contains information about light hitting the *surfaces* in your scene
- Good for static environments
- Can bake for a large scene



# Baked Lighting - Light Maps



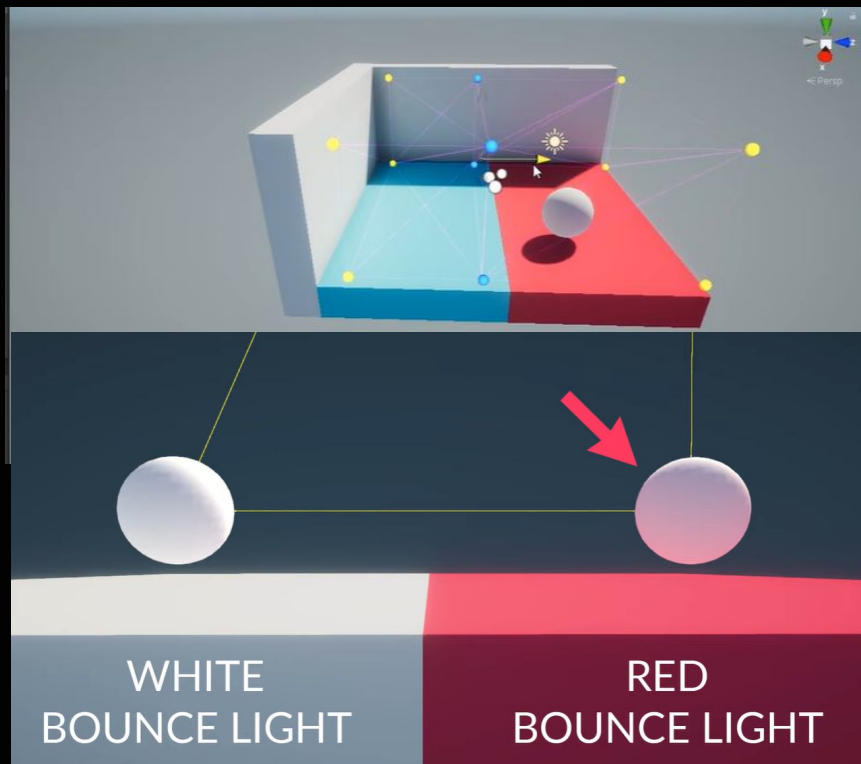
- The one on the right more computationally expensive, would be too laggy if not baked



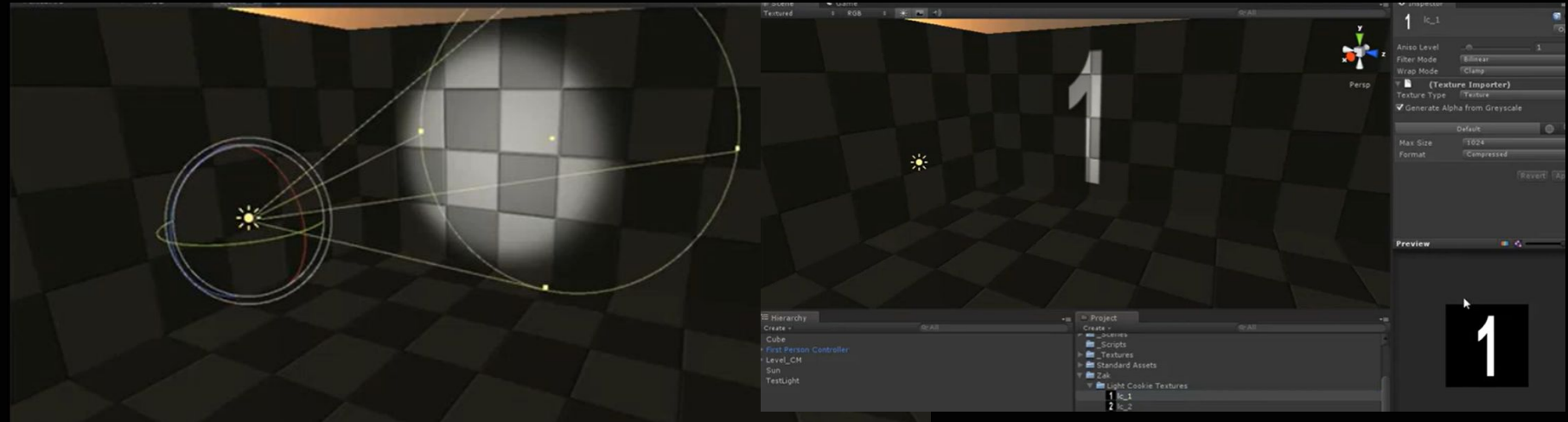
# Baked Lighting - Light Probes

## Light Probes:

- It provides high quality lighting (including indirect bounced light) on **moving objects** in your scene.
- light probes store information about light passing through *empty space in your scene.*
- Can set points so that it know which part is bright/dark/reflects color
- Good light probe tutorial [here](#)



# Light Cookies



## Light Cookie:

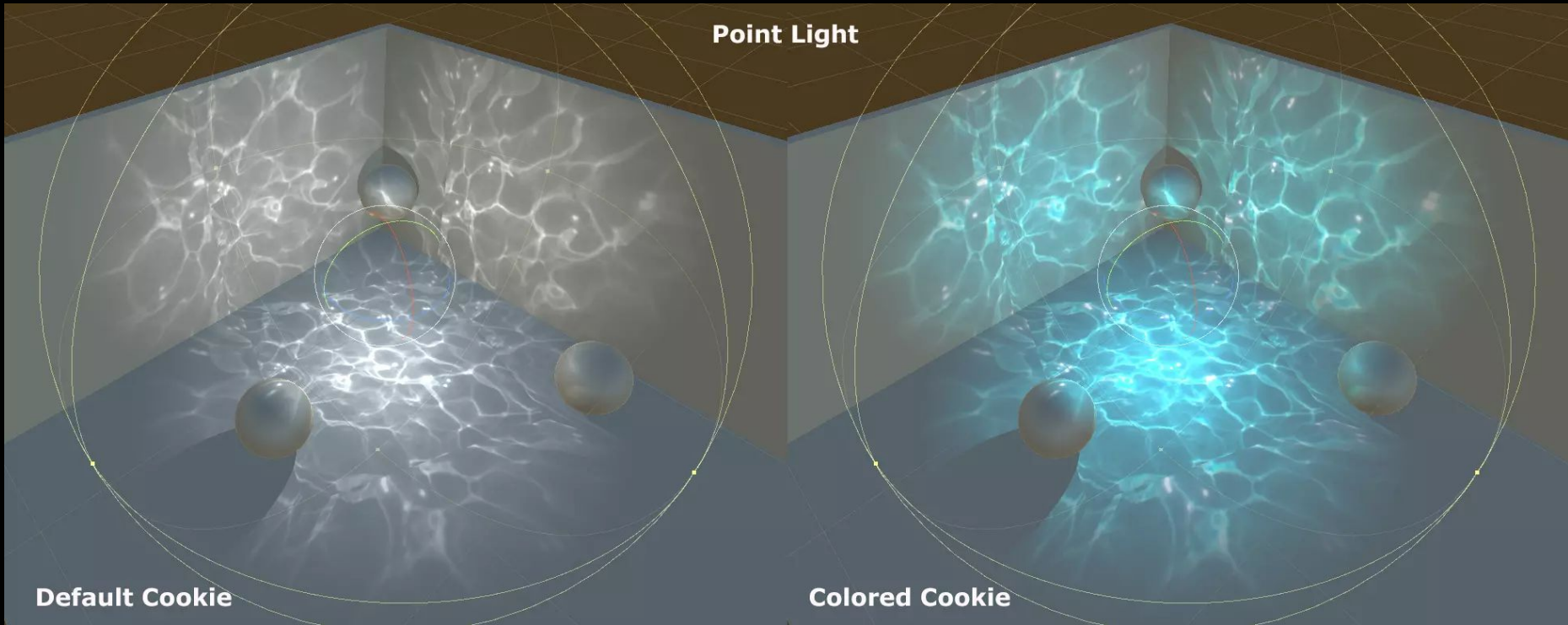
- Provides a mask for the light
- In this example, the cookie texture of 1 is applied to a point light

# Light Cookies

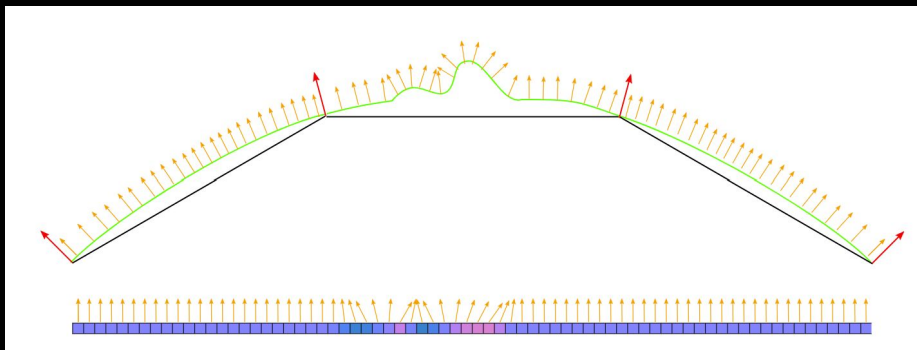
Point Light

Default Cookie

Colored Cookie



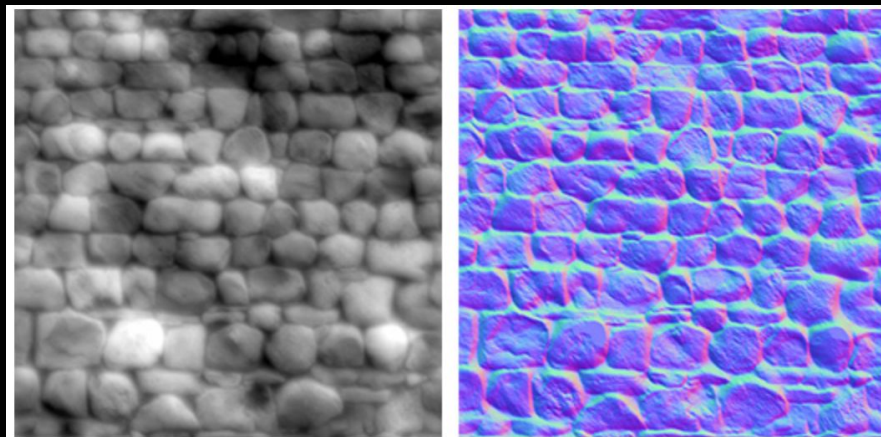
# Normal Map & Mask Texture



- The an image texture mapped to the surface of a model
- Uses a texture to store information about how to modify the surface normals across the model

-The RGB colour values are used to store the X,Y,Z direction of the vector

- Left: Height Map, Right: Normal Map



# Wall Demonstration

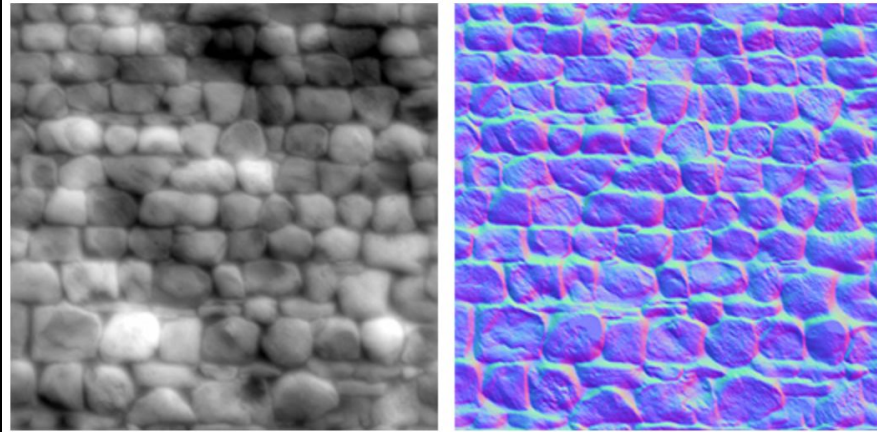
Top Pic: Wall without  
normal map effect



Bottom Pic: Wall with  
normal mapping applied



# Sprite Editor: Normal Map



- sprite editor

Change the texture to normal map

Being able to make normal maps with grey scale

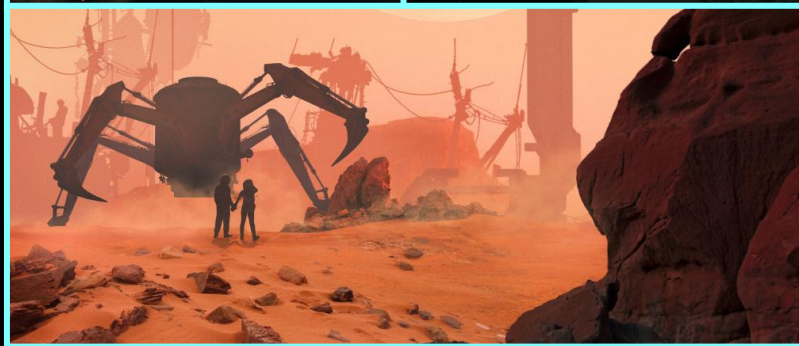
Sets to do it

- To evaluate this, you'll use the Frame Debugger window.

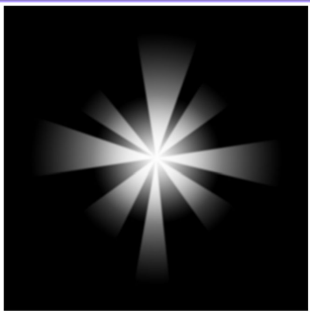
03

# 2D Lighting

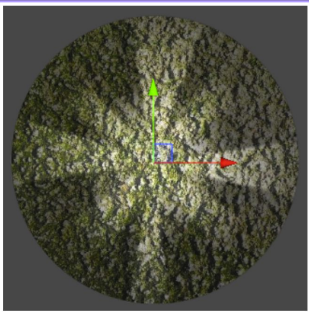
URP & Lit Materials



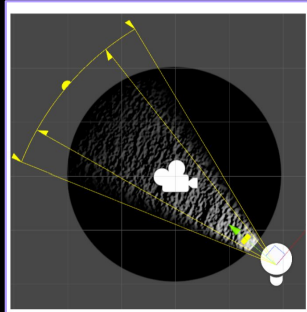
# Common Light Types



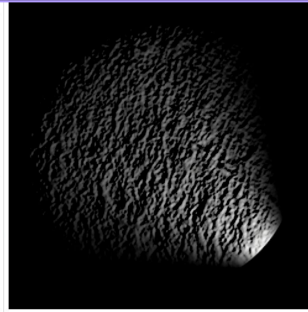
Selected Sprite



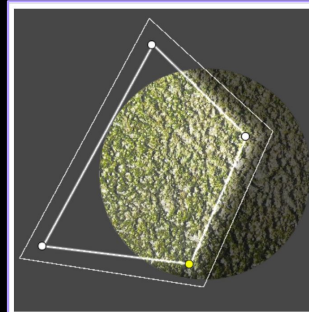
Resulting Light effect



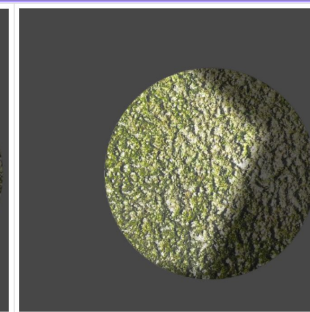
Point Light in Edit mode



Resulting Light effect



Freeform Light in edit mode



Resulting Light Effect

## Selected Sprite

Select a Sprite as the Light source

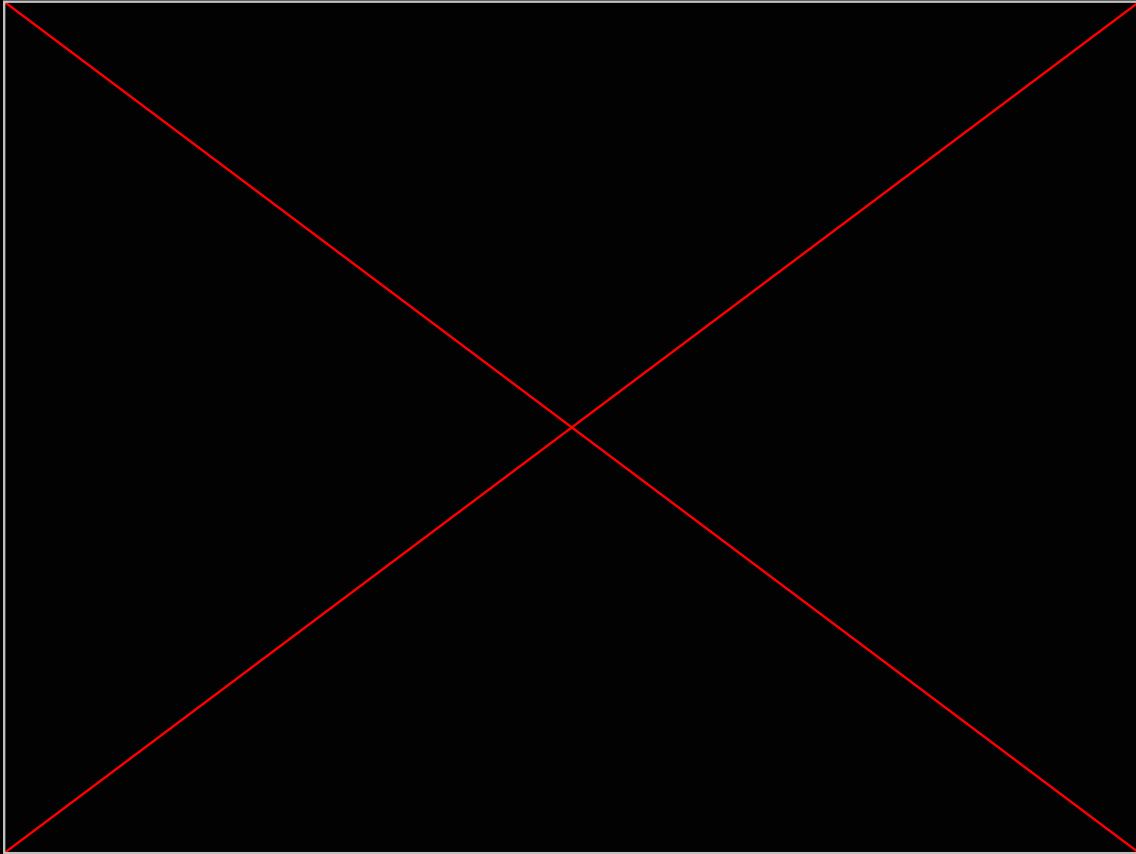
## Point Light

Great control over the angle and direction

## Freeform Light

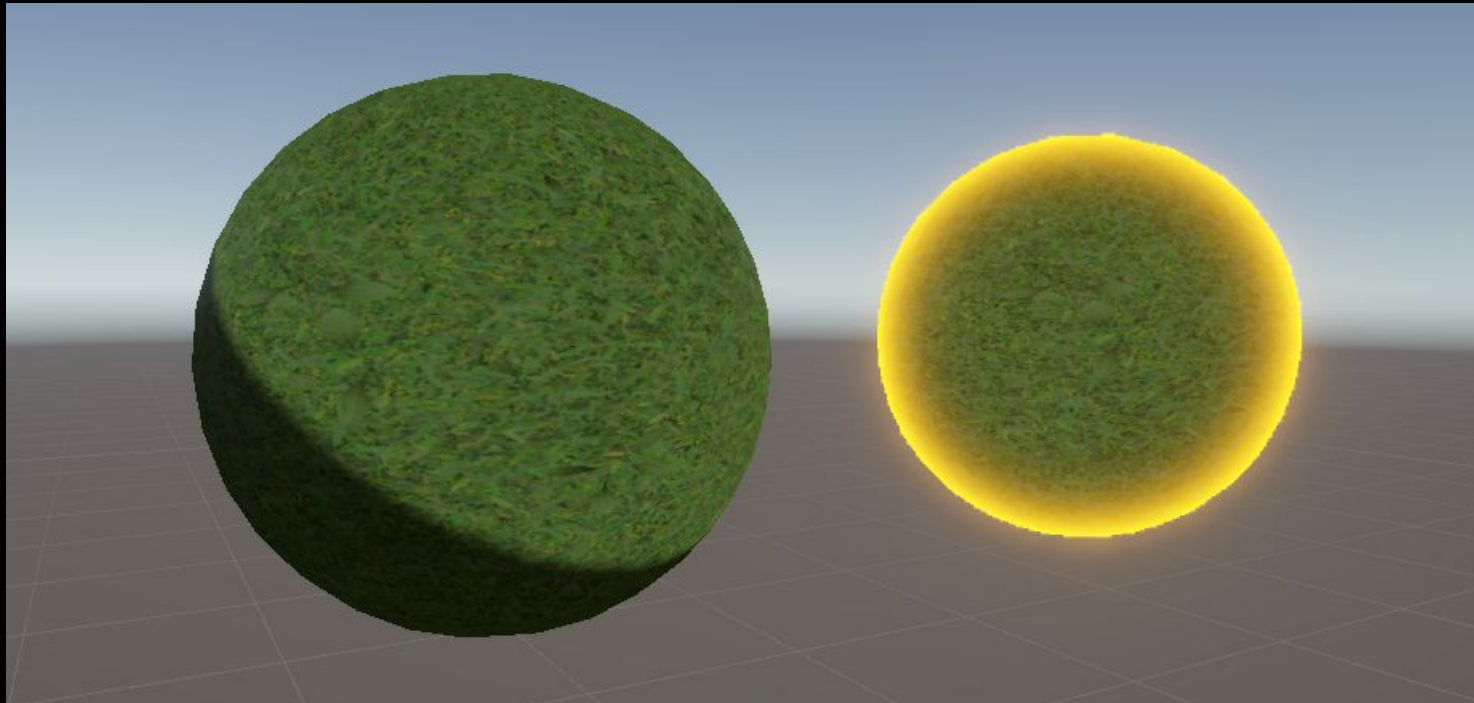
Create a Light from an editable polygon with a spline editor

# Global Lighting

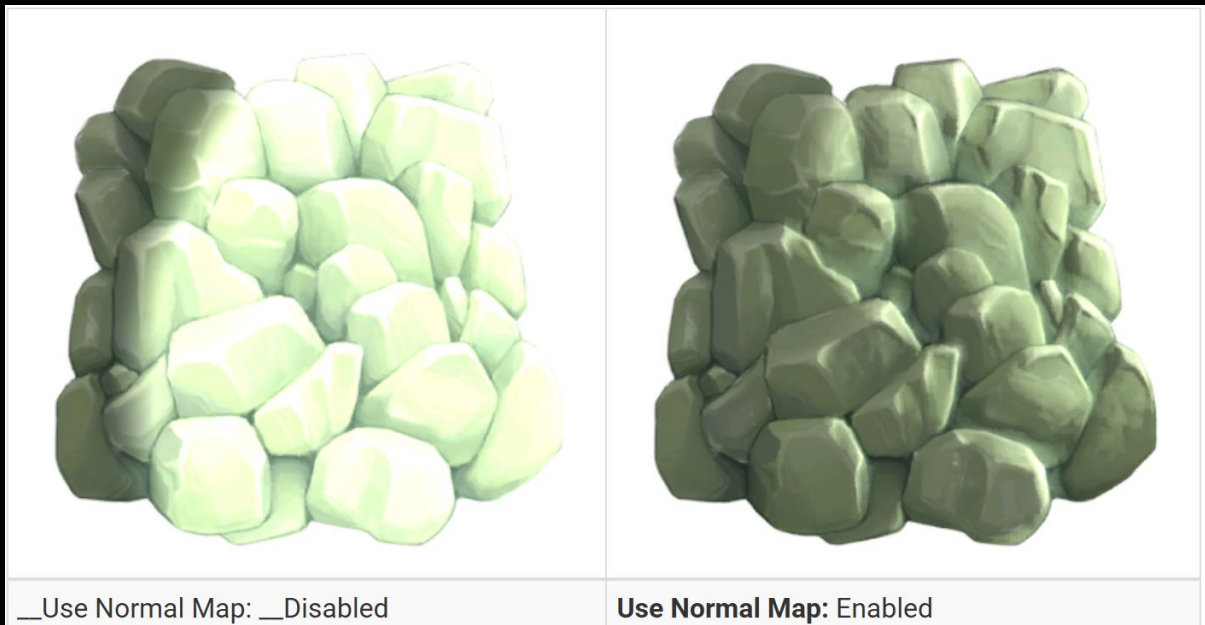


# Rim Lighting

outline with peripheral lighting - 2d fresnel effect



# 2D Example with Normal Map



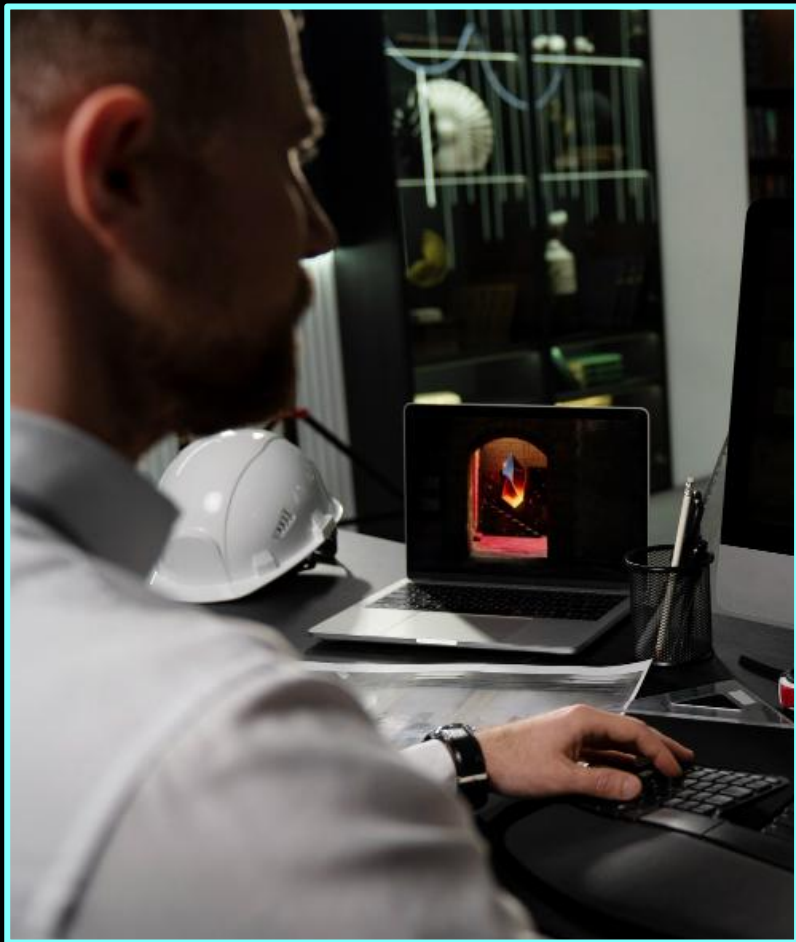
Refer to this documentation for more:

<https://docs.unity3d.com/Packages/com.unity.render-pipelines.universal@7.1/manual/2DLightProperties.html>

# Demo In Unity

Take object and make it into normal map in clip studio,  
Import into Unity  
Add a light , drag light around

[Bing Videos](#)



04

# Post Processing

More visual effects on top!



**Apply Full Screen Filter and Effects**

# Bloom



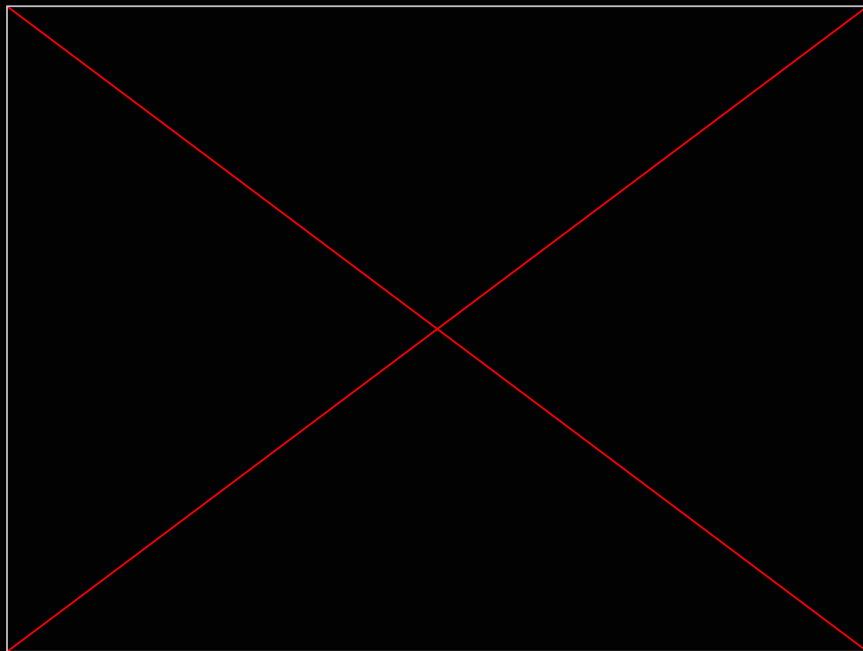
- A glow effect!
- simulate the visual artifact of extremely bright light overwhelming a camera lens

# Radial Blur



- A effect that gives a sense of speed!
- Lines in the direction of the camera direction

# Auto Exposure



- Sample all pixels from the screen to get average luminance factor and then change exposure and brightness
- Simulates the human eye when handling light

# Post Processing Volume

Post-process volumes allow you to add effect all or just a subsection of your Scene

## Post Processing

3.4.0 · December 18, 2023

From **Unity Registry** by Unity Technologies Inc.

*com.unity.postprocessing*

[Documentation](#) | [Changelog](#) | [Licenses](#)

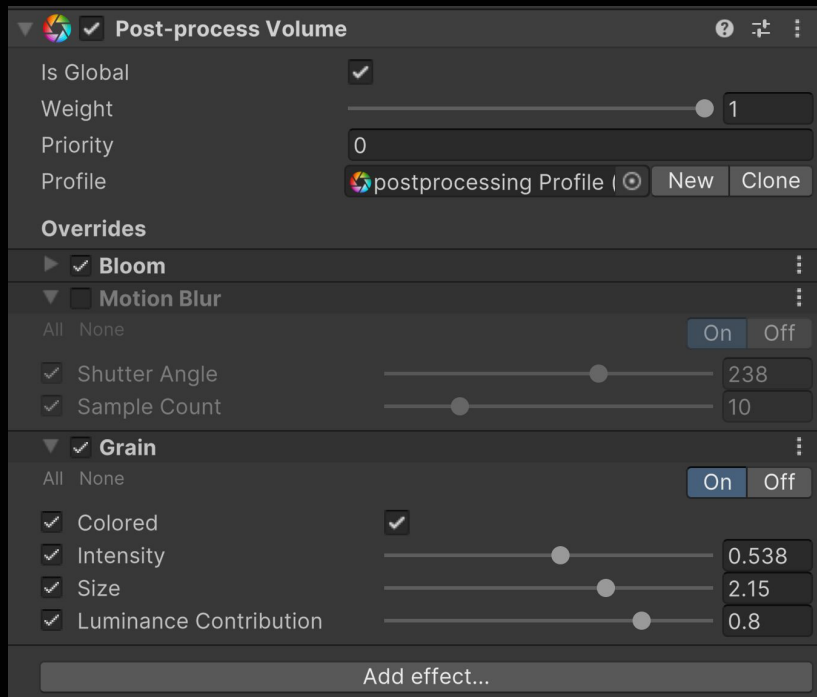
Remove

[Description](#) | [Version History](#) | [Dependencies](#)

The post-processing stack (v2) comes with a collection of effects and image filters you can apply to your cameras to improve the visuals of your games.

# Post Processing Volume

Post-process volumes allow you to add effect all or just a subsection of your Scene



[Bing Videos](#)

[Post Process Volumes - Unity Learn](#)

**Demo**