

Introduction

BEAM is a volumetric lighting & fog effect for Unity, targeting desktop and mid-high end mobile devices. Using froxel-based implementation, it maximizes the computation capability of your GPU cores, low VRAM usage while providing believable visual effect.

BEAM supports all kinds of Unity's realtime lights (directional, point, spot), with shadow and light cookies.

BEAM was built upon Renderer Feature & Volume framework so you can easily manage the graphics option for your game as well as setting up different moods for each region in the scene (outdoor light rays, indoor, basement fog, etc.)

Project requirements

- Unity 6.0 or higher.
- Universal Render Pipeline.
- Render Graph module enabled (compatibility mode is not supported).
- Graphics API: DX11, DX12, Vulkan.

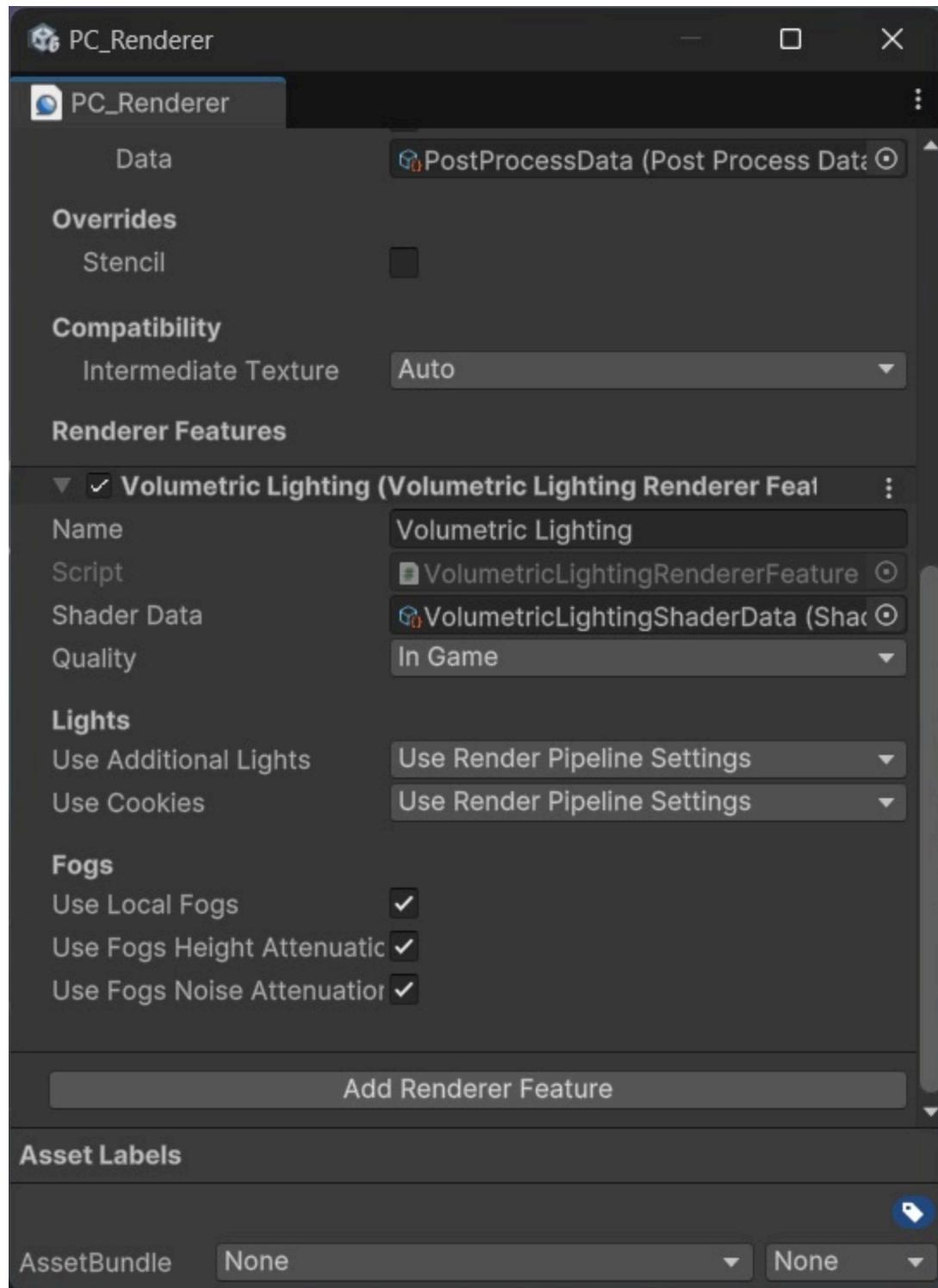
Build platforms

- Windows PC (DX11, DX12, Vulkan), high-end Android mobile (Vulkan).

Getting started

After importing BEAM into your project, you can quickly set it up with the following steps:

1. Select the default URP Renderer Data Asset (or the one in used), then add a Volumetric Lighting Renderer Feature to it:



2. Add a global volume (and volume profile) to your scene. (*optional*)
3. Add a Post Processing Custom/ Volumetric Lighting component to your volume profile.

4. Override all properties and increase the Intensity value. Although it's recommended to only override some properties, just override them all for now to explore the effect.

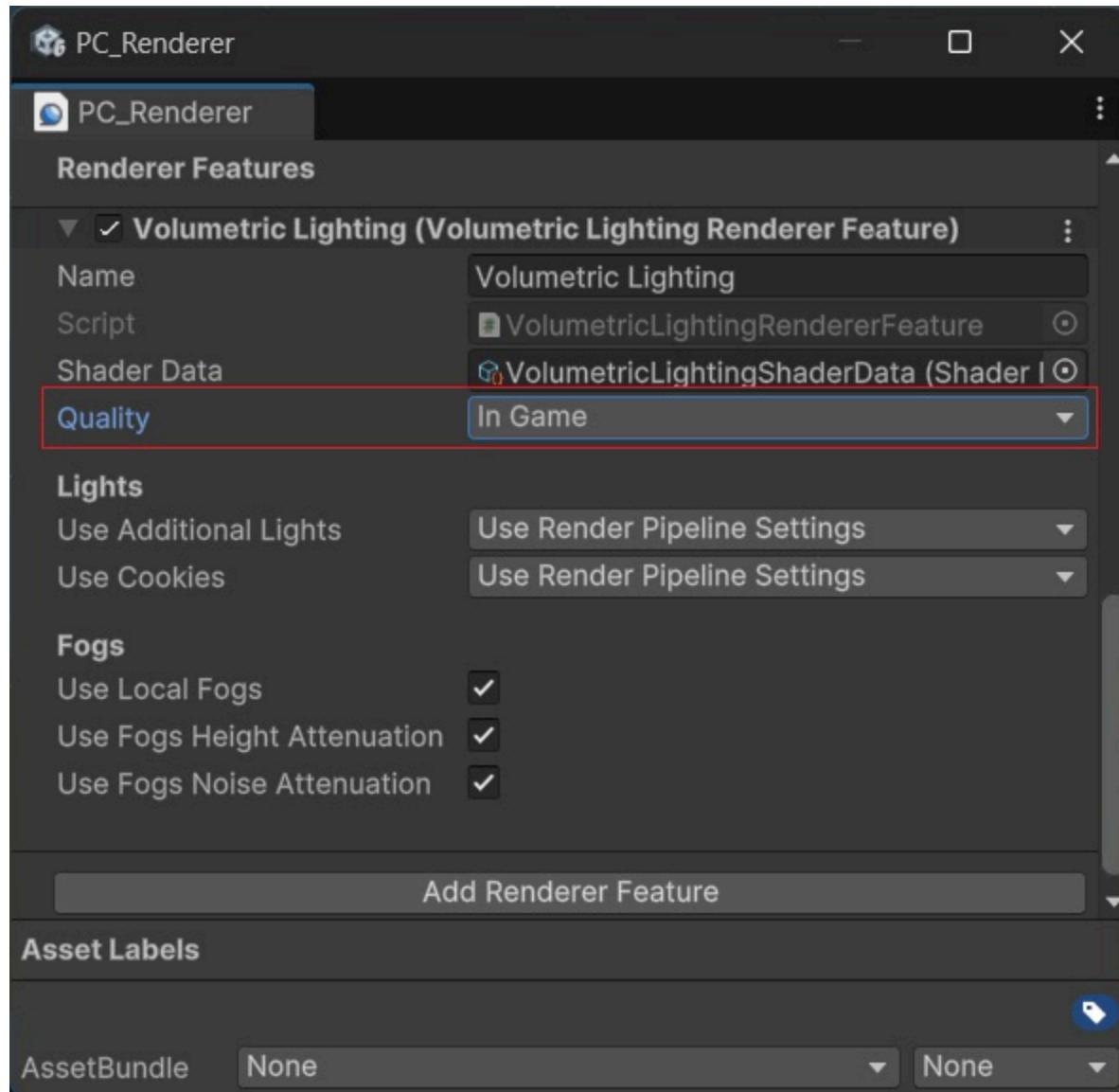


At this step you should see light rays while looking at the sun. Congratulations!



Quality levels

In your Volumetric Lighting Renderer Feature, you can choose the quality level for the effect corresponding to that URP Renderer Data Asset. This way you can add an in-game option for graphics quality by switching URP Renderer Data Asset.



The quality levels determine size of the Froxels Texture 3D, or in contrast, the number of pixels covered by 1 froxel in screen space (coverage).

Find details in this table:

	In Game Low	In Game	Cinematic	Cinematic High
Coverage	16	8	4	2
Tex3D size*	120x72x64	240x136x64	480x272x64	960x544x64

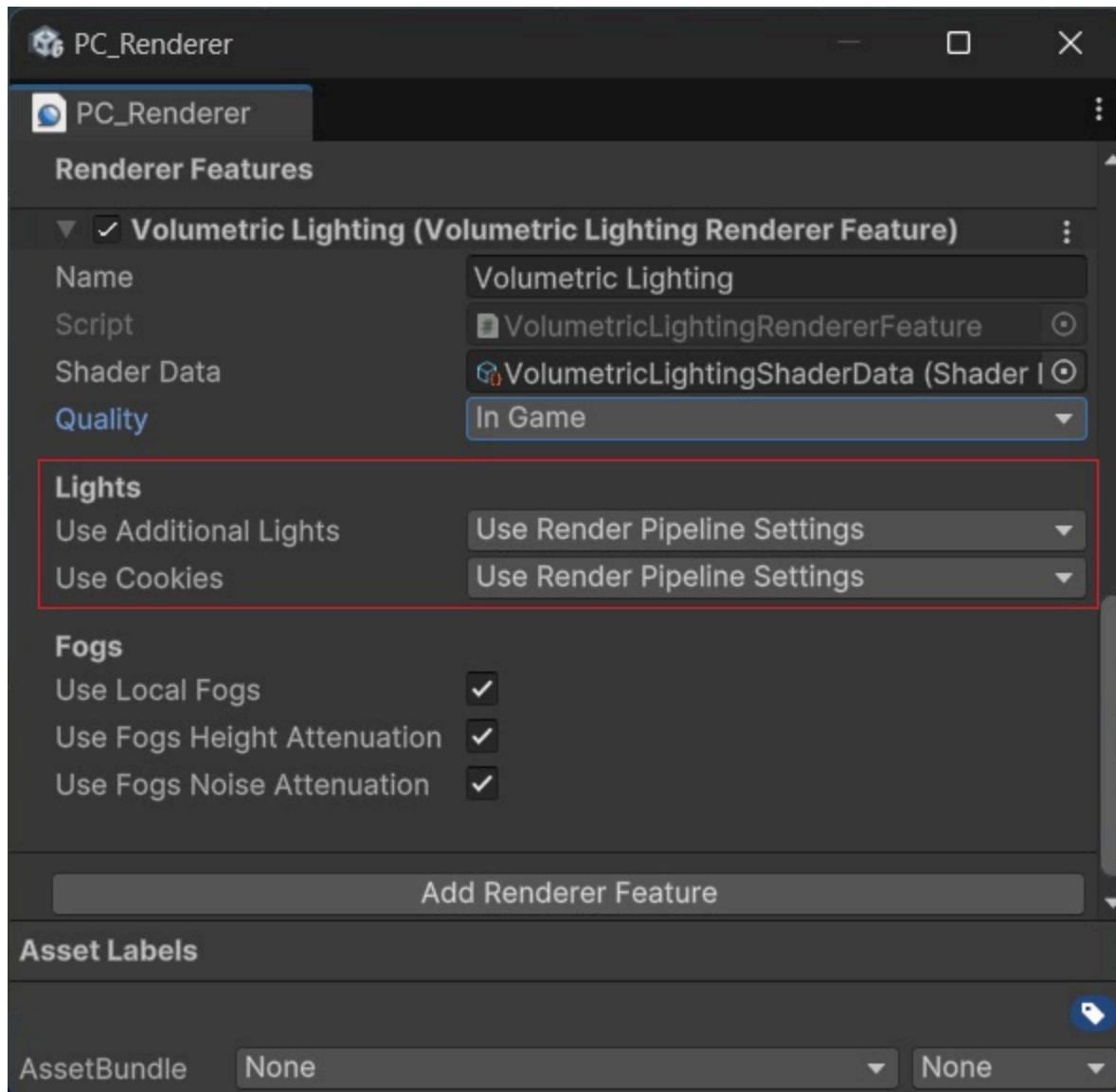
	In Game Low	In Game	Cinematic	Cinematic High
VRAM usage* (estimated, Tex3D only)	4 MB	16 MB	66 MB	267 MB
Use case	Gameplay with low settings	Gameplay with normal settings	Gameplay with high settings, pre-rendered videos, screenshots.	Pre-rendered videos, screenshots.

*Reference screen resolution: 1920x1080

Lights settings

BEAM uses the settings in your URP Asset for its calculation such as shadows, light limits, cookies, etc.

However, you can override unnecessarily settings in your renderer feature to improve performance.



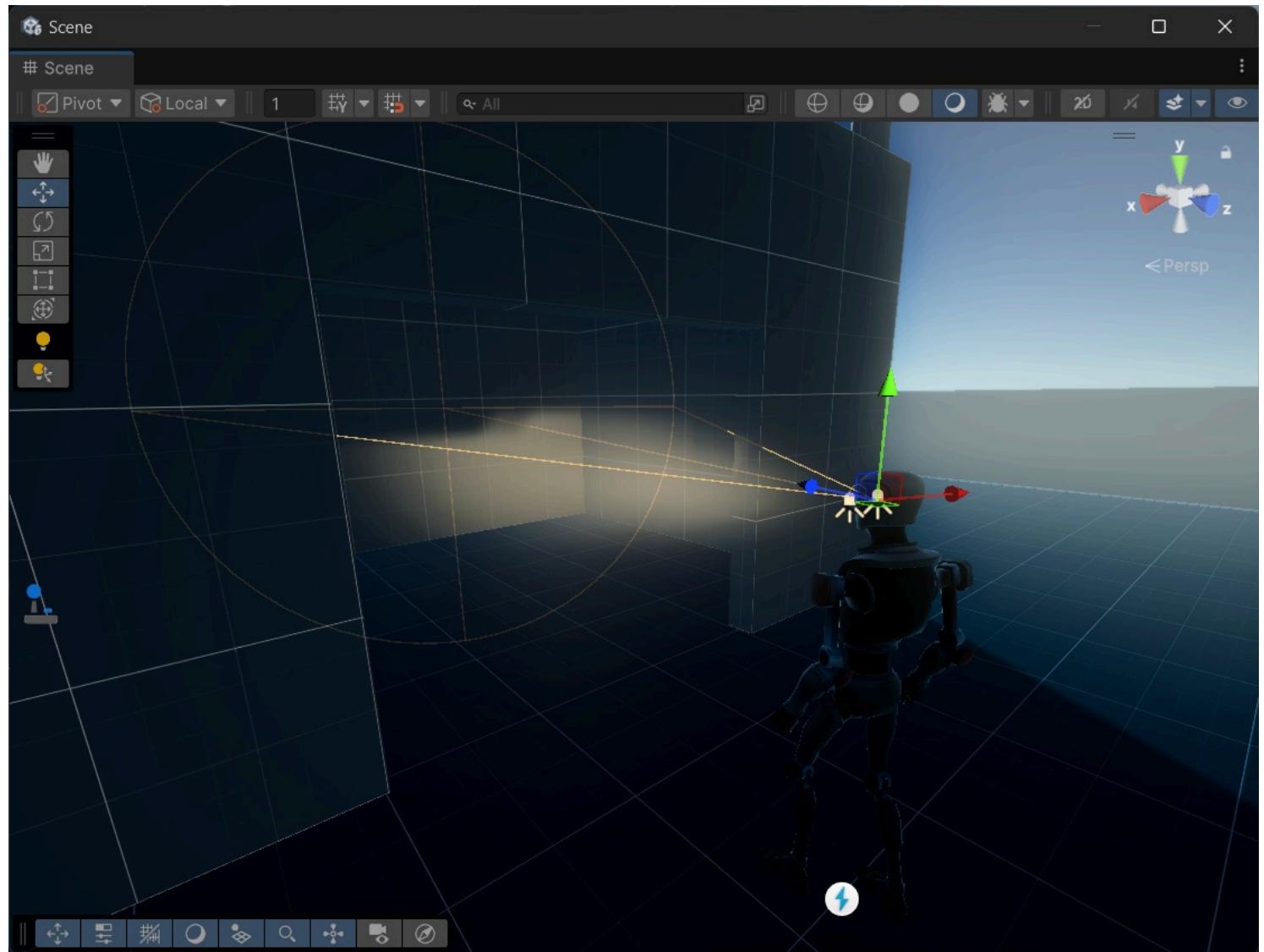
Where:

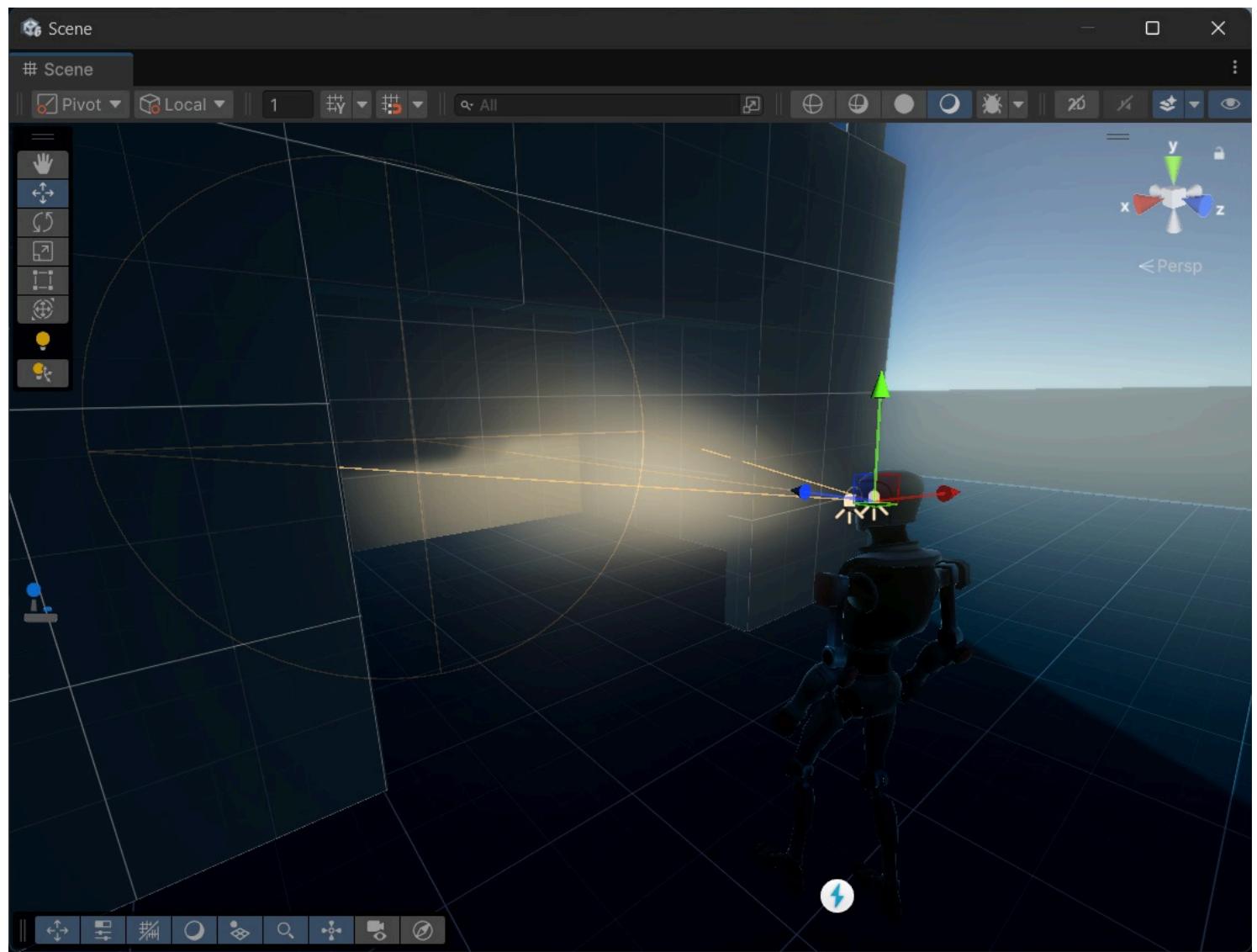
- **Use Additional Lights:** Should it calculate volumetric effect on additional (directional, point, spot) lights? This will calculate shadows too.
- **Uses Cookies:** Should it take cookies texture into account? If your cookie texture doesn't yield a distinctive visual (a flashlight with roundish cookies) then this can be off.

Possible values are:

- **Use Render Pipeline Settings:** This will not override anything. For example, if additional light shadows are on, it will render volumetric shadows for additional lights too.

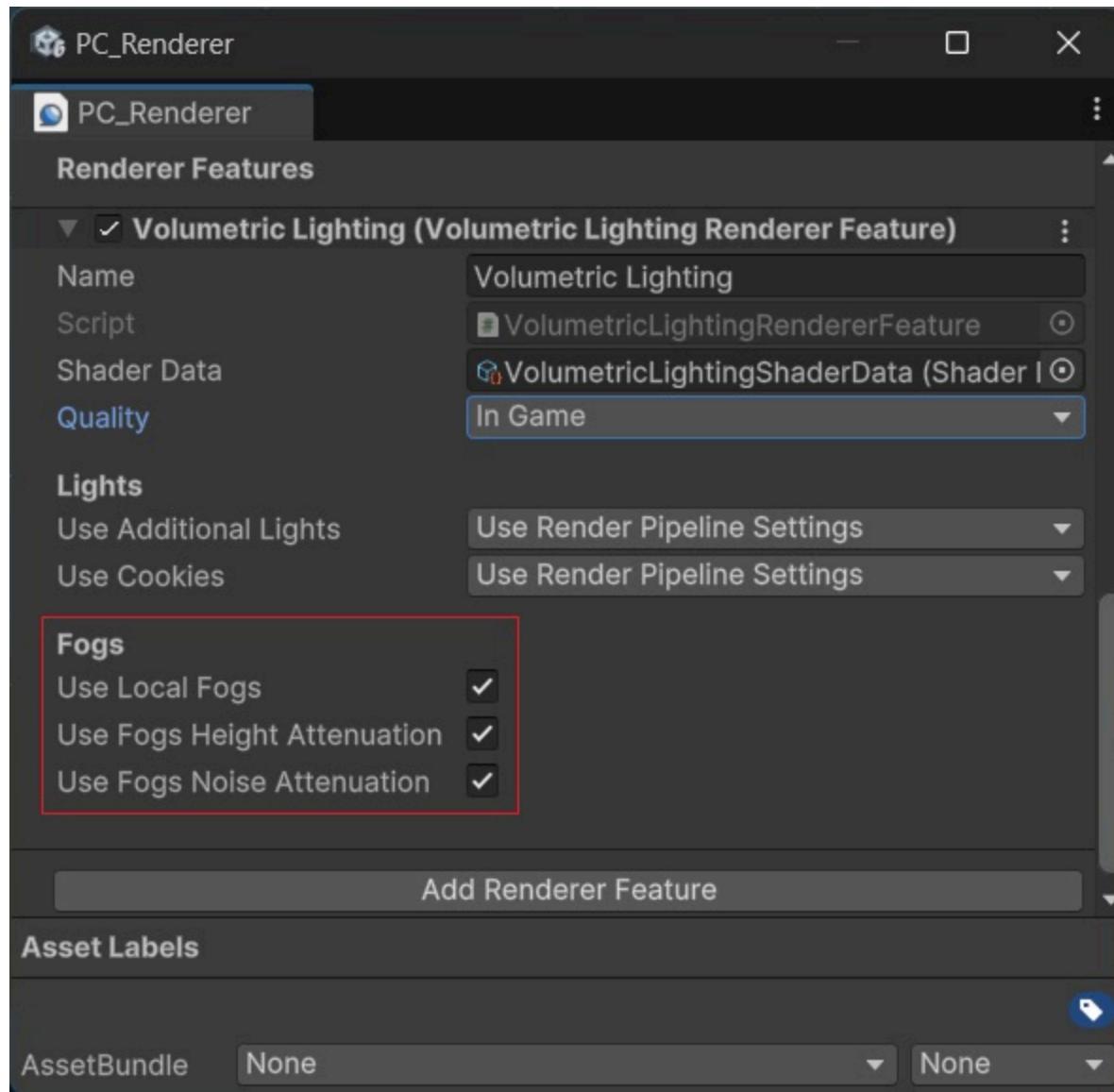
- **Off:** Force the settings to off, skip computing that effect.





Fogs settings

BEAM simulates the interaction between lights and fog particles in the air. In your renderer feature, you can change the settings for fogs:



Where:

- **Use Local Fogs:** Turn this on to manually place fog volumes at many places in the scene with different color and density. It will fallback to a global volume with white color if this is off.
- **Use Fogs Height Attenuation:** Exponentially fade out fog density as the world space height increases.
- **Use Fogs Noise Attenuation:** Add variant to fog density with animated noise.

More settings on height/noise attenuation can be found in your [Volume Override](#).

 **NOTE**

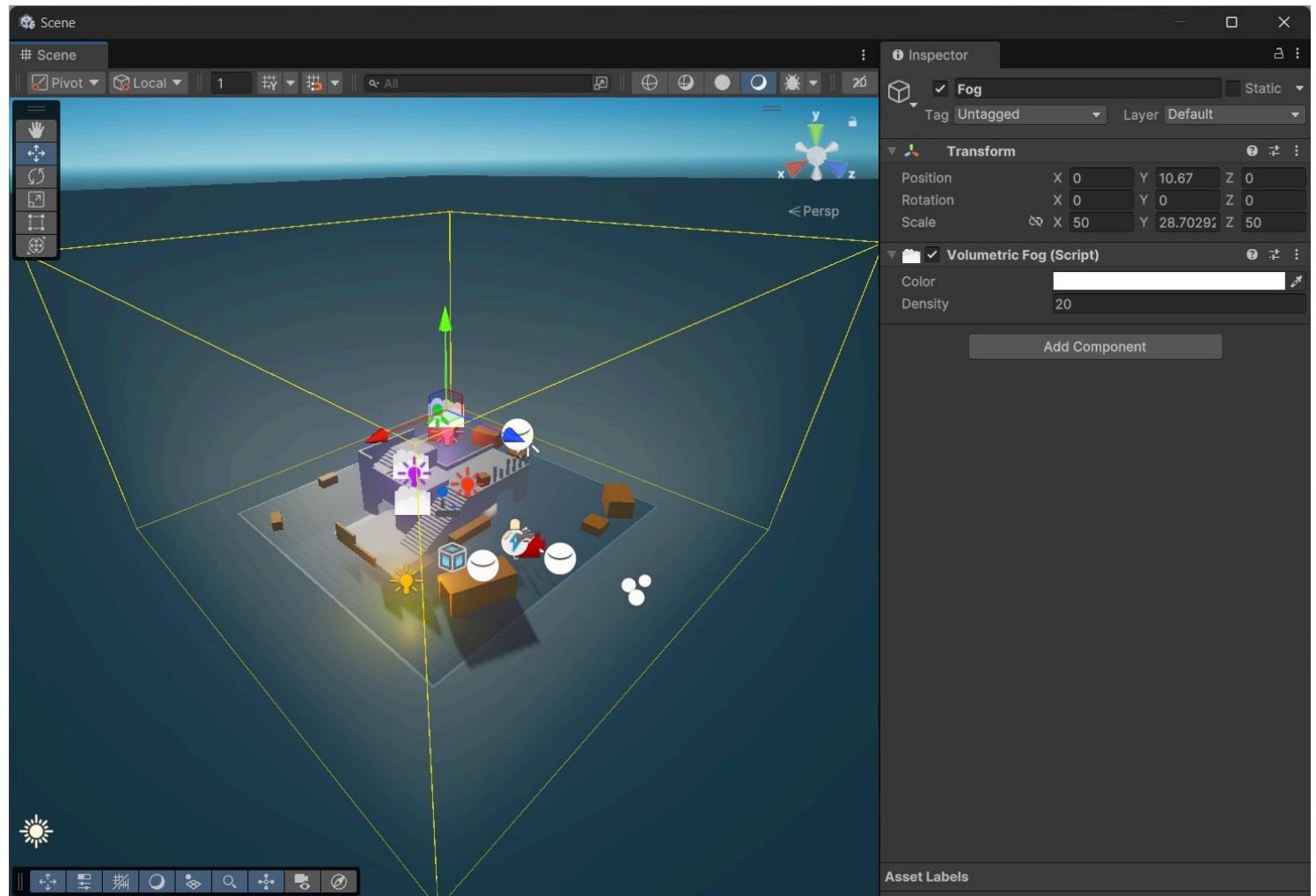
We limit the definition of "fog" to particles that scatter & emit light, in contrast to "smoke" which absorb light. That means fog volumes work best with bright colors. Setting their color to a dark value such as black will make them disappear.

Global and local fogs

By default, BEAM uses a global fog volume with infinite extents to cover the whole scene. This volume has the color of white and density of 1. But in some cases, you may want fogs to look different depending on player location, consider a scene with outdoor, indoor and basement area.

To enable local fogs, go to your renderer feature and turn on **Use Local Fog**.

To add a local fog volume, right click on the Hierarchy and select **Effects > BEAM - Local Fog Volume**.



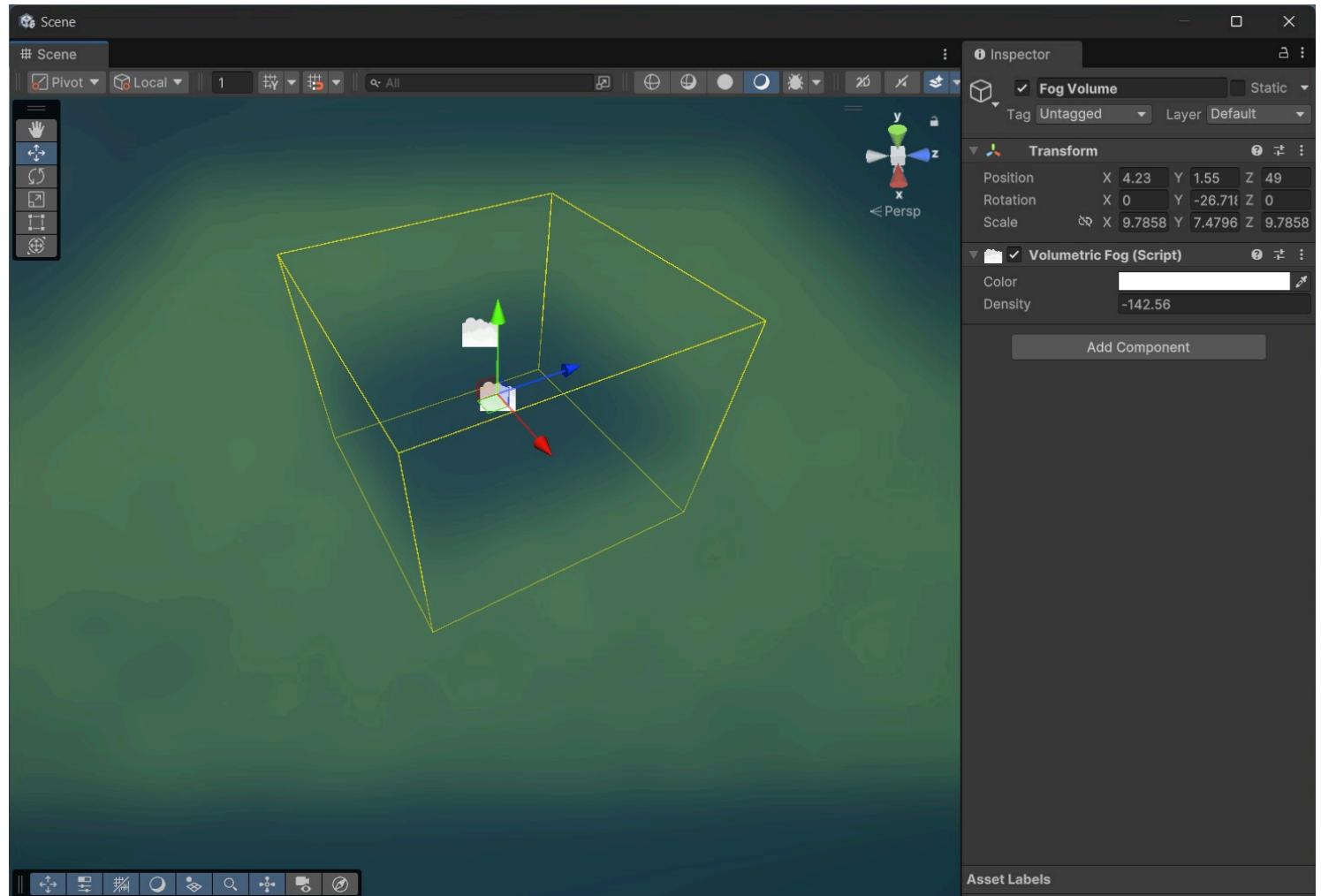
In the Inspector, you will see the following properties:

- **Color:** Tint color for the fog.
- **Density:** Density of the fog. Higher value makes the light effect stronger.

You can place as many fog volumes as you like. However, the maximum number of visible fog volume (seen by a camera) in a frame is 256.

Subtractive local fogs

Sometimes you want to exclude a region from fogs, e.g: an indoor area. To do that, create another fog volume and set its density volume to a negative value where fog is no longer visible.



i NOTE

This method only works when **Use Local Fog** is enabled in your renderer feature.

Volume Override

You can gradually change the effect settings by using Unity's Volume Framework. Select a volume and add a **Post Processing Custom/ Volumetric Lighting** component to it:



Light Scattering:

- **Intensity:** Overall strength of the effect. Too high might cause visual artifacts.
- **Anisotropic:** Define scatter angle of incoming light. Low value makes photons scatter more forward, while high value makes them scatter in all directions.
- **Max Distance:** Maximum distance from the camera to simulate light scattering effect. Low value will miss possibly visible light, while too high value might produce flickering artifacts.

Fog Height Attenuation:

- **Fog Min Height:** Minimum world space height where fog is visible.
- **Fog Max Height:** Maximum world space height where fog is visible.
- **Fog Height Attenuation Factor:** Contributes in fog attenuation based on world height.

Fog Noise Attenuation:

- **Fog Noise Frequency:** Frequency of the 3D noise used in fog attenuation.
- **Fog Noise Wind Direction:** Direction of the wind, contributes in fog animation.
- **Fog Noise Wind Strength:** Strength of fog wind, contributes in fog animation.

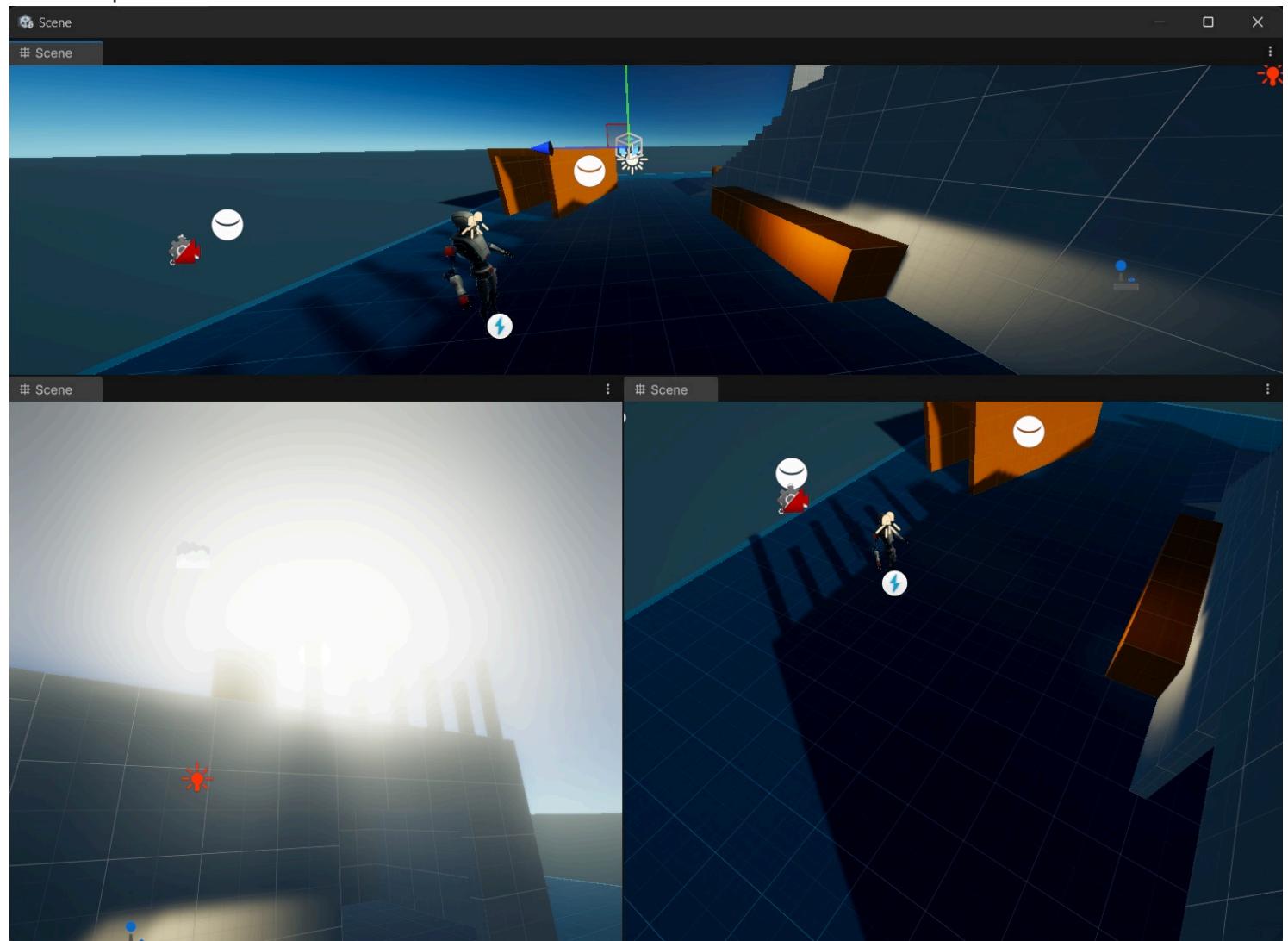
Filter:

- **Softness:** Make the volumetric light looks softer, reducing aliasing due to low resolution simulation.
- **Depth Bias:** Help remove banding/acne artifacts on the floor.

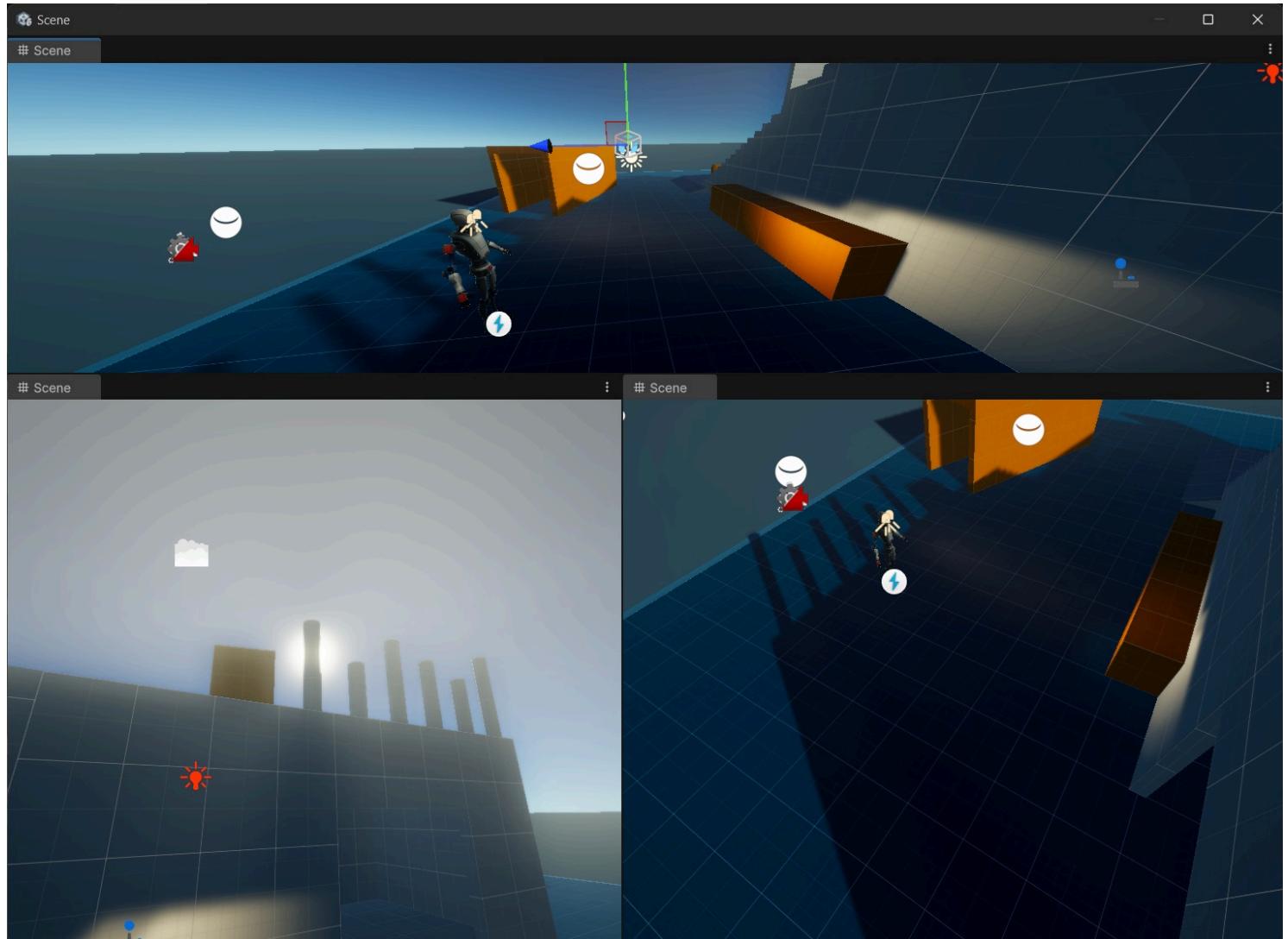
Anisotropic

BEAM uses the famous Henyey-Greenstein phase function (HG) to compute the scattering direction of incoming light. Anisotropic value plays a big role in this part, where low values make light scatter more forward (seen by looking directly at the light source) and high values make light scatter in all directions (seen from the opposite direction of the light source, think of a top down RPG camera).

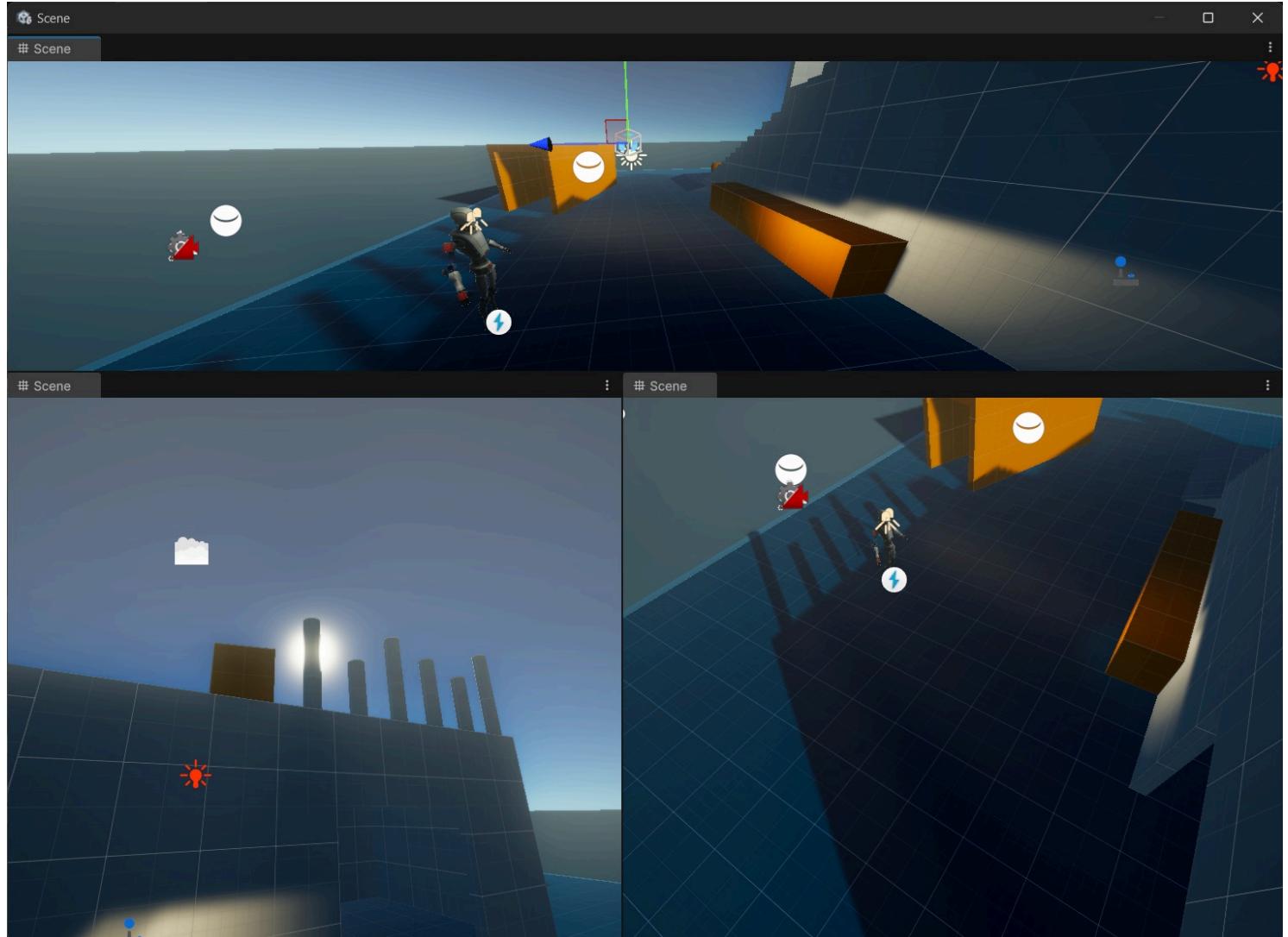
Anisotropic = 0.3



Anisotropic = 0.7



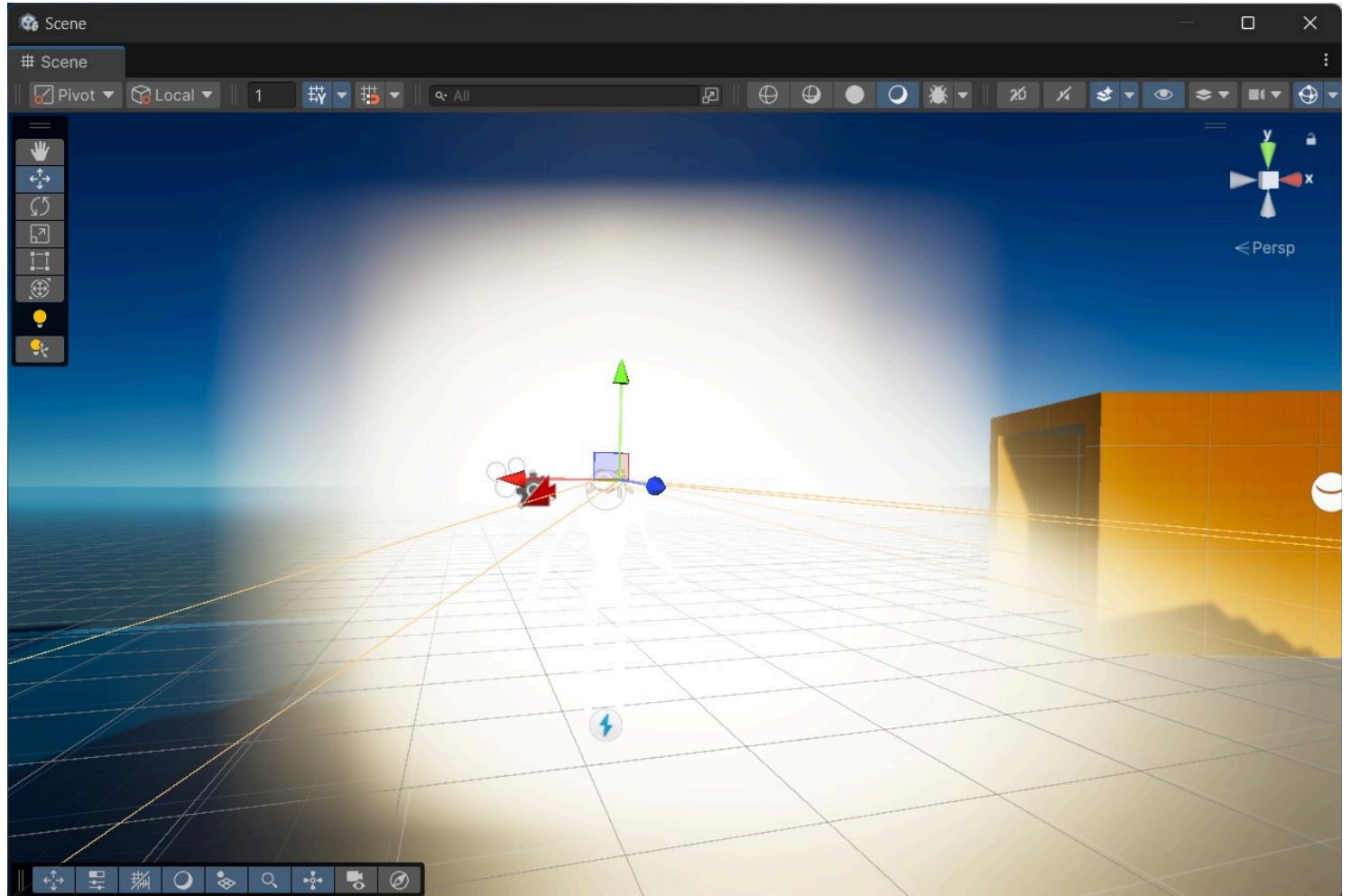
Anisotropic = 1.0



Visual artifacts

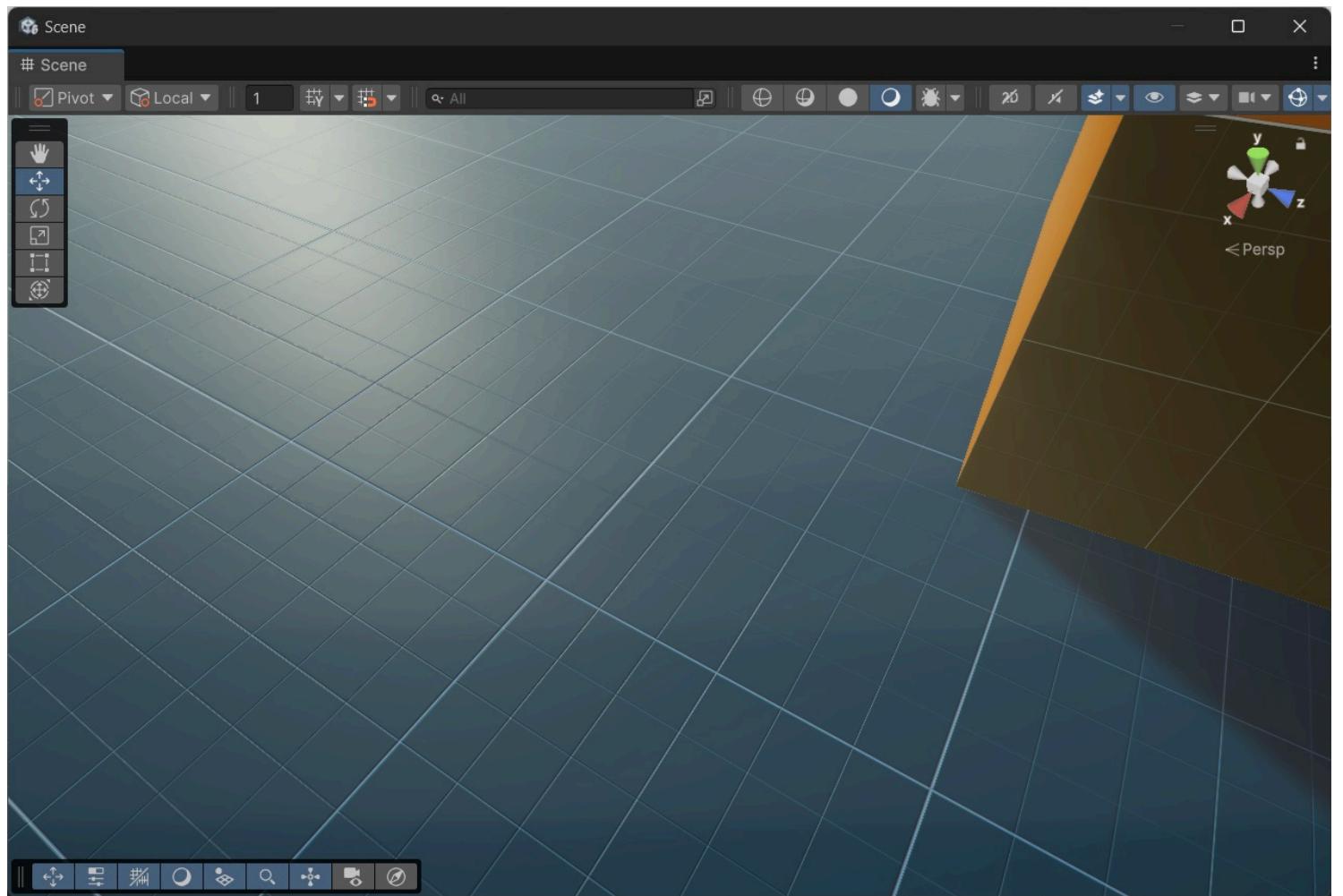
Below are some visual artifacts that can appear and how to fix them.

Blocky lights



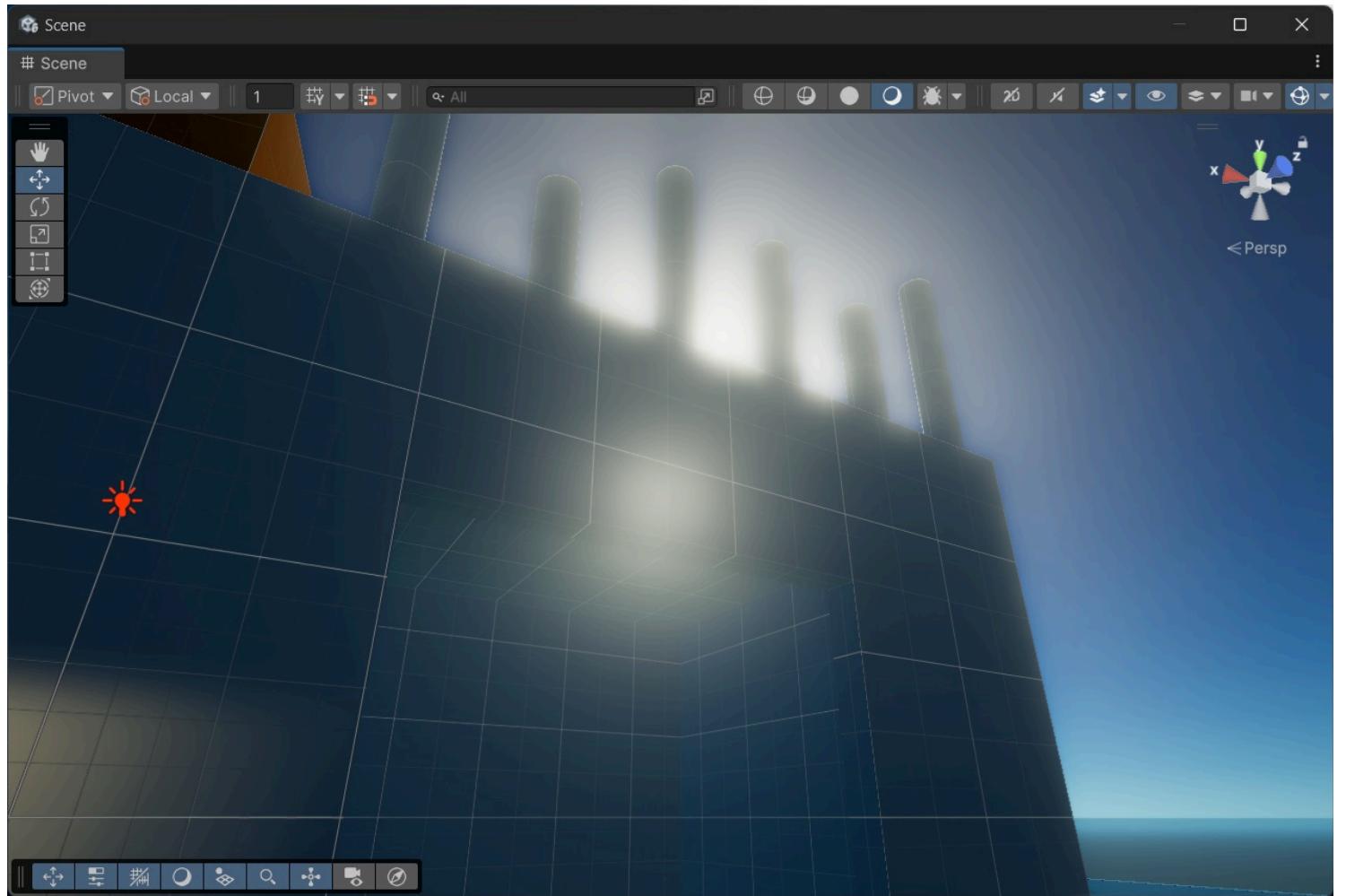
- Lower light intensity, local fog density, volume override intensity.
- Higher anisotropic & softness.

Light banding/acne



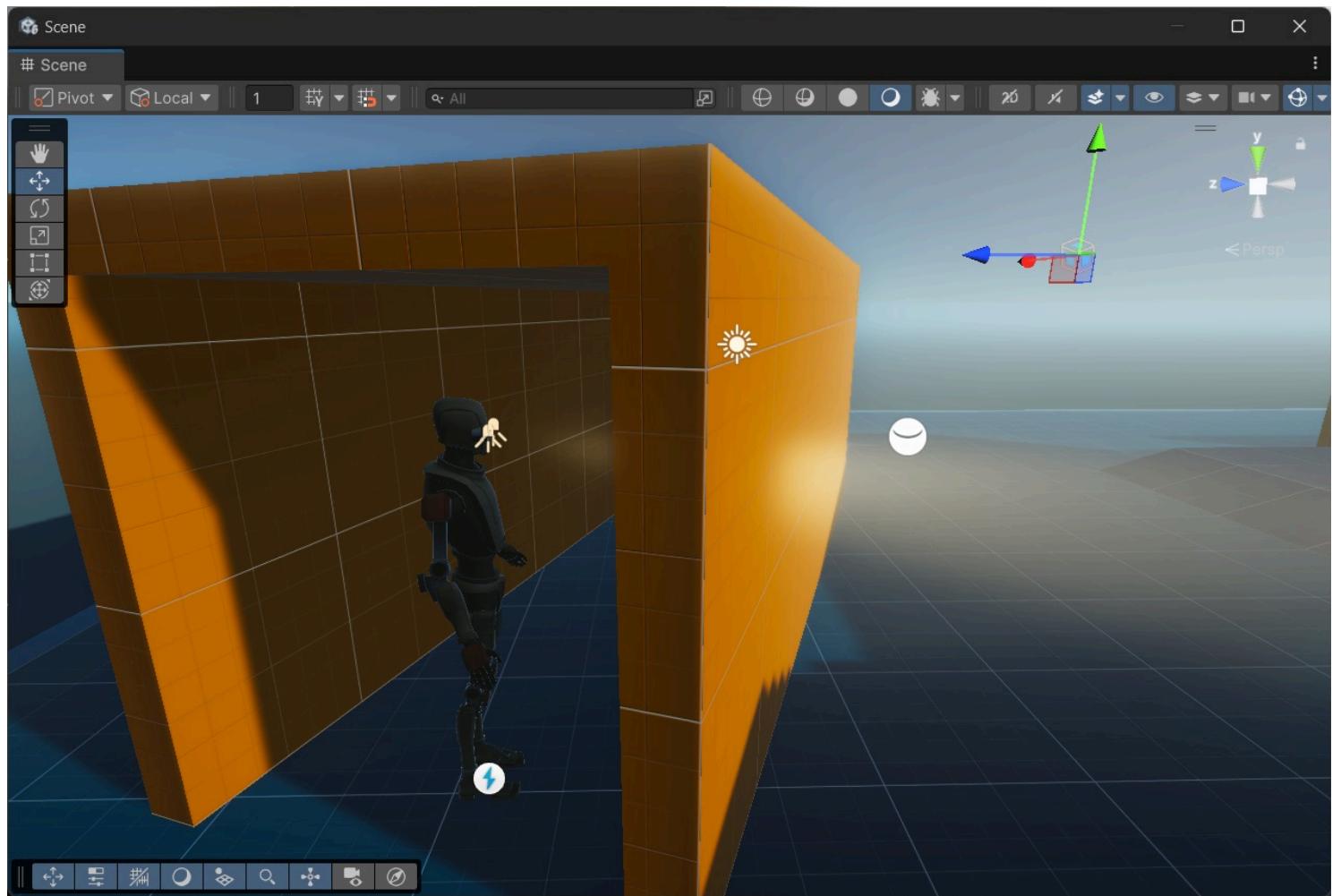
- Increase depth bias.

Sun can be seen through a wall



- Increase anisotropic.

Spot/Point lights leak through a wall



- Enable additional light shadows in URP settings.

Pixelated lights on tree leaves

With artifact

[Without artifact](#)



BEAM renders volumetric light on a canvas that is smaller than the screen resolution, so aliasing artifacts is somewhat inevitable but can be tone down so it is harder to spot.

- Reduce Alpha Cutoff value if your tree shader has one.
- Increase Softness in volume override.
- Decrease Intensity in volume override.
- Choose higher quality level in the renderer feature if affordable.