Runtime GI Probes

Atler Gibby

Table of Contents

1. Requirements

Runtime GI Probes was originally made on Unity 2021.3.8f1, and on the Universal Render Pipeline 12.1.7. If there are issues with running on slightly older versions of Unity or the URP, first check to see if any errors were made during setup before deciding on upgrading your Unity project.

Runtime GI Probes does not replace real-time lighting, meaning it will not create shadows like Unity's built-in light mapping solution. Also, URP's SSAO or Deferred Rendering needs to be enabled to create the "_CameraNormalsTexture" which is necessary for *Runtime GI Probes* to know world space normal information. Therefore, it is not recommended for mobile or Quest 2 standalone since real-time lighting combined with SSAO or Deferred Rendering may be too expensive in terms of performance for the average game running on those platforms. *Runtime GI Probes* is better suited for gaming PCs and modern gaming consoles. Although, it is possible to run on modern mobile devices that support Vulkan at lower settings.

2. Basic Setup

Before you can run the demo scene, the most basic steps are:

1. Go to **Edit > Project Settings > Player > Other Settings > Color Space** and select Linear. Gamma will make the demo scene overly bright.
2. Go to **Edit > Project Settings > Graphics > Scriptable Render Pipeline Settings** and select *GIProbesRPAsset*.
3. Go to **Edit > Project Settings > Quality > Render Pipeline Asset** and select *GIProbesRPAsset* for the current active quality level (If set to none, Unity will default to the render pipeline asset used in the graphics settings).

This will activate the *GIProbesRPAsset*, which is preconfigured with the *GI Render Feature*. At this point global illumination should be visible in the demo scene. The *GIProbesRPAsset* is set up with Screen Space Ambient Occlusion and Screen Space Shadows. To improve performance, you can lower the quality of SSAO. SSAO must use Depth Normals and not Depth, or you can set the rendering mode of the *GIProbesRPAsset* to Deferred instead of Forward and completely disable the SSAO Render Feature.

3. Advanced Setup

Assuming the basic setups steps have been completed and you wish to add *Runtime GI Probes* to a new or existing scene, the steps are:

1. Go to **RuntimeGIProbes > Prefabs** and drag and drop the *GILightBaker* into your scene.
2. Go to **RuntimeGIProbes > Prefabs** and drag and drop the *GIBoundingBox* into your scene.
3. Go to **RuntimeGIProbes > Prefabs** and drag and drop the *GIBasicCamera* into your scene.

The *GI Light Baker* has all the *GI Probes* parented to it. You can duplicate and move them to fit the scene. You can view the results of the light baking in the inspector window.

The *GI Bounding Box* is used to contain the effect of the global illumination. For example, a sunny outdoor scene with a dark cave can have a bounding box filling the cave only.

The *GI Basic Camera* is a camera setup to view the global illumination. It contains the *GI Camera Properties* component, with the *GI Camera Shader* material pre-assigned.

4. VR Support

Runtime GI Probes supports VR using the Open XR package and single pass / multi-view rendering or multi pass rendering:

1. Go to **Edit > Project Settings > XR Plug-in Management > Mock HMD Loader** to preview the VR Rendering.
2. Go to **Edit > Project Settings > XR Plug-in Management > MockHMD > Render Mode** and set it to Single Pass Instanced.
3. Go to **RuntimeGIProbes > Prefabs** and drag and drop the *GIVRCamera* into your scene.

Following the above steps will allow you to see the global illumination in mock runtime, but it should still work using a VR headset. The default *GIVRCamera* is a camera setup to work in VR by setting the VR Camera Boolean to true in the *GI Camera Properties* component and setting target eye to Both on the Camera component.

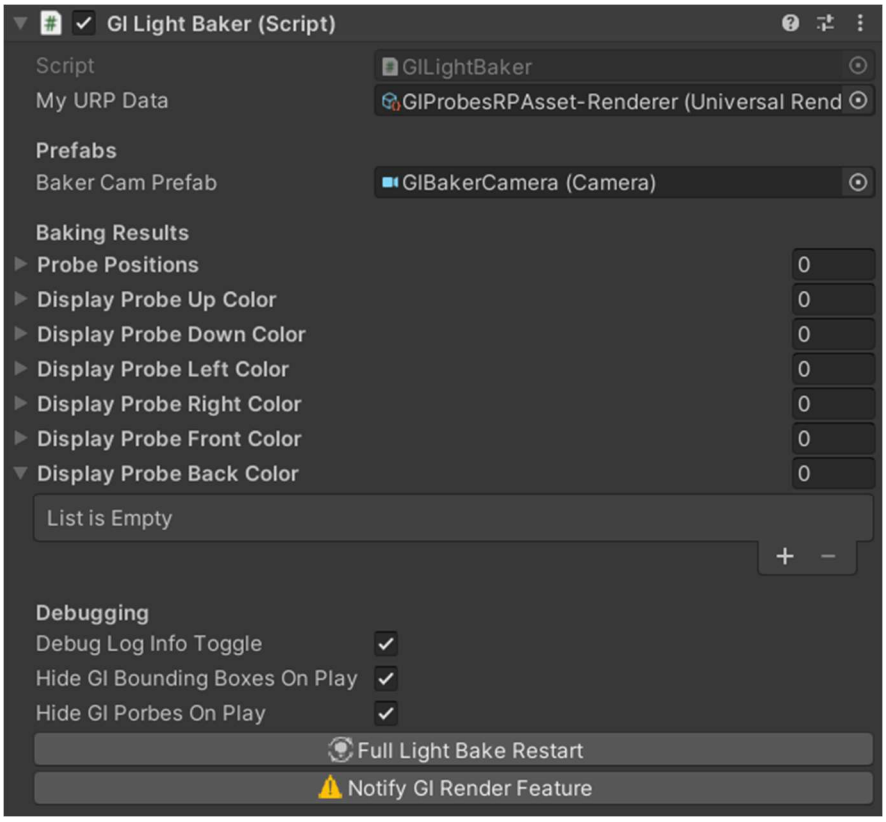
5. Mobile Support

Runtime GI Probes supports mobile through Vulkan:

1. Repeat basic setup steps except using *GIProbesRPAsset-Mobile*
2. Replace *GIProbesRPAsset-Renderer* with *GIProbesRPAsset-Renderer-Mobile* on *GI Light Bakers* and *DemoFreeCamera*.
3. Go to **Edit > Project Settings > Player > Other Settings (Android) > Graphics APIs** and select the "+" icon and select Vulkan.

On mobile, please lower the light map resolution settings on the *GI Render Feature* to a maximum of 64. The baking times will be too long if greater than that. To accommodate larger scenes, you will have to increase light map scaling at the expense of resolution.

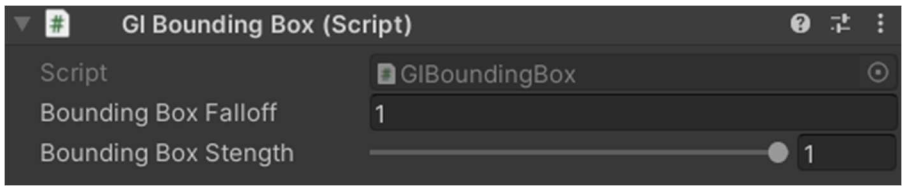
6. Component Inspector Variables



GILightBaker variables:

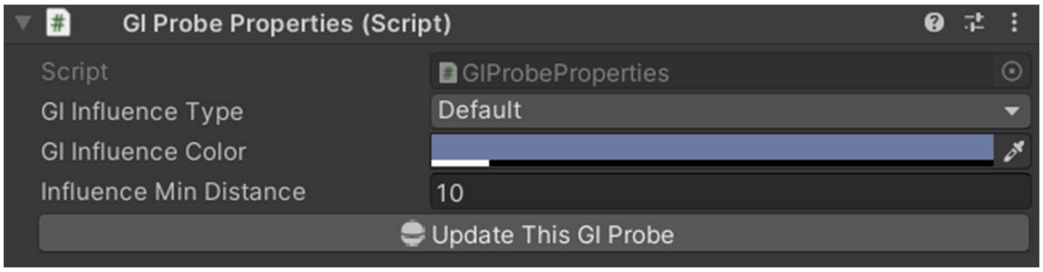
- **My URP Data**: Set to *GIProbesRPAsset-Renderer* by default, since it contains the GI Render Feature. Can be swapped out if you use your own render pipeline.

- **Baker Cam Prefab**: A default camera spawned in to capture global illumination information at each GI probe. This can be safely left alone for most purposes.
- **Debug Log Info Toggle**: The *GI Light baker* prints some debugging information by default.
- **Hide GI Bounding Boxes on Play**: Can be disabled to see *GI Bounding Boxes* in game.
- **Hide GI Probes on Play**: Can be disabled to see GI Probes in game.
- **Full Light Bake Restart**: Starts a full light bake restart for when the lighting of an entire scene has changed.
- **Notify GI Render Feature**: If the light maps have not updated after the light baker's values changed, this will directly tell the *GI RenderFeature* to recalculate the lightmaps.



GIBoundingBox variables:

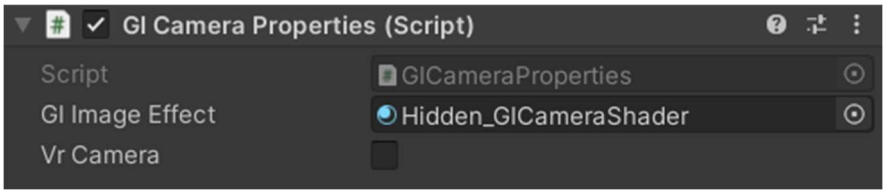
- **Bounding Box Falloff**: Distance outside the Bounding Box where its effect falls off creating a rounding / smoothening effect. You can manipulate these values in game and see results immediately.
- **Bounding Box Strength**: The strength Bounding Box from 0 to 1. You can manipulate these values in game and see results immediately.



GIProbeProperties variables:

- **GI Influence Type**: Determine how the GI Influence Color affects the GI Information caught at this GI Probe. The Default result will be un-tinted.

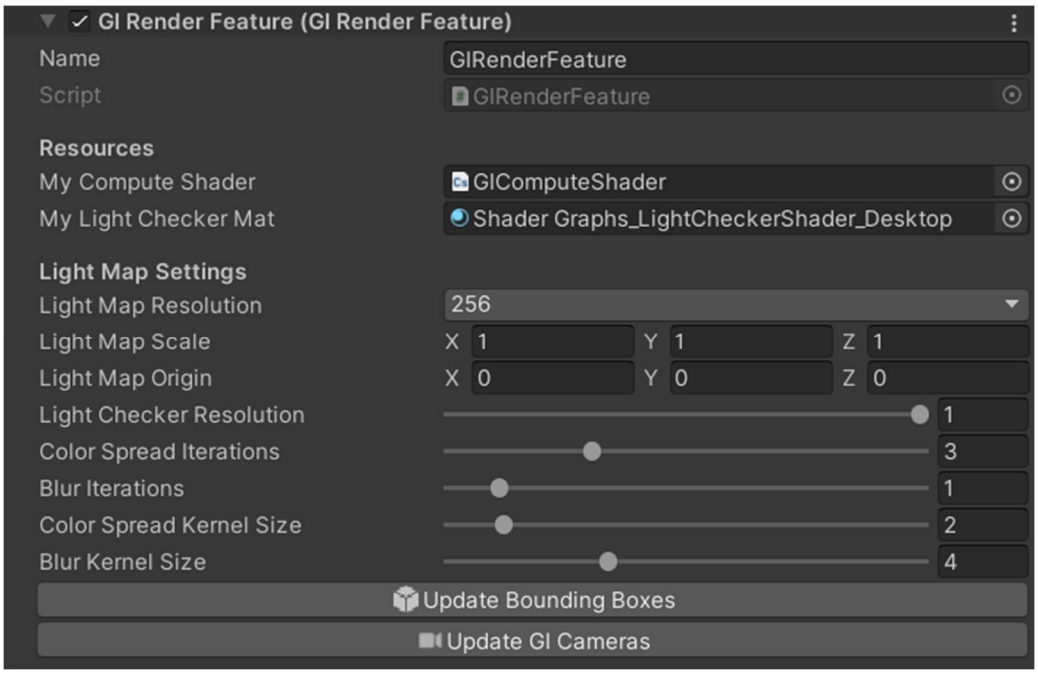
- **GI Influence Color**: The color used in tinting the GI probe info. The alpha value determines the strength of the influence type and an alpha of 0 will result in an equivalent to the Default influence type.
- **Influence Min Distance**: When multiple GI probes are in a scene the distance one probe fills in the light map is the minimum distance between it and any other probe. If an area of a map has tightly packed GI Probes, the coverage of those probes will be limited so setting a higher minimum distance may be necessary. This value is relative to the scale of the lightmap.
- **Update This GI Probe**: Updates the light baker when one of its existing GI probes has had its values changed or has been moved.



GICameraProperties variables:

- **GI Image Effect**: The GI Camera Shader material used by this camera.
- **VR Camera**: Used to determine if this camera is for VR. Make sure VR cameras have two child cameras for light checking both eyes. Only one VR Camera is allowed in a scene.

7. GI Render Feature Inspector Variables



- **My Compute Shader**: Should have *GIComputeShader* assigned to it.
- **My Light Checker Mat**: The Material used to create the direct lighting pass / light checker texture.
- **Light Map Resolution**: The resolution of the 3D lightmaps generated where each pixel at the default scale is a 1m x 1m x 1m area.
- **Light Map Scale**: Grow or shrink the area one pixel takes up in Lightmaps.
- **Light Map Origin**: The center of the lightmap in world space.
- **Light Checker Resolution**: The resolution of render textures used by the light checker texture, relative to the camera's resolution. Can be used to improve performance by decreasing their resolution, but will leave artifacts around the edges of shadows.
- **Color Spread Iterations**: How many times to perform color spreading operations (Fills in uncolored portions of lightmaps).
- **Blur Iterations**: How many times to perform blur operations.
- **Color Spread Kernel Size**: The area around a pixel used to get the average color in a color spreading operation.
- **Blur Kernel Size**: The area around a pixel used to get the average color in a blur operation.
- **Update Bounding Boxes**: When new bounding boxes are added during runtime, this will tell the *GI Render Feature* to look for all bounding boxes in the scene.
- **Update GI Cameras**: The area around a pixel used to get the average color in a blur operation.

8. Shader Variables

- **Fog Mode**: What fog mode to use, based on Unity's fog options. Since this image effect is drawn over opaque objects, we need to reapply fog.
- **Fog Color**: The color of the fog.
- **Density (Exponential Fog)**: How dense the fog is. Only does anything if using Exponential or Exponential Squared Fog.
- **Start (Linear Fog)**: Where the fog begins. Only does anything if using Linear Fog.
- **End (Linear Fog)**: Where the fog ends. Only does anything if using Linear Fog.
- **GI Hue Influence**: How much does global illumination color the scene.
- **GI Brightness**: The brightness of the global illumination.
- **GI Contrast**: The contrast of the global illumination.
- **GI Zone Falloff**: Globally adds to all bounding box falloff values.
- **GI Strength**: The overall strength of the global illumination effect.
- **Light Checker Strength (Direct Lighting)**: The strength the light checker texture has on the direct lighting input to this shader.
- **Screen Space Shadow Map Strength (Direct Lighting)**: The strength the Screen Space Shadow Map (if used) has on the direct lighting input to this shader.
- **Debug Direct Lighting**: See direct lighting info.
- **Debug Mode**: You can view the lightmap's influence on the scene on its own to better understand anything that looks incorrect.

9. Runtime GI Probes API

Below is a list of functions that can be called from your own scripts when using the *GIProbesRuntime* namespace. If there is a change in the lighting of a scene, the *GI Light Baker* can be passed a list of its GI probes to update before telling the *GI Render Feature* to recalculate the lightmaps.

- *GILightBaker*: **UpdateGIProbes (int whichGIProbe)**: Update a single GI probe before recalculating 3D Light Maps.
    - **whichGIProbe**: The sibling index of a probe parented to a light baker.
- *GILightBaker*: **UpdateGIProbes (int[] whichGIProbe)**: Update multiple GI probes before recalculating 3D Light Maps.
    - **whichGIProbe**: An array of sibling indexes of probes parented to a light baker.
- *GILightBaker*: **NotifyGIRenderFeatureFunction ()**: If the light baker information has not been updated in the light maps, this function can be called to directly tell the *GI Render Feature* to recalculate the lightmaps.
- *GILightBaker*: **AddGIProbe (GameObject whichGIProbe)**: Add a GI probe to the light baker.
    - **whichGIProbe**: The probe to add to the light baker.
- *GILightBaker*: **RemoveGIProbe (GameObject whichGIProbe)**: Remove a GI probe from the light baker.
    - **whichGIProbe**: The probe to remove from the light baker.
- *GILightBaker*: **FullLightBakeRestart ()**: Do a full restart of the light bake.
- *GILightBaker:* **fullLightBakeComplete**: A callback that is triggered when a light bake has completed. Useful for disabling a loading screen when the light bake is over.
- *GILightBaker:* **fullLightBakeRestart**: A callback that is triggered when a full light bake restart has begun. Useful for triggering the start of a loading screen.

If you want to change lightmap settings in real-time, the *GI Render Feature* will recalculate the lightmaps with your updated settings. If you want to add new cameras to a scene or add new *GI Bounding Boxes* to a scene in real-time, you will have to notify the *GI Render Feature*.

- *GIRenderFeature*: **UpdateGICameraInfo ()**: Update camera info when new GI Cameras are created or destroyed. Called from external scripts.
- *GIRenderFeature*: **UpdateBoundingBoxInfo ()**: Update bounding box info when new bounding boxes are created or destroyed. Called from external scripts.
- *GIRenderFeature*: **UpdateLightBaker ()**: Update Light baker if using a different Light baker. Called from external scripts.

10. Limitations

Runtime GI Probes will render before transparent objects, meaning transparent objects will be excluded when applying global illumination. This can be a problem if you have an existing project that makes heavy use of those kinds of materials for vegetation, water, or hair. Also, this can make *Runtime GI Probes* incompatible with other Unity assets that make use of transparent materials. For example, *Runtime GI Probes* would not work with Unity's built-in grass billboards since they use a transparent material. Opaque materials with alpha clipping are an exception to this since they are not transparent.

The maximum Light map resolution is 256 x 256 x 256. On mid to high end PC hardware, this size will produce a small lag spike. However, a size of 512 x 512 x 512 results in the computer freezing for a minute and a half since this is exponentially bigger. This is unacceptable and of course, anything larger than this is out of the question. On mobile devices, the maximum should be 64 x 64 x 64.

Lowering the resolution of the light checker texture can cause holes to appear in the texture since the material uses alpha clipping to hide objects behind in the depth buffer. If the light checker resolution is smaller than the depth buffer texture, there can be errors. The *Shader Graphs Light Checker Shader* material has a depth error variable to help correct this by lowering it. The desktop version has a depth error value of -0.01 since it is expected that the resolution be at 1 or 100%. The mobile version of the material is set to -0.5. The only way to completely remove artifacts though would be to set the light checker resolution at 1.

In a scene there can be a maximum of 128 *GI Bounding Boxes*.

11. Tips

If you want to move or edit *GI Bounding Boxes*, you can do that in real-time without having to call any functions in code. If you want to add a *GI Bounding Box* to your scene, you are going to have to call the UpdateBoundingBoxInfo function on the *GI Render Feature* or turn off the *GI Render Feature* and turn it back on again, since it automatically looks for all *GI Bounding Boxes* in its Create function.

Emissive materials become muted by the global illumination, since the effect of GI dimming overpowers the brightness of emission in materials. To work around this, you can make emissive materials transparent. If the environment lighting is too dark, the result of the light bake may be too dark. Try keeping some ambient lighting even in night time scenes, and let the GI make areas pitch black if it is appropriate.

If you have a larger area you want to cover with global illumination, make use of the scaling options in the *GI Render Feature* or move the origin point / offset of the lightmap in the *GI*

Render Feature depending on player position. Disabling GI probes can speed up light baking times, so if the player is in one location, you can disable all other GI probes to reduce the *GI Light Baker's* work load. At lower resolutions or larger scales, you may need to increase the influence minimum distance of GI probes to avoid making a scene too dark.

Getting the right lighting may take a couple iterations of placing around GI Probes and seeing if it looks good. In general, you do not need many GI Probes to capture the global illumination of a simple room and it is better to keep the GI setup simple. Use the demo scene for reference, you would be surprised at how a small number of GI probes can illuminate a large area. The result is not going to have the level of detail found in a traditional baked lightmaps; the idea is to capture subtle ambient lighting like SSGI (Screen Space Global Illumination).