



✓URP

RADIANT GI

REALTIME GLOBAL ILLUMINATION



Contents

Introduction	3
Setup	3
Quick Start and Demo Scenes	5
Radiant GI: what's it how to use it in your project	6
General Settings	7
Quality section	8
Performance section	9
Artistic Controls section	9
Debug section	10
Virtual Emitters	11
Hints and known limitations	12
About the Fallbacks	13
Reuse Rays	13
Using Radiant GI with reflection probes	13
Using Reflective Shadow Maps	13
Selection Criteria and Performance	14
Fallback Recommendations	14

Introduction

Thank you for purchasing!

Radiant GI brings screen space, real-time, global illumination to Universal Rendering Pipeline.

We hope you find the asset easy and fun to use. Feel free to contact us for any enquiry.

Visit our Support Forum on <https://kronnect.com> for help and access to the latest beta releases.

Kronnect Technologies

Email: contact@kronnect.com

Support Forum: <https://www.kronnect.com/support>

Setup

Requisites

This package is designed for URP. It requires Unity 2021.3 or later.

Please make sure an URP asset is present in Project Settings/Quality and/or Project Settings/Graphics.

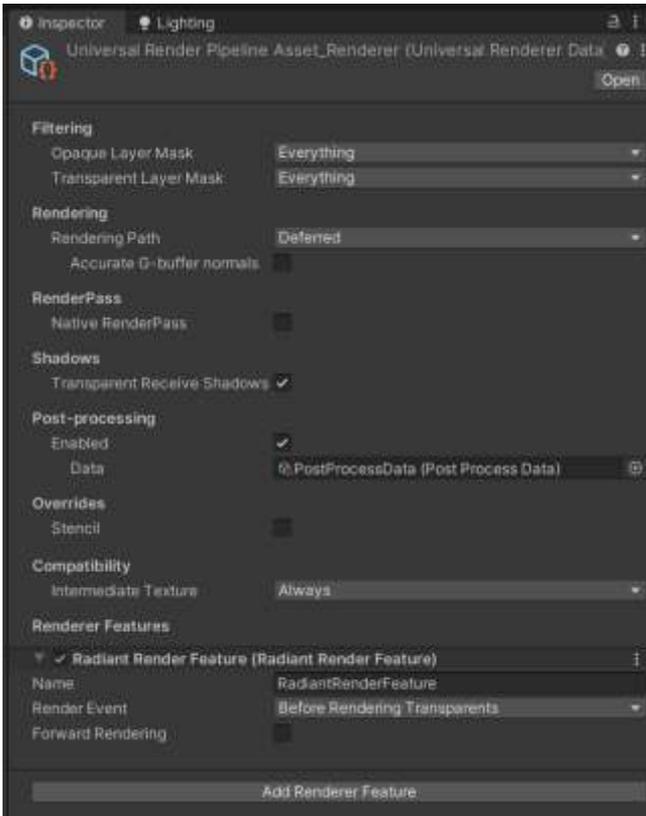
Configuration

Check out the introduction video here: https://youtu.be/sjcUA_nx5s4

To install the plugin correctly, you need to add the "Radiant Render Feature" to the URP asset:

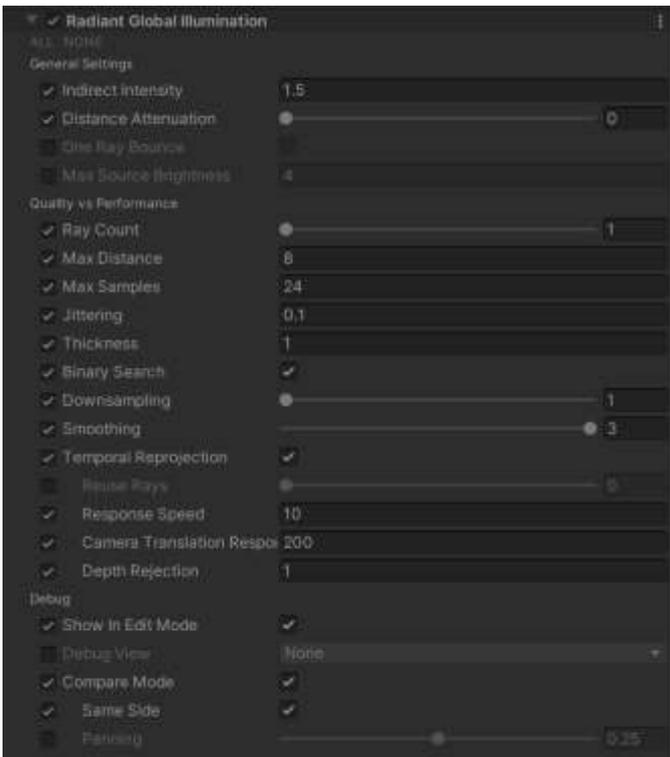
- 1) Double click the Universal Rendering Pipeline asset to select it (do this for all URP assets assigned in your Project Settings / Quality and Graphics).
- 2) Double click the Universal Renderer asset.
- 3) Click "+" to add the Radiant Render Feature to the list of the Forward Renderer Features.

Note: URP assets can be assigned to Project Settings / Graphics and Project Settings / Quality. Check both sections!



(You can also find a Universal Rendering Pipeline asset and ready to use in the RadiantGI / Demos / URP Settings folder. Make sure the Radiant Render Feature is listed in the Renderer Features of the Forward Renderer in the pipeline asset).

4) Next, add Radiant Global Illumination effect to any post processing volume in your scene and customize the settings (see next sections for more details).



Quick Start and Demo Scenes

Demo1: CornellBox Scene

This scene is intended to calibrate the effect. In this case, there's no lights in the scene, just a white emissive object on the ceiling of the room. When you run the scene, Radiant GI will propagate the light from this shiny/emissive object and illuminate the rest of the scene.

Select the "Radiant GI Volume" gameobject in the hierarchy to inspect and experiment with the settings.

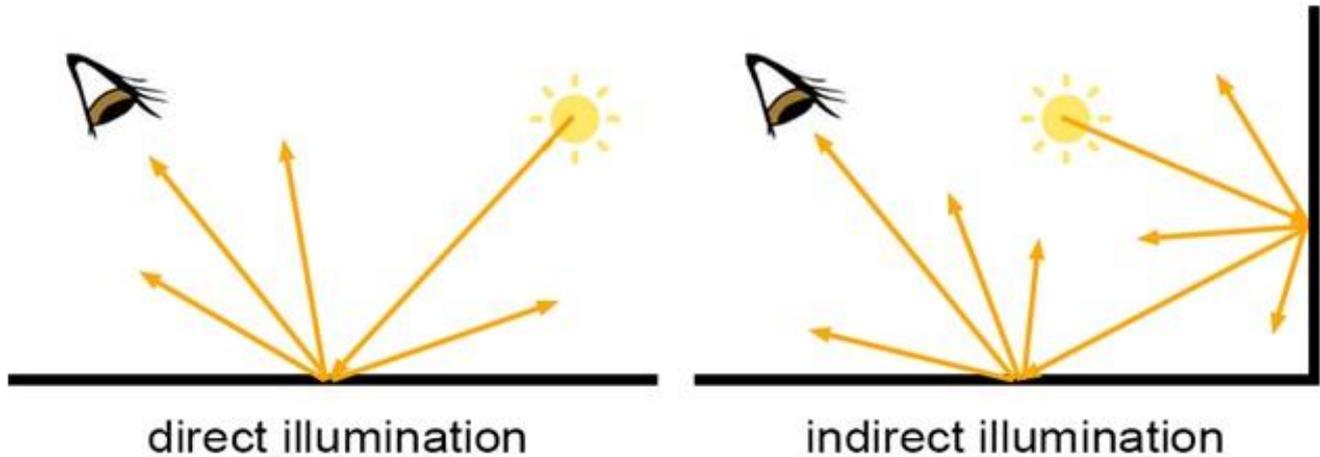
Demo2: Atrium

This demo scene represents the most complex challenge for a screen-space effect like this. In this scene, lot of emissive objects bounce in the scene while the camera moves constantly around.

Select the "Radiant GI Volume" gameobject in the hierarchy to modify the settings.

Radiant GI: what's it how to use it in your project

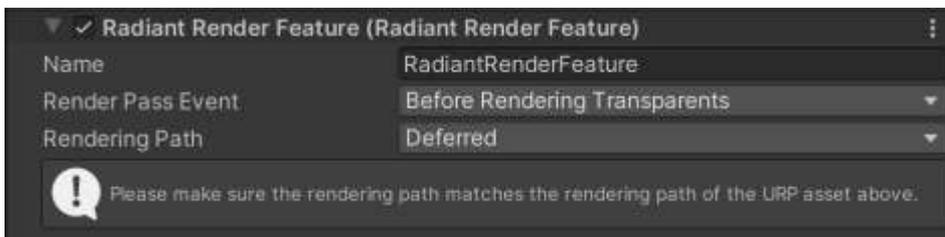
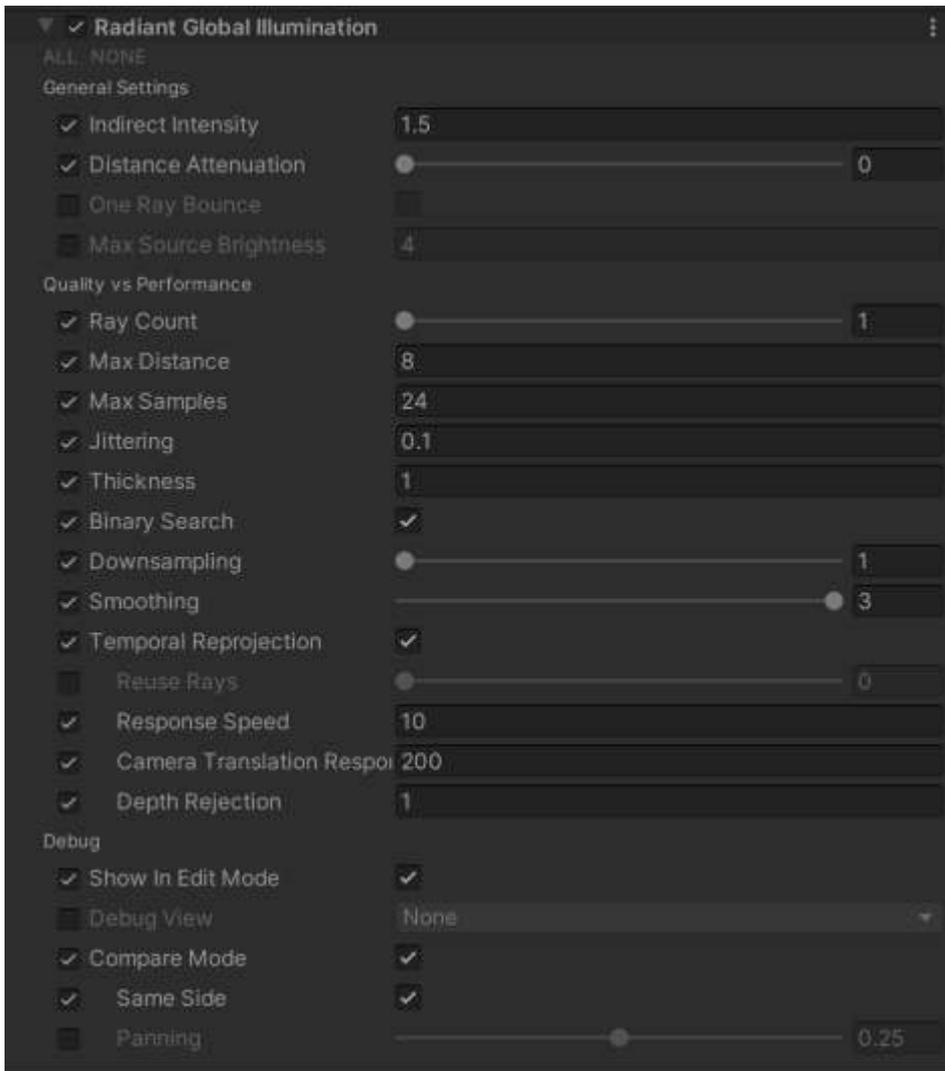
Global Illumination refers to the indirect / diffuse lighting caused by the bounces of the light on any surface. In Unity, all lights (directional, spot, point lights) are called “direct lighting” and they can illuminate surfaces that're directly reachable. Indirect lighting refers to the lighting that's reflected by those objects and that can also add some lighting to the scene.



Radiant GI computes and adds that indirect/diffuse lighting in real-time producing more natural/realistic lighting in your scenes. It's important to understand how to setup and configure the settings properly, as indirect/diffuse lighting can be subtle but convenient, to give more realism to your scene.

Radiant GI is a post-processing effect created for the Universal Rendering Pipeline. It can use the forward or deferred rendering path; however, it's strongly encouraged to use deferred rendering as it produces more realistic results. Important: make sure you set the same rendering path as in the URP asset!

The effect is designed as a render feature (check the previous setup section) and then, the settings are specified in a post-processing volume inside your scene. This means that you can use the effect as a global volume (affects the entire scene) or only in certain areas (adding the effect to specific post processing volumes):



The settings that can be configured in Radiant GI are:

General Settings

- **Indirect Intensity:** the global intensity of the effect. Please note that indirect lighting should be subtle and a complement to regular lighting as well as baked lighting. This means that you should not use a intensity greater than 2 for example. Good values are 0.3, 0.5 or 1 (this will depend on your scene).
- **Distance Attenuation:** light intensity reduces by the square of the distance. To give more artistic control, we added this setting, so you can specify the attenuation behaviour. A value of 0 means there's no attenuation. A value of 1 means the indirect intensity will be reduced by the square of the distance to the source light.
- **One Ray Bounce:** when enabled, Radiant will also use the indirect lighting computed in the previous frame, simulating an additional bounce of the light into the scene. This option is not expensive.

- **Max Source Brightness:** let you clamp or limit the brightness from the reflecting surfaces. Very bright spots like emissive pixels can add lot of noise. You can reduce the overall noise by setting a very low value to Max Source Brightness, like 0.2, and increasing the Indirect Intensity value. That will create a more uniform indirect contribution for the whole picture.
- **Normal Map Influence:** enhances lighting on surfaces with normal maps. Reduce this value to produce a foggier effect.
- **Luma Influence (only applies in forward rendering path):** since albedo data is not available in forward, this setting can add variety and improve results in this mode. Adjust this value to your liking. Set this value to 0 to disable luma influence.

Quality section

- **Ray Count:** by default, a single ray is computed per pixel in the screen. However, you choose to use up to 4 rays per pixel. Each additional ray will add considerable overhead to the processing. Usually, a single ray is enough to create a compelling effect. Only use 2 or more rays on higher-end GPU cards.
- **Max Distance:** the maximum distance where the indirect light can reach. For accuracy, try to reduce this value as much as possible (for example, make sure it just covers the room where the volume is).
- **Max Samples:** let you specify the maximum number of samples along the ray-march loop. This is a max number and intended to avoid very expensive loops. Values of 24 or 32 usually gives best results.
- **Jittering:** adds a bit of randomization to the ray-march step size to reduce banding. Useful when using low sample count.
- **Thickness:** the thickness or depth tolerance is used to determine when the ray hits a surface. This value represents the minimum distance of the ray-march position to a nearby surface to consider a hit. Values of 0.3 to 1 are usually the best.
- **Binary Search:** when enabled, the algorithm will perform a refinement of the hit position, producing more accurate results. This option will add a bit of overhead to the ray-march loop.
- **Smoothing:** specify the amount and intensity of blur to reduce noise. The higher this value, the more blur passes are used. A value of 4 will execute the denoiser twice, producing the smoothest illumination.
- **Temporal Reprojection:** uses velocity buffers and reprojection to accumulate hits over frames and produce a more pleasant/smooth result. It's recommended that this option is enabled.
- **Response Speed:** how fast the spatio-temporal filter reacts to changes in the screen. A lower value will produce softer results but can introduce ghosting.
- **Chroma Threshold:** determines the maximum color difference of the current image with regards to the history buffers. If the threshold is exceeded, the history buffer is clamped to reduce noise.

- **Camera Translation Response Speed:** when camera moves (changes position only) and to reduce smearing, this setting will reduce the effectivity of the spatio-temporal denoiser.
- **Fallback Modes:** what to do when a ray misses a target. By default, no fallback is enabled which means that ray won't add any GI data. To produce a smoother and accurate result, you can enable one or more fallbacks:
 - "Reuse Rays": the asset will check if the ray from previous frame succeeded and reuse it. A value of 0 disables this feature, while a greater value will increase the intensity of old ray hits. Please note that this option can introduce ghosting/smearing in moving objects so try to keep it low. This option is very performant.
 - **Depth Rejection:** when reusing rays, this value represents the maximum allowed depth difference between the history and current buffer. Default value is 1 meter. This check usually produces less disocclusion artifacts.
 - "Reflection Probes": the asset will use the info from nearby reflection probes that have the "Radiant Probe" script attached. Add this script to any probe that you want to be considered by Radiant GI.
 - "Reflective Shadow Map" is a technique that renders the scene from the directional light point of view. It's quite expensive if the directional light rotates. Check dedicated section for details.

Both Reflection Probes and Reflective Shadow Map provides useful fallbacks for light emitters that are not visible in front of the camera. If a reflection probe is used, the reflective shadow map option will be ignored as well as the reuse rays option. However, if the reflective shadow map doesn't provide light data due to shadows, the Reuse Rays can still be used.

Performance section

- **Raytracer Accuracy:** by reducing the raytracer accuracy, performance can be improved. Technically this option is effectively downscaling the input depth buffer to improve GPU cache efficiency.
- **Downsampling:** let you downscale the input image affecting all render passes, so the entire effect execution is faster (in exchange of quality of course).

Artistic Controls section

- **Brightness Threshold:** determines the minimum brightness value of a pixel to be considered as indirect source for lighting. Every pixel should cast indirect light (even if it's very dim) although you may want to limit this effect to certain bright pixels only using this setting.
- **Maximum Brightness:** clamps the maximum brightness of the resulting GI contribution. This setting let you avoid burning pixels with too much lighting.
- **Saturation:** adjusts the color saturation of the computed GI. A value of 0 makes the GI grayscale.

Debug section

- **Show In Edit More:** renders the effect while not in play mode. Currently, Temporal Reprojection is only available in play mode so the result can be quite noisy.
- **Debug View:** useful to inspect the different buffers used by the effect.
- **Compare Mode:** let you compare the impact of the effect and the current settings by showing a side-by-side or split screen view of your scene.

Virtual Emitters

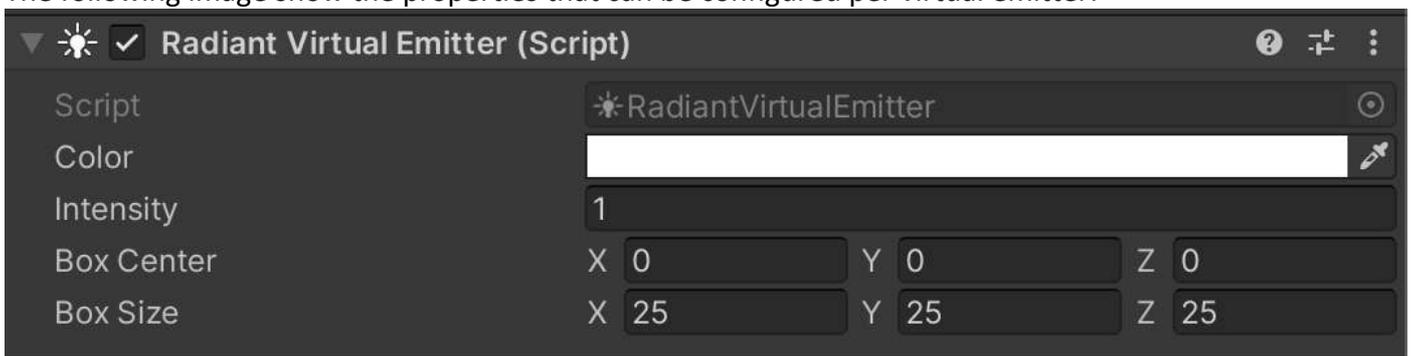
Virtual Emitters are GI injectors that can be placed anywhere in the scene. This feature must be enabled in the Radiant GI inspector:



To add a virtual emitter to your scene, select the top menu `GameObject -> Create Other -> Radiant GI -> Virtual Emitter`.

You can also add the `RadiantVirtualEmitter` script to any gameobject or character in the scene.

The following image show the properties that can be configured per virtual emitter:



Please note that light from virtual emitters can leak through walls. Use the Box Center / Box Size properties to limit the volume where the light of this virtual emitter can be used.

Hints and known limitations

- 1) Since Radiant GI is a screen-space effect, it cannot use emissive or lighting coming from objects that are not actually visible on the screen in an accurate way unless you enable some of the fallbacks (see dedicated section). When you move the camera around, you can see a fluctuation of the indirect illumination when bright objects leave or enter the camera frustum. This is by design in all techniques based on screen space effects. If this is a problem in your scene, enable one or more fallbacks or reduce the intensity of the effect or disable it completely in that area of your scene (use multiple post processing volumes to adjust it to the requirements of each area if needed).
- 2) Deferred rendering path is recommended. In this mode, Radiant will use the albedo and specular data from the g-buffers to produce a more accurate result. Forward rendering will still work but the indirect light cast on very dark objects will make them look “whitish”. Use the Luma Influence setting in forward rendering mode to boost dark pixel colors.
- 3) Performance: a balanced configuration includes the following settings (like the Atrium demo scene):
 - a. Indirect Intensity = 1.5.
 - b. Ray Count = 1.
 - c. Max Distance = 8.
 - d. Max Samples = 24 (or lower, then increase jittering to 0.5 or 1).
 - e. Thickness = 1.
 - f. Binary Search enabled.
 - g. Downsampling = 1.
 - h. Raytracer Accuracy = 8.
 - i. Smoothing = 3 or 4.
 - j. Temporal Reprojection = enabled.
 - k. Reuse Rays = 1.
 - l. Response Speed = 10
 - m. Camera Translation Response = 200
 - n. Depth Rejection = 1
 - o. Chroma Threshold = 0.25

About the Fallbacks

Radiant GI is a screen space effect. It provides instant global illumination, but it's limited to the screen content. Non-visible objects (i.e. objects behind the camera or out of screen) can't illuminate other objects in the scene. On the other hand, Radiant is 100% GPU based, requires no pre-computation, works with any kind of 3D scene including dynamic/procedural content and any surface or light, and it's very easy to use.

To complement light data from off-screen objects, we've provided "fallbacks" methods. These fallbacks can be enabled in the Radiant GI inspector and are described here.

You can enable any number of fallbacks. Some are more expensive than others and the system will automatically pick the best one automatically.

Reuse Rays

This technique is very cheap and uses data from previous frames. When a ray misses a target during the main Radiant raytrace pass, it can look for rays from previous frames that were emitted from the same positions and reuse them. The result is a smoother GI composition.

The downside of this option is that it could add some ghosting or trails to fast moving objects so you may want to use a lower Intensity value for this fallback looking for a satisfying balance. Also, it's still limited to in-screen surfaces.

Using Radiant GI with reflection probes

When using the reflection probes option in the Fallbacks section, Radiant GI will automatically use the cubemap rendered by nearby reflection probes when rays miss their target in screen space. Although this can't be 100% accurate, it will improve the effect.

To use reflection probes with Radiant, add the "Radiant Probe" script to any number of reflection probes. Radiant will automatically use the 2 nearest reflection probes to the camera. Of course, this technique requires that you use reflection probes in the scene.

You can add a reflection probe as a child to the camera and make it real-time probe to provide the most accurate result possible although this will be quite expensive. Or just add reflection probes to the center of rooms with bake mode set to Awake.

Radiant GI will use the importance and intensity values of each reflection probes as well as the blending distance and boundaries defined by box size to determine the weight of the sampling of the probe cubemap texture.

Using Reflective Shadow Maps

This feature can be enabled in the Fallback section of Radiant GI inspector. When you enable this option, make sure the script "Radiant Shadow Map" is added to the main directional light.

Reflective Shadow Maps is a technique used to compute global illumination for certain lights. It works like shadow maps, but instead of capturing only the depth map from the light point of view, it also captures the emitted light (also called flux), the normals and world positions of those pixels. Using this data, the

technique computes the bounces of the light onto nearby pixels or surfaces generating a GI map which is later used during the main step of Radiant GI.

The benefit of this technique is that it can supply GI data to Radiant GI for objects that are not within the camera frustum.

For optimization purposes, Radiant GI only computes the buffers (flux, normal, world positions) of the reflective shadow maps when the directional light rotates or when the camera position changes at least 5 meters.

Selection Criteria and Performance

If reflection probes are available, the system will always use their data. If they're not available, it will try to use data from the reflective shadow maps. And finally, if no reflection probes or reflective shadow map data is available, it will use the "Reuse Rays" fallback (only if these fallbacks are enabled of course, you can decide which fallbacks can use the system).

- Reuse Rays: very fast.
- Reflection Probes: medium.
- Reflective Shadow Maps: expensive when it's updated, but medium normally (it's updated only when directional light rotates or camera moves a certain distance so while not updating,

Fallback Recommendations

Use fallbacks only if lighting changes are distracting or affecting your game experience. Please remember that Radiant GI or global illumination in general, can be used along other lighting effects. This means that Radiant GI should be used wisely, to make your scene better looking by adding a subtle extra lighting to it.

Regarding fallbacks, our recommendations are:

- 1) Since "Reuse Rays" fallback is very cheap, you should use it always, unless ghosting is an issue.
- 2) Using Reflection Probes with at least one probe at the center of a room (or several if the room is big) with Bake mode = Awake at least, will provide light data from off-screen objects.
- 3) If reflection probes can't be used, reflective shadow maps can improve the results. In outdoors or large open areas for example, they can be a better alternative to reflection probes if the Sun is high in the sky (so the scene capture is as wide as possible – remember the RSM capture image data from the directional light point of view to account for shadows).