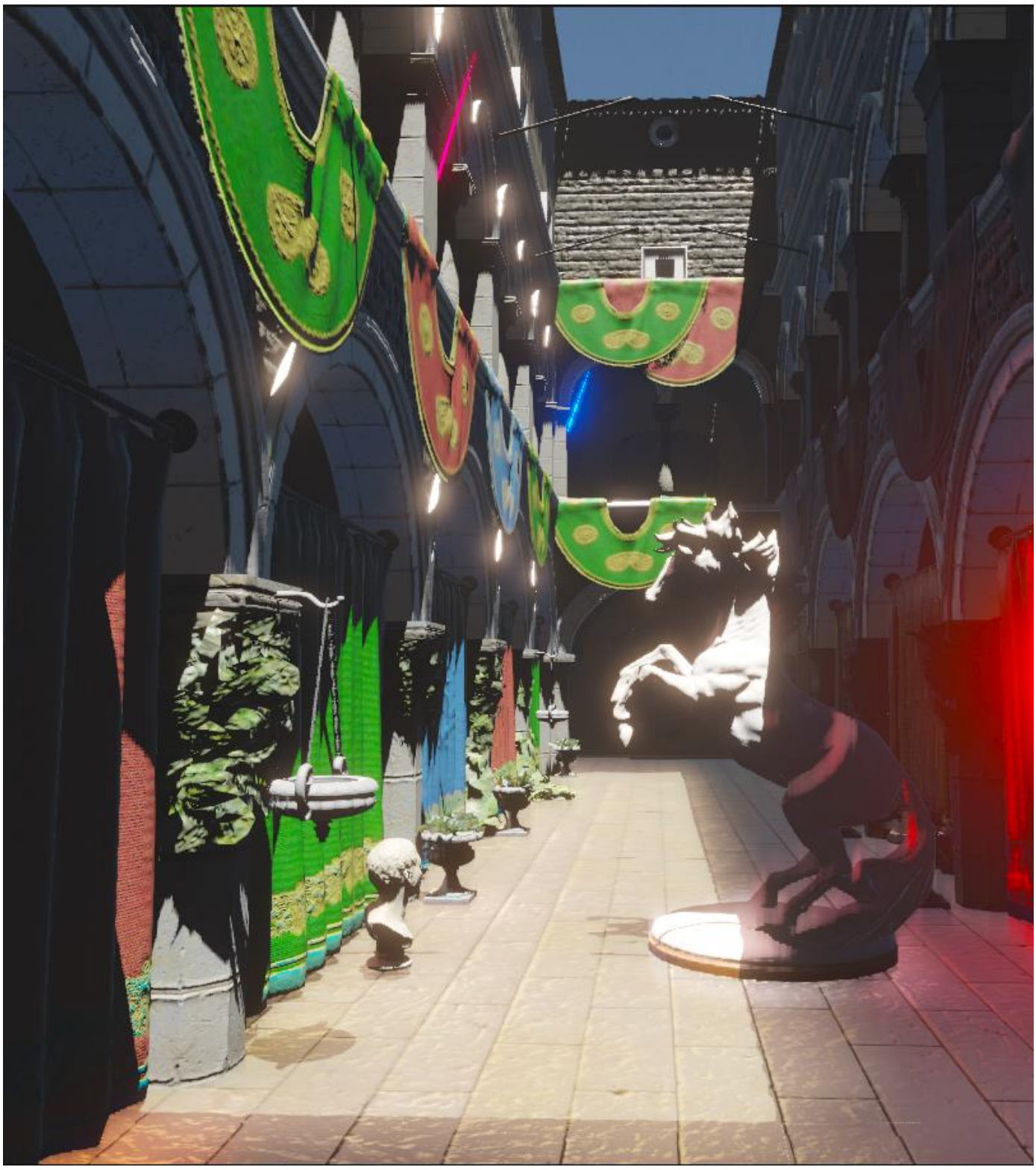
LUMINA 2024 URP Real-Time Global Illumination



LUMINA 2024 v2.3

LUMINA 2024 URP Setup Guide

Overview

The **LUMINA URP Real Time Global Illumination** system is a realistic lighting framework for directly bringing your scenes to life with real to life indirect and direct illumination additionally to Unity base lighting.

The system is compatible with **Sky Master ULTIMATE URP** version, and can be combined with the **Ethereal** module to add volumetric lighting to further enhance the look.

The system also includes a screen space sun shafts effect to quickly add volume to the atmosphere when the sun is inside or near the camera field of view.

LUMINA GI can be upgraded with a big discount to the **Environment Building Bundle** that includes **Sky Master ULTIMATE** and all other major **ARTnGAME** assets.

For quick start on the main system, please refer to the tutorial video playlist in the following link:
LUMINA Tutorial Playlist

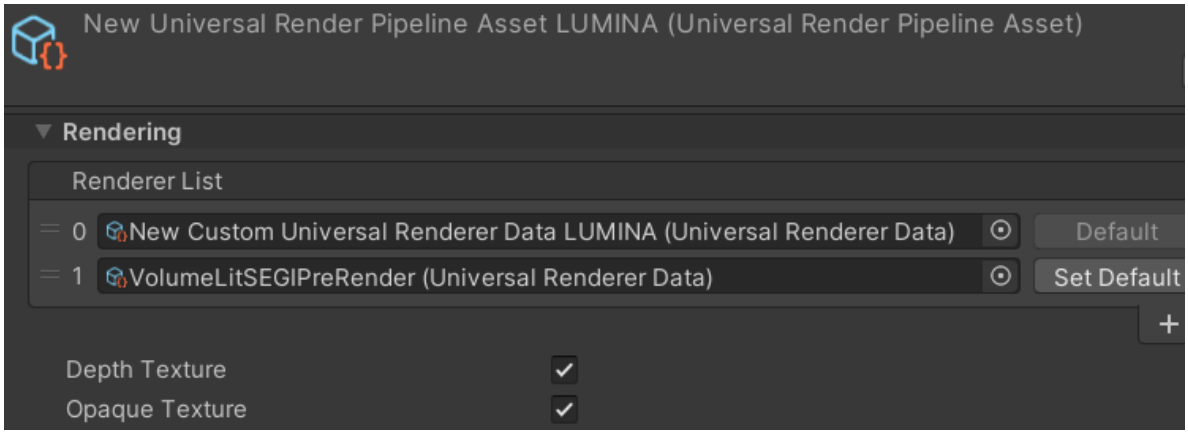
Video playlist showcasing the use of LUMINA with Sky Master ULTIMATE URP Volumetric Lighting & Clouds.
LUMINA with Sky Master ULTIMATE URP Video Playlist

For any questions please contact me in ARTnGAME discord channel:
https://discord.gg/X6fX6J5

Or in ARTnGAME email:
artengames@gmail.com

LUMINA URP Setup steps

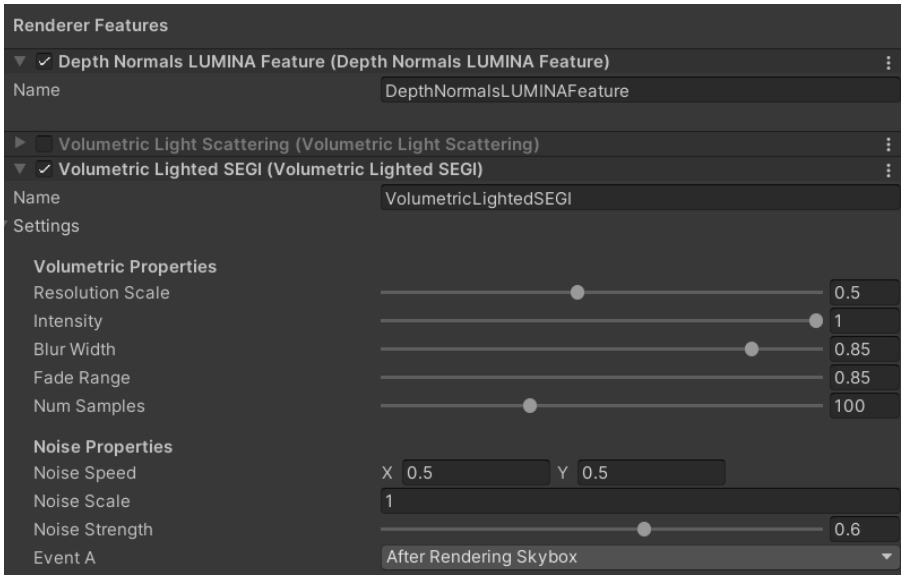
The system setup requires the URP pipeline to have been setup and two renders assigned to the pipeline Renders list. Unity 2022.3 and above is required. The system has been tested up to Unity 6000.0.25, and in the new URP RenderGraph.



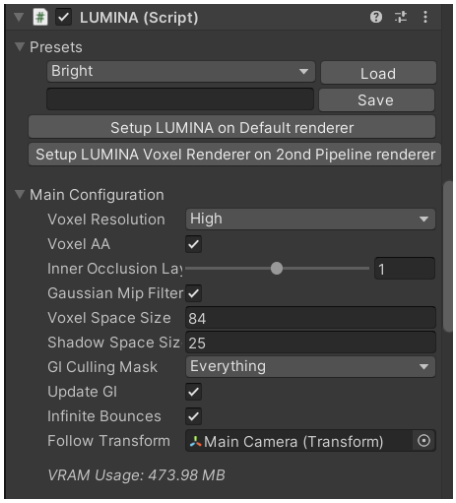
The first main renderer that will work on the main scene camera must have the "**Volumetric Lighted SEGI**" renderer feature for the main Lumina effect. Also the **"Depth Texture"** and **"Opaque Texture"** checkboxes must be enabled and **Linear color space** must be used.

The setup of the system is done by placing the "**LUMINA**" script in the Main Camera in the scene. Make sure the camera is also tagged as "**MainCamera**" and is assigned the renderer that was added above to hold the LUMINA renderer feature. Then can add the modules step by step, by pressing the relevant buttons. The "`Setup LUMINA on Default renderer`" button will setup the Lumina required renderer feature in the Default URP pipeline Forward renderer (**VolumetricLightedSEGI**).

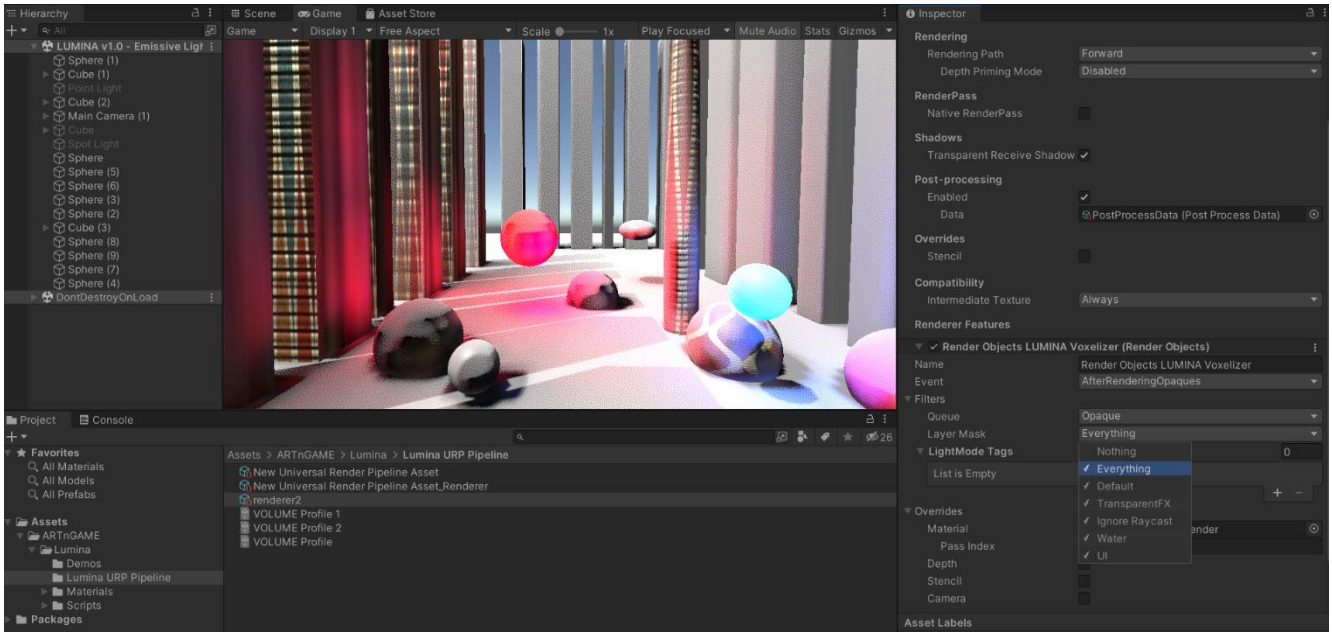
Alternatively can manually add it to the renderer, same as with other renderer features.



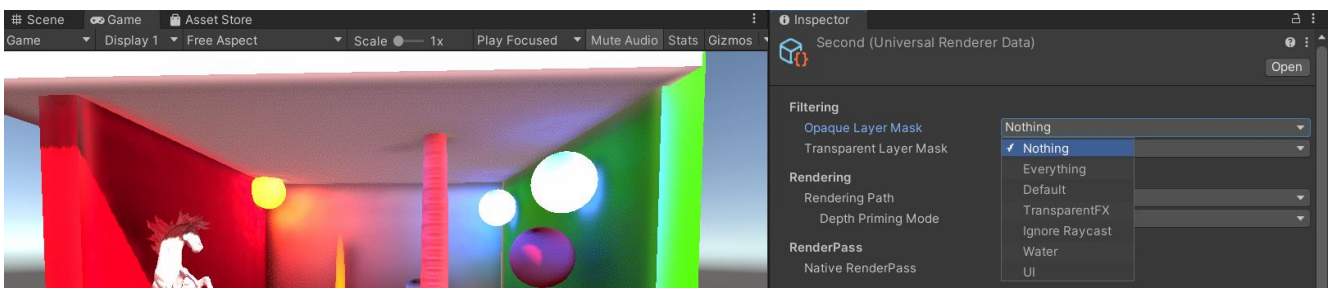
The "`Setup LUMINA Voxel Renderer on 2ond Pipeline renderer`" button will setup **the second renderer** with a renderer feature that voxelizes the scene. If the effect is needed to be setup to a different renderer, please first set the "**HelpererRendererID**" variable to the proper number corresponding to its position in the pipeline Renders list and then press the setup button. If the renderer changes order in the pipeline Renders list, should also set the "**HelpererRendererID**" variable to the new corresponding number.



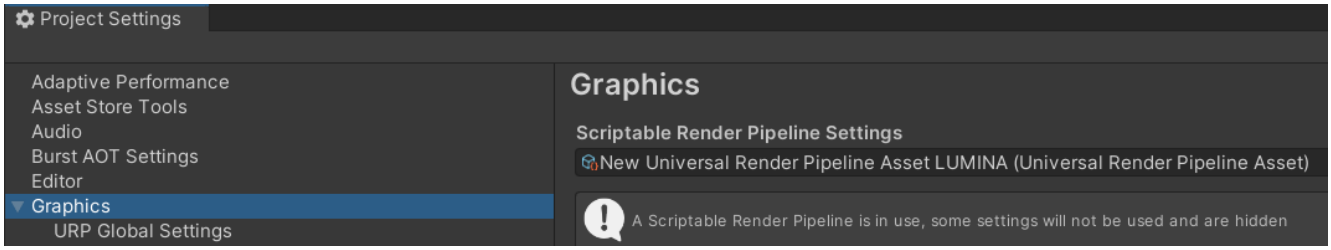
Make sure that the pipeline has two renderers before apply the button. Additionally, after adding the Render Objects feature to the second renderer, must select the layers to render GI from, as shown in the following, in the **Layer Mask** pull down menu, in the **Render Objects** renderer feature.

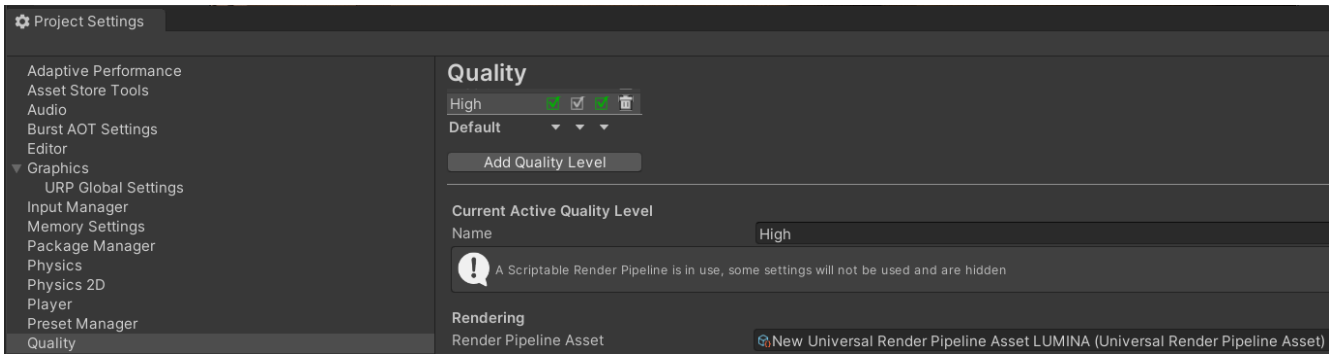


Also **in the 2ond renderer should disable all layers in the Filtering section**, as shown below and must be set in Deferred mode.



NOTE: The fastest method to have the system setup ready for use is to apply the included sample pipeline named as "**New Universal Render Pipeline Asset LUMINA**". The pipeline must be inserted in both the **Graphics** and **Quality** Unity settings, in the relevant pipeline slots, as shown below.

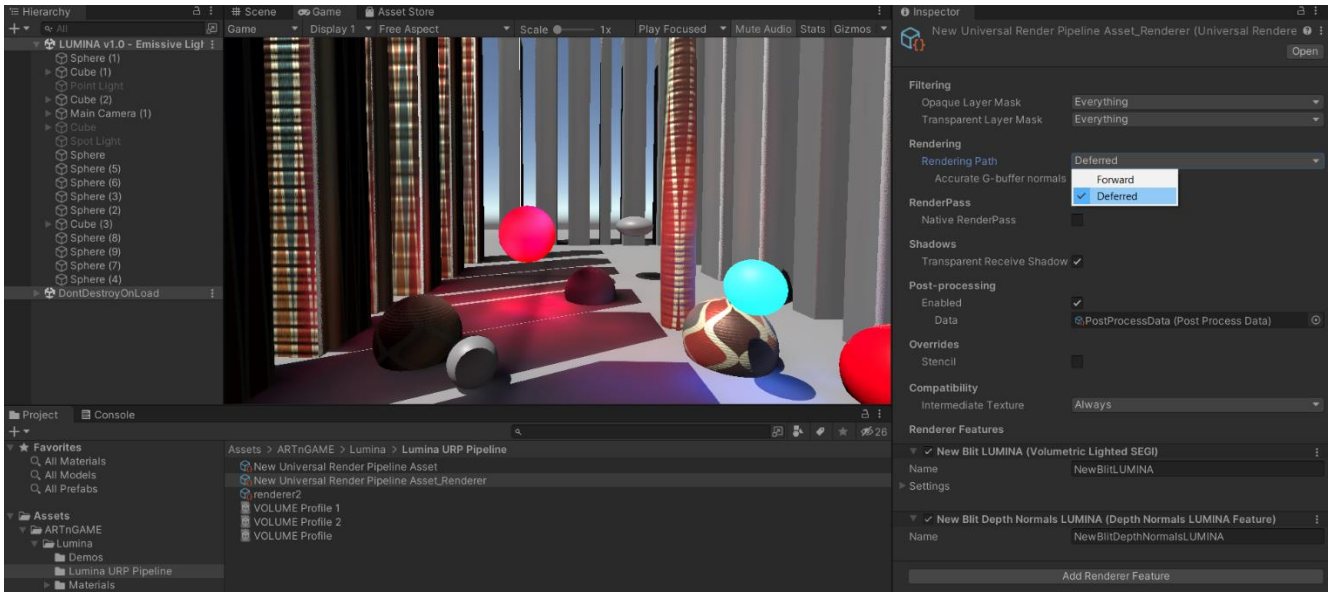




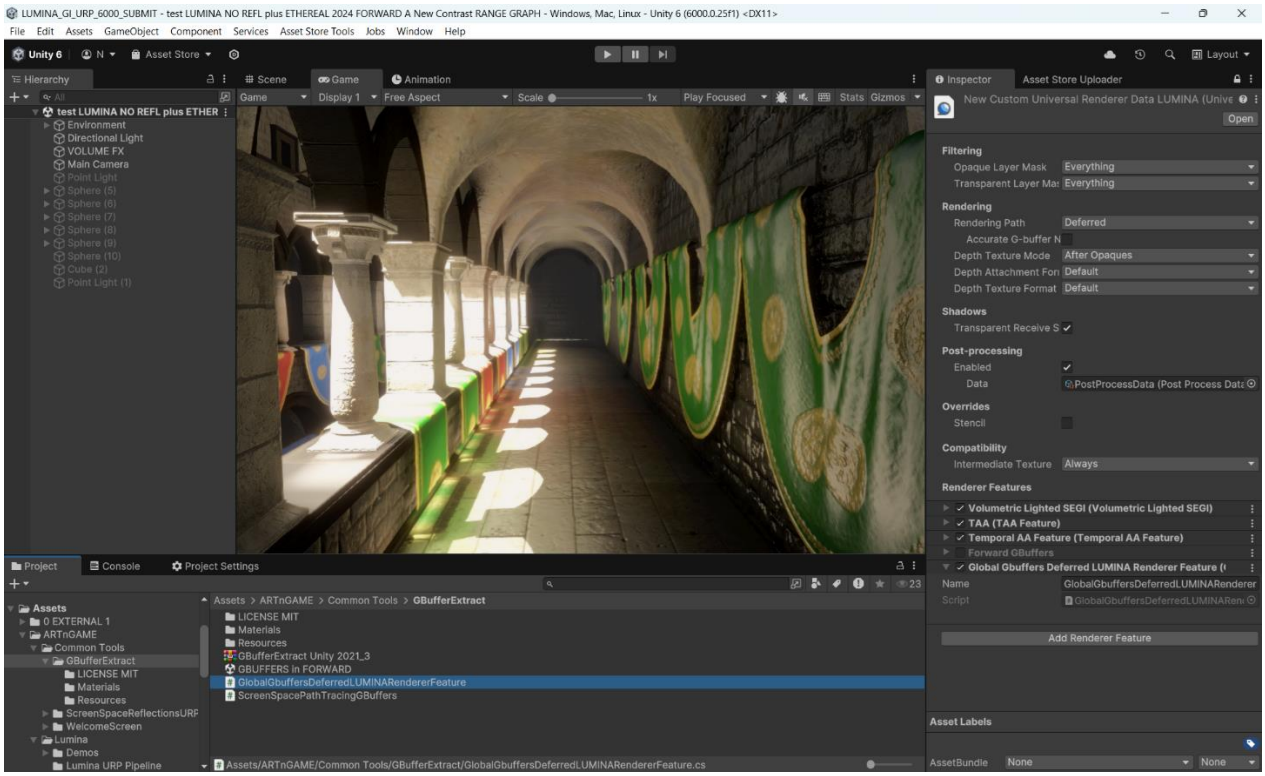
Rendering Modes

Deferred in Unity 2022.3

The LUMINA renderer must be set to the **Deferred rendering mode** as shown below, otherwise if set to Forward or Forward+, will require one extra renderer feature.



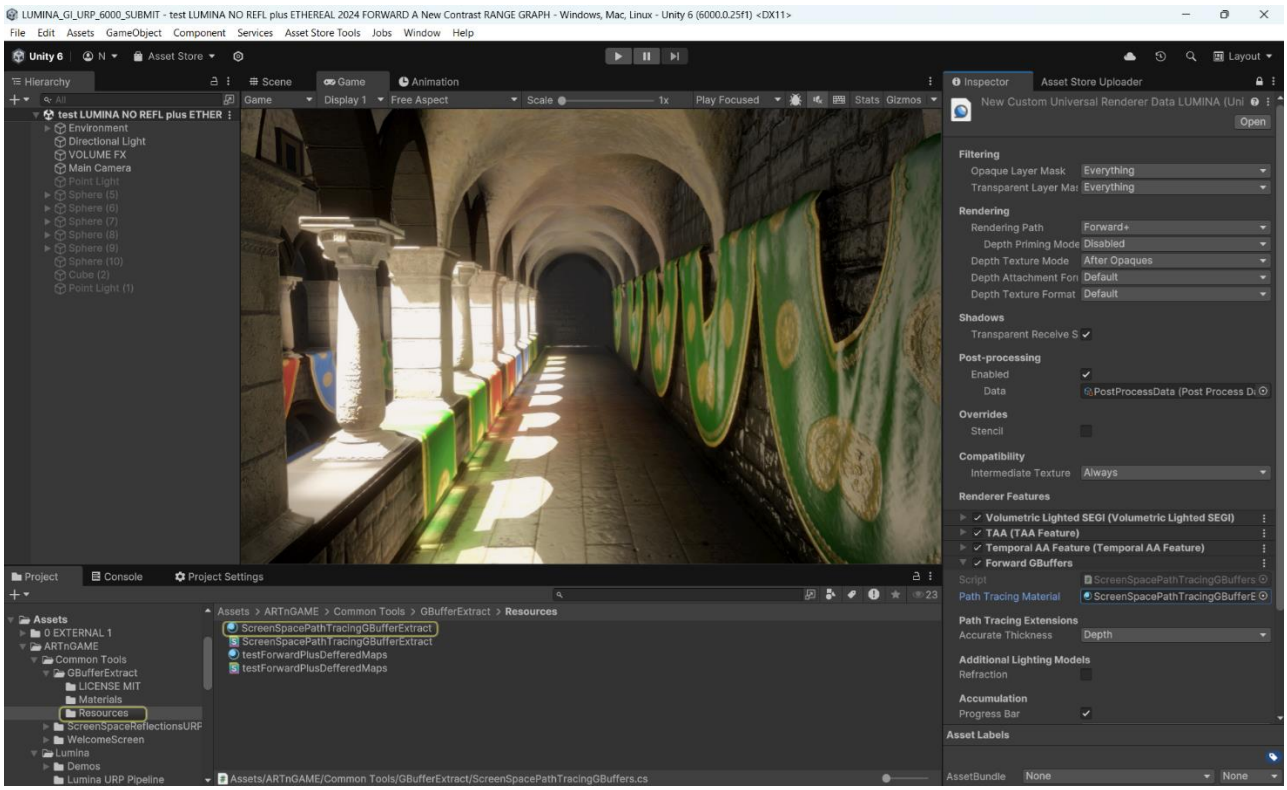
Deferred in Unity 6000.0.25



If the renderer is set to Deferred in Unity 6 LTS, will require one extra renderer feature added to the renderer, named "**Global Gbuffers Deferred LUMINA**", as shown above.

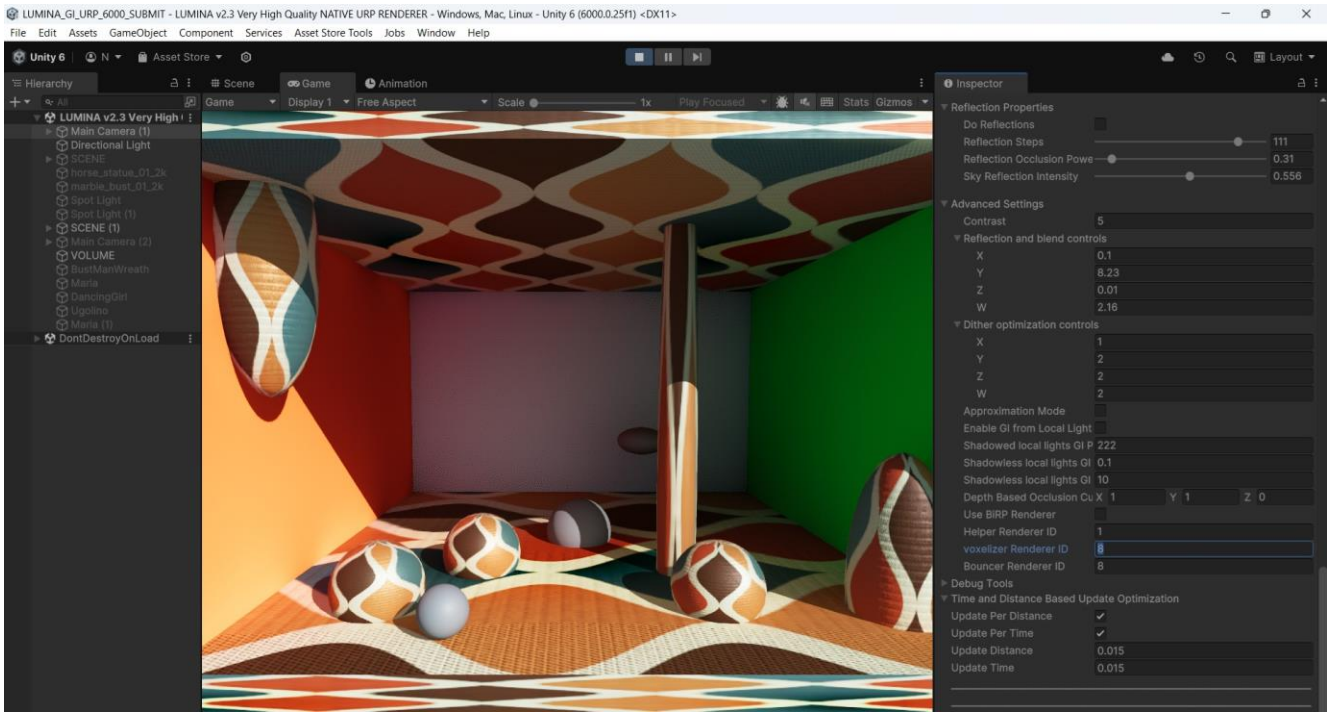
Forward and Forward+ in Unity 2022.3 and Unity 6000.0.25

If the renderer is set to Forward or Forward+, will require one extra renderer feature, named "**Forward Gbuffers**" as shown below.



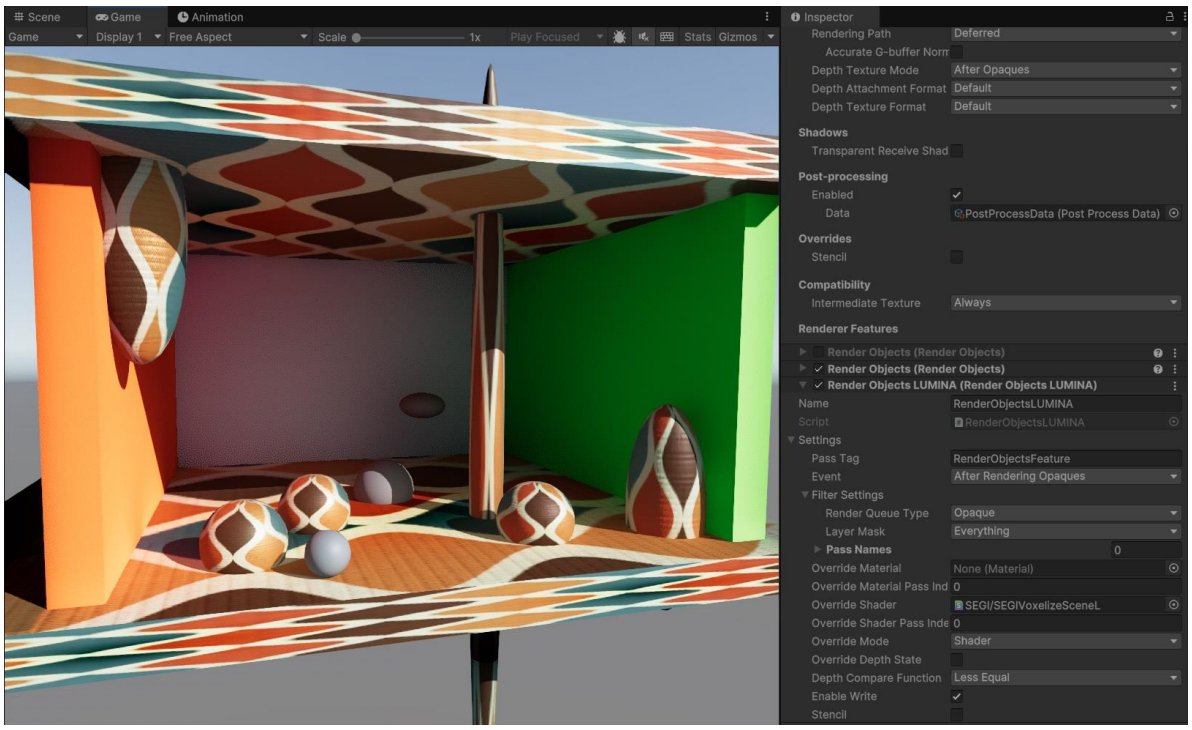
LUMINA URP Native Mode

The system by default uses the Standard BiRP Pipeline to do the scene voxelization. In LUMINA 2024 v2.3 a new option has been added **for Unity 6 RenderGraph** to **use the native URP pipeline for the voxelization**. For this mode first disable the "**Use BiRP Renderer**" checkbox as shown below.



Then select the number of the renderer that has the following setup, with the "**RenderObjectsLUMINA**" renderer feature, e.g the 8th in the included sample LUMINA pipeline. For RenderGraph can choose the same number for both Voxelizer and Bounce renderers.

The "**RenderObjects**" feature is only applicable in the Compatibility mode in Unity 6, in which case should also use a different renderer for the Voxelizer and Bounce renderers. In the included sample LUMINA pipeline is renderer 8 and 9.



LUMINA Parameters

The system has an extensive range of control variables to regulate the effect as required per scene. The variables are explained in the following, the format is variable name in scripts for API reference, variable name and details. **The scene directional light must be set** in the **"Sun"** slot of the system.

1. Main Configuration

(smoothNormals) – Smooth Normals, This new option corrects the scene normals, with one using the precision normals and any other value can be used for other artistic options. It is recommended to use the value of 1 as base value to configure the scene with the exact normals and then regulate for any extra effect if needed.

(voxelResolution) - Voxel Resolution, The resolution of the voxel texture used to calculate GI. This is the most important variable for performance, thus depending on the target platform should be carefully regulated.

(voxelAA) - Voxel AA, Enables 8x supersampling anti-aliasing during voxelization for higher precision voxels. The performance cost depends on the scene complexity and this option helps with smaller and moving objects contribution to the indirect lighting.

(innerOcclusionLayers) - Inner Occlusion Layers, Enables the writing of additional black occlusion voxel layers on the back face of geometry. Can help with light leaking but may cause artifacts with small objects.

(gaussianMipFilter) - Gaussian Mip Filter, Enables gaussian filtering during mipmap generation. This can improve visual smoothness and consistency, particularly with large moving objects. This option has a big performance impact when the voxel resolution is high.

(voxelSpaceSize) - Voxel Space Size, The size of the voxel volume in world units. Everything inside the voxel volume will contribute to GI. The higher this value the bigger the created voxels will be, to cover the extra space. Select the camera with LUMINA in scene view to visualize the volume in the scene.

(shadowSpaceSize) - Shadow Space Size, The size of the sun shadow texture used to inject sunlight with shadows into the voxels in world units. It is recommended to set this value similar to Voxel Space Size.

(giCullingMask) - GI Culling Mask, Which layers should be voxelized and contribute to GI. This is very useful to both enhance performance and also create items that will cast GI but not appear in the scene and other possible manipulations.

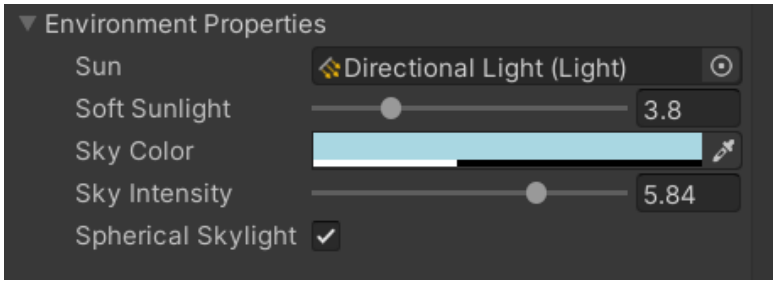
(updateGI) - Update GI, Whether voxelization and multi-bounce rendering should update every frame. When disabled, GI tracing will use cached data from the last time this was enabled. If the scene is small and not have moving objects or when is known objects that give lot of GI wont move, can be disabled on the fly to enhance performance.

(infiniteBounces) - Infinite Bounces, Enables infinite bounces. This is expensive for complex scenes and is still experimental. Also will have a small lag behind the light changes in the scene as is iteratively updated.

(followTransform) **- Follow Transform**, If provided, the voxel volume will follow and be centered on this object instead of the camera. Useful for top-down scenes.

(vram) **– VRAM Usage**, Shows how much VRAM is used by the voxelization solution.

2. Environment Properties



(Sun) **– Sun**, The main directional light that will cast indirect light into the scene (sunlight or moonlight). If not assigned the sun light wont contribute to the GI.

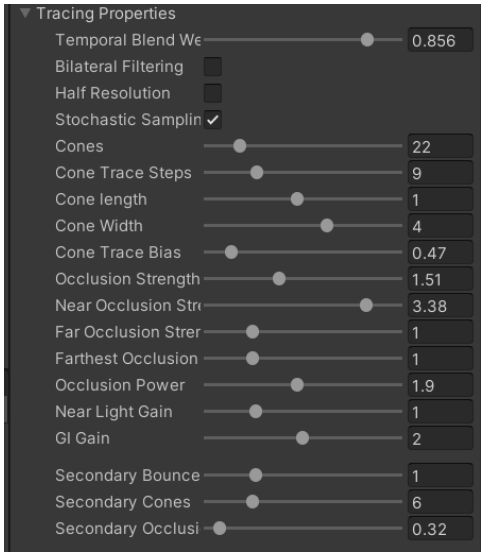
(softSunlight) **- Soft Sunlight**, The amount of soft diffuse sunlight that will be added to the scene. Use this to simulate the effect of clouds/haze scattering soft sunlight onto the scene.

(skyColor) **- Sky Color,** The color of the light scattered onto the scene coming from the sky.

(skyIntensity) **- Sky Intensity,** The brightness of the sky light.

(sphericalSkylight) **- Spherical Skylight,** If enabled, light from the sky will come from all directions. If disabled, light from the sky will only come from the top hemisphere.

3. Tracing Properties



(temporalBlendWeight) **- Temporal Blend Weight**, The lower the value, the more previous frames will be blended with the current frame. Lower values result in smoother GI that updates less quickly. A value of one will not take into account previous frames.

(useBilateralFiltering) **- Bilateral Filtering**, Enables filtering of the GI result to reduce noise, use with **Stochastic Sampling**. This is an experimental feature.

(halfResolution) - **Half Resolution**, If enabled, GI tracing will be done at half screen resolution. Improves speed of GI tracing.

(stochasticSampling) - **Stochastic Sampling**, If enabled, uses random jitter to reduce banding and discontinuities during GI tracing.

(cones)**– Cones,** The number of cones that will be traced in different directions for diffuse GI tracing. More cones result in a smoother result at the cost of performance.

(coneTraceSteps) - **Cone Trace Steps**, The number of tracing steps for each cone. Too few results in skipping thin features. Higher values result in more accuracy at the cost of performance.

(coneLength) - **Cone length,** The number of cones that will be traced in different directions for diffuse GI tracing. More cones result in a smoother result at the cost of performance.

(coneWidth) - **Cone Width,** The width of each cone. Wider cones cause a softer and smoother result but affect accuracy and increase over-occlusion. Thinner cones result in more accurate tracing with less coherent (more noisy) results and a higher tracing cost.

(coneTraceBias) - **Cone Trace Bias,** The amount of offset above a surface that cone tracing begins. Higher values reduce "voxel acne" (like "shadow acne"). Values that are too high result in light leaking

(occlusionStrength) - **Occlusion** Strength, The strength of shadowing solid objects will cause. Affects the strength of all indirect shadows.

(nearOcclusionStrength) - **Near Occlusion Strength**, The strength of shadowing nearby solid objects will cause. Only affects the strength of very close blockers.

(farOcclusionStrength) - **Far Occlusion Strength**, How much light far occluders block. This value gives additional light blocking proportional to the width of the cone at each trace step.

(farthestOcclusionStrength) - **Farthest Occlusion Strength**, How much light the farthest occluders block. This value gives additional light blocking proportional to (cone width)^2 at each trace step.

(occlusionPower) - **Occlusion Power**, The strength of shadowing far solid objects will cause. Only affects the strength of far blockers. Decrease this value if wide cones are causing over-occlusion.

(nearLightGain) - **Near Light Gain**, Affects the attenuation of indirect light. Higher values allow for more close-proximity indirect light. Lower values reduce close-proximity indirect light, sometimes resulting in a cleaner result.

(giGain) - **GI Gain**, The overall brightness of indirect light. For Near Light Gain values around 1, a value of 1 for this property is recommended for a physically-accurate result.

(secondaryBounceGain) - **Secondary Bounce Gain**, Affects the strength of secondary/infinite bounces. Be careful, values above 1 can cause runaway light bouncing and flood areas with extremely bright light!

(secondaryCones) - **Secondary Cones**, The number of secondary cones that will be traced for calculating infinite bounces. Increasing this value improves the accuracy of secondary bounces at the cost of performance. Note: the performance cost of this scales with voxelized scene complexity.

(secondaryOcclusionStrength) - **Secondary Occlusion Strength**, The strength of light blocking during secondary bounce tracing. Be careful, a value too low can cause runaway light bouncing and flood areas with extremely bright light!

4. Reflection Properties

Reflections is an experimental feature and will be updated with more options added in next versions.

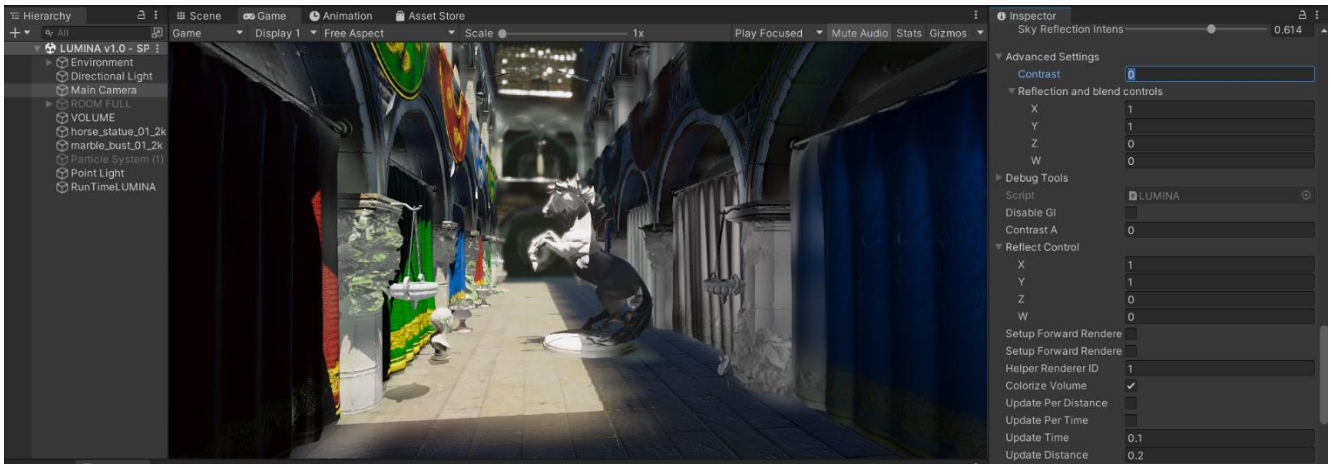
(doReflections) - **Do Reflections**, Enable this for cone-traced reflections.

(reflectionSteps) - **Reflection Steps**, Number of reflection trace steps.

(reflectionOcclusionPower)-**Reflection Occlusion Power**, Strength of light blocking in reflection trace

(skyReflectionIntensity) - **Sky Reflection Intensity**, Intensity of sky reflections.

5. Extra properties



The system also has few advanced variables, **ContrastA, Dither Optimization, Approximation Mode** and **Reflect Control** that are experimental. The **"Clear Bounce Render Target"** can be used for extra brightness in the scene. The **Update per Distance** and **Update per Time** can be used to regulate the voxel solution update of the scene per time or per distance when the player moves, by time and distance intervals regulated by the **Update Time** and **Update Distance** fields.

Note that when the player is not moving, the voxel solution is calculated always, or by time interval. Those variables are not saved or restored through the Presets system.

(contrastA) - **Contrast**, Change scene Contrast by (1-contrastA)*_GBuffer3 + (contrastA x _GBuffer1 x _GBuffer3) if **set to 20**. The GBuffers correspond to the URP Deferred mode GBuffers. Note the contrast also adds an extra red tint. **Set to zero** for not use of extra contrast (normal mode). In **LUMINA 2024** can also **set the variable to 5** to enable a higher contrast mode with more correct color scene grab. The GI power may need to be readjusted based on the chosen mode.

(ReflectControl) - **Reflection and blend Controls**, Specular power(X), GBuffer0 (Y) power, GBuffer1 (Z) power, Z=Not used.). The GBuffers correspond to the URP Deferred mode GBuffers.

(DitherControl) **– Dither optimization Controls**, Enable Dither X (0 disable, 1 enable), Y is dither divider over trace step, Z is dither divider over trace secondary tracing, W is dither divider over reflection steps. The values should be above one, otherwise the system will use more resources. Also should not use values of zero or below zero in Y-Z-W.

(ApproximationMode) **– Approximation Mode**, This is a new experimental mode in v1.4 that allows the voxelization of the scene without using Geometry Shaders, this potentially can make the system compatible with platforms that not support the Geometry Shaders. This system is less accurate and requires different variable tweaking then the main mode. Can be used for different artistic effect as well, as can create more exaggerated color grabbing in some configuration. The demo scenes with PROXY word in their name implement this new mode.

NOTE: The Approximation Mode currently can only be previewed when the editor is in play mode, so is best practice to enter play mode to tweak the effect & copy the script properties to the editor.

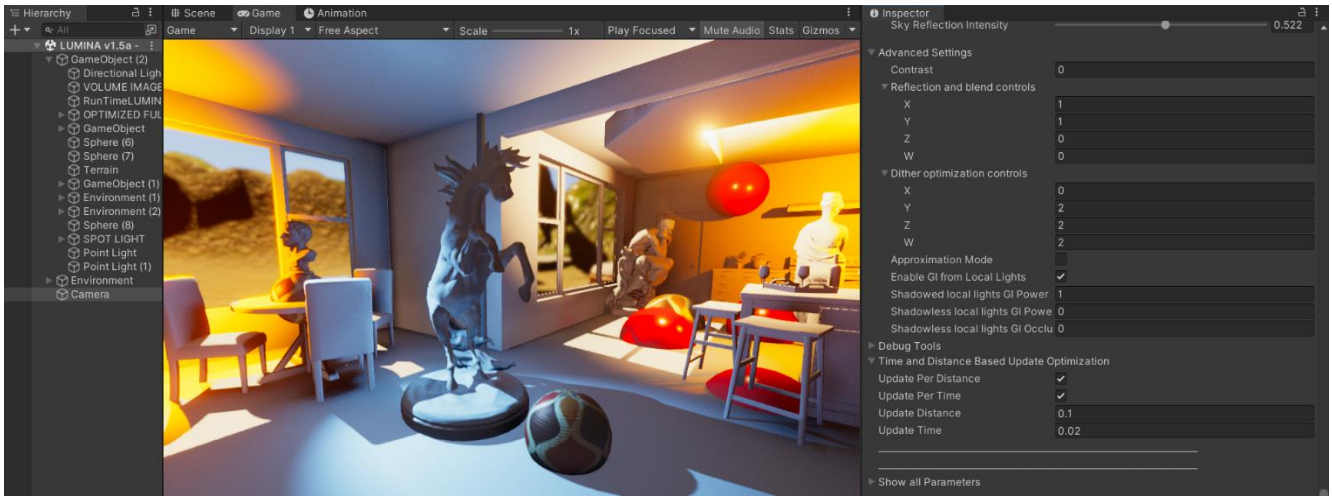
(EnableGIFromLocalLights) **– Enable GI from Local Lights Mode**, This new in v1.5 mode allows the Spot and Point lights in the scene to cast GI from the light they apply to the surfaces.

(ShadowedLocalLightsGIPower) **– Shadowed local lights Power**, Power of local lights GI respecting the shadows they cast on the environment.

(ShadowlessLocalLightsGIPower) **– Shadow less local lights Power**, Power of local lights GI in their whole volume, independent of scene occlusion.

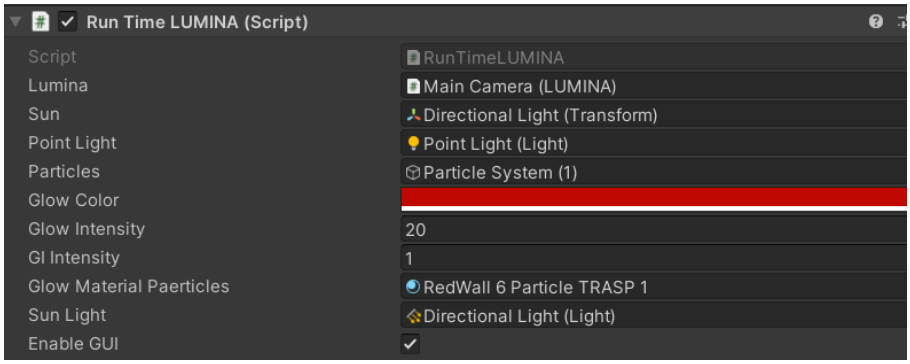
(ShadowlessLocalLightsGIOcclusion) **– Shadow less local lights occlusion**, Power of local lights GI occlusion applied from the voxel space.

(DepthBasedOcclusionCutoff) **– Depth based occlusion**, use of camera depth to eliminate some of the possible light leaking. This is an experimental option.



6. Run Time LUMINA Control and API

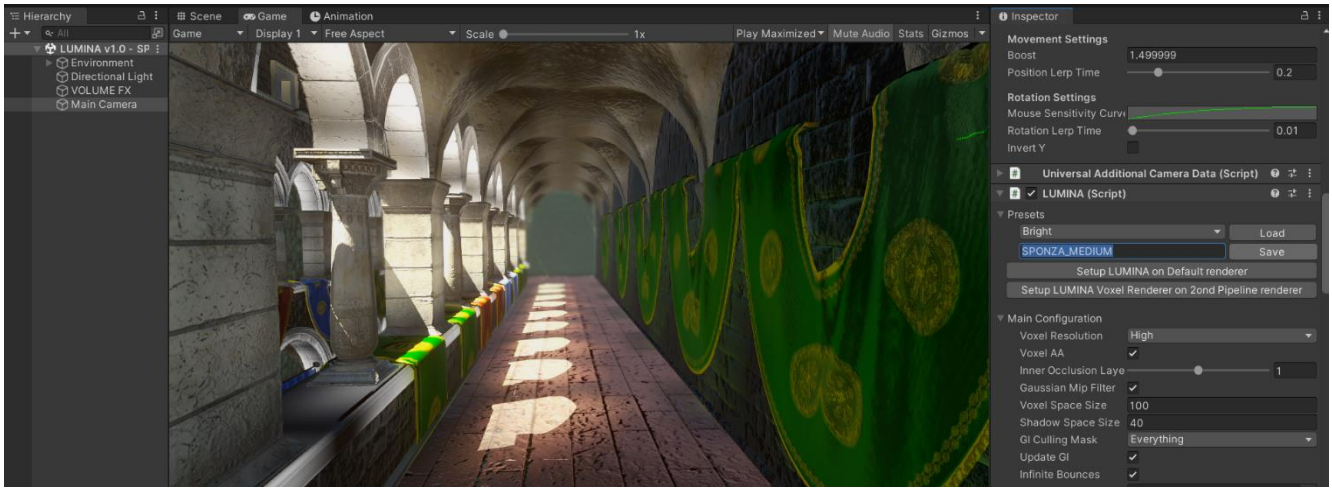
A run time controller is available as shown below (**RunTimeLUMINA**) that provides some basic run time controls and a GUI to access them and can be used as an API control sample and base for further runtime customization and controls.



7. Presets

The system allows the save of presets of parameters and load them in another scene. Note that optimization variables (Time and Distance based updating regulation) are not saved-loaded with the system.

Enter a name next to the “Save” button text field and press the “Save” button to save the settings. Then in a new scene, select a preset from the pull down menu next to the “Load” button and press the “Load” button to enable the preset variables.

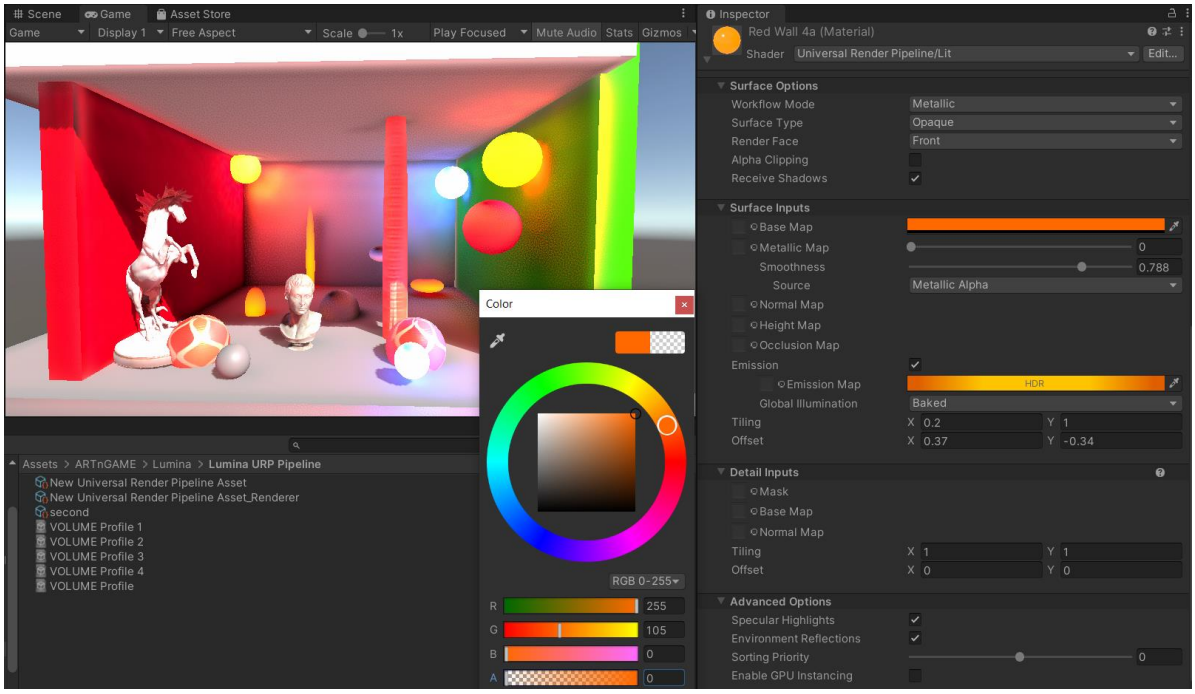


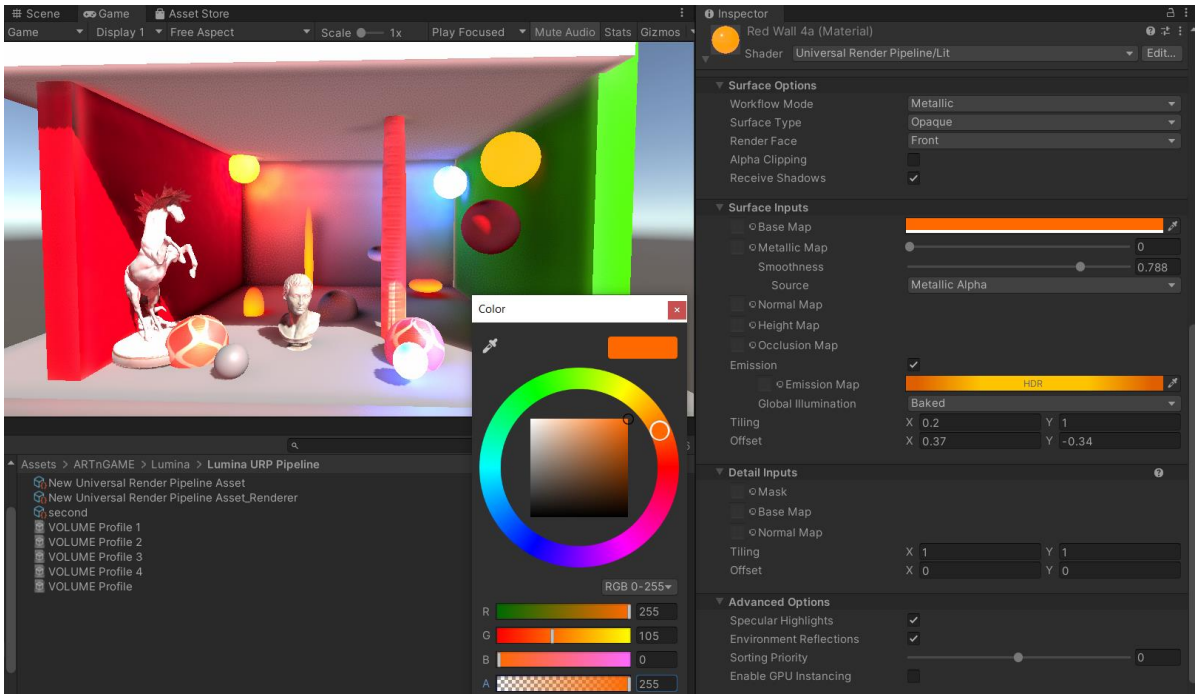
Presets can be saved in run time as well, so can preview changes in run time for a more accurate representation using the full updating cycles and save the preset with those settings.

8. Create Light sources

The system can **create a light source from any mesh with an emissive material**, the **more the HDR emission color the larger the light contribution to the GI**. In some cases it is needed to add a light this way but not show the mesh, this is possible by assigning the mesh with the emissive material to a layer that is excluded from the Main camera culling mask, but is included in the "**GI Culling Mask**" pull down menu Layers. This will allow the object to be calculated in the GI lighting, but not appear in the scene.

The effect **can be regulated using the alpha value of the main color** of the emissive material. Use a zero alpha for maximum emission contribution to the GI and a full alpha for minimum, as in the following images showing the regulating of the orange top right emitting mesh.





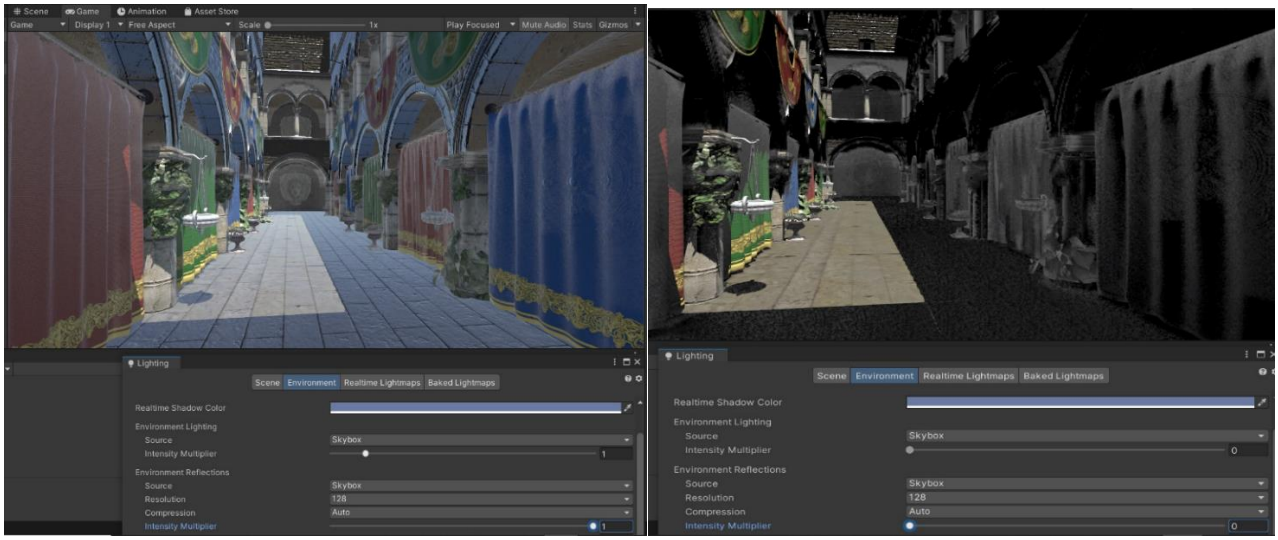
9. Scene Voxelization by layers

The system applies GI by voxelizing the scene. Specific layers may be chosen to be included in this voxelization for artistic and performance reasons, using the “GI Culling Mask” pull down menu Layers.

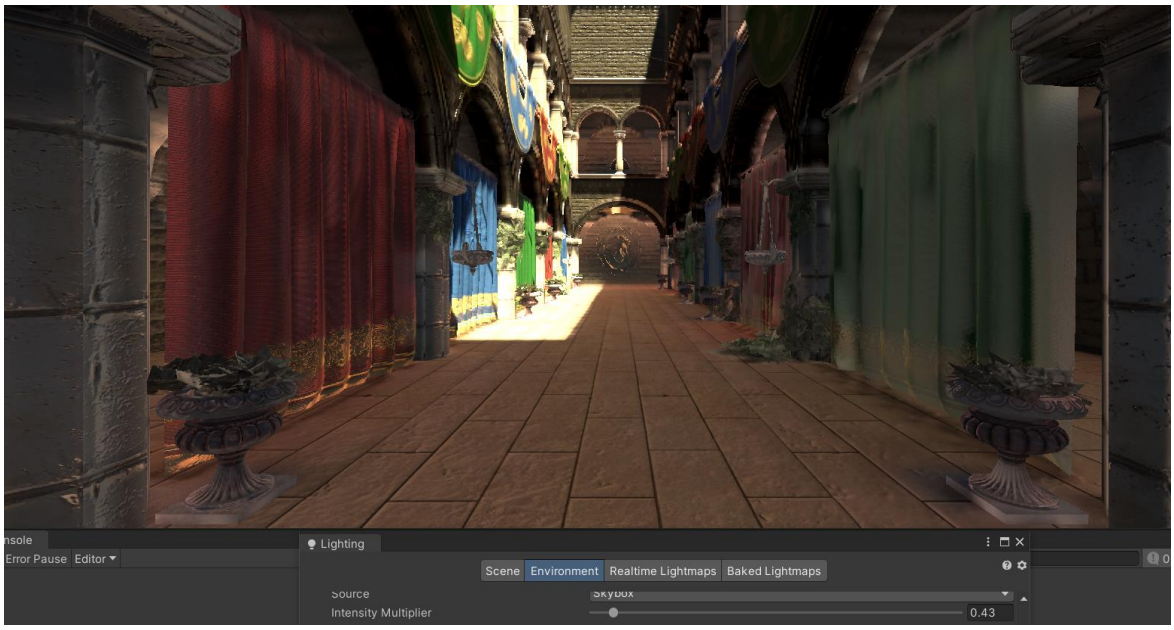
Also it is possible to use a lower polygon mesh for the GI contribution and voxelization, to increase performance. The layer of the low end models should be excluded from the main camera.

10. Ambient regulation

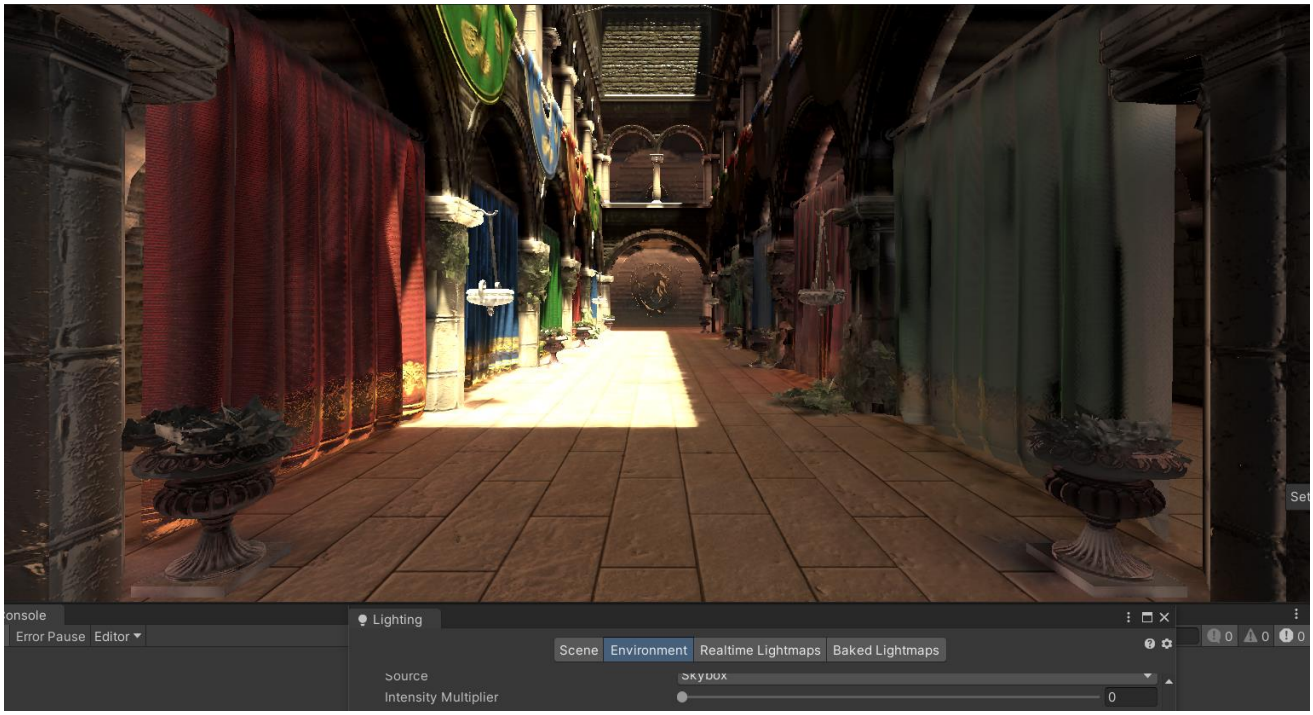
The system has its own sky contribution to the scene, but the Unity ambient may also be set to higher than zero to contribute as well for a more smooth effect. The two images show the Atrium scene with and without the Unity ambient, without the LUMINA GI. (Video A) – (Video B)



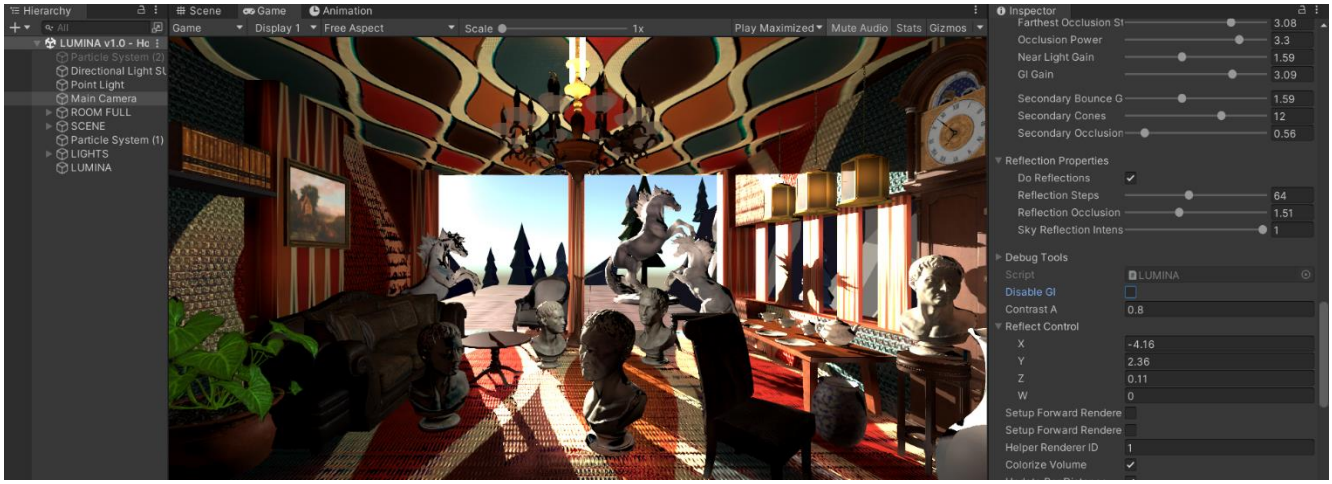
The image shows the GI added with a medium value in the Unity ambient intensity multiplier.



The following image shows the GI added with a zero value in the Unity ambient intensity multiplier.



11. Disable GI



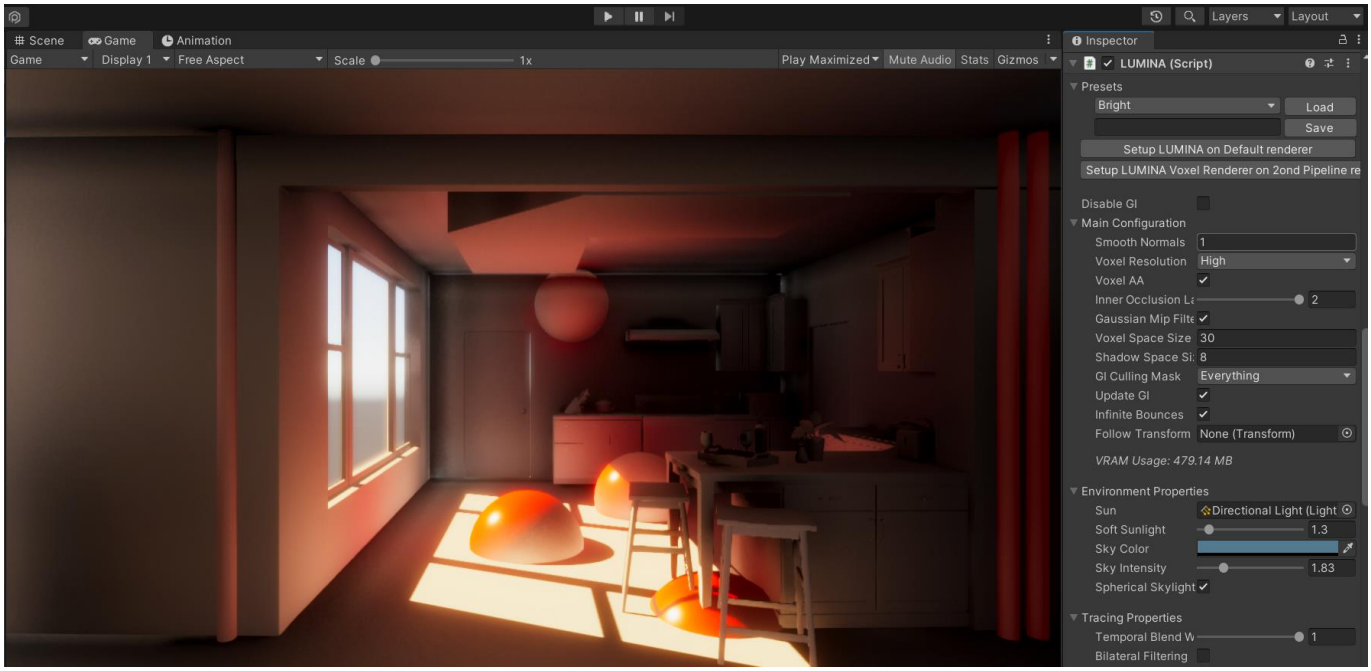
The system can be disabled by enabling the **“Disable GI”** checkbox. Also uncheck the “**Update GI”** to stop the voxelization of the scene, if a scene is small there is no need to keep the voxelizer active, and can be regulated using the “**Update GI” checkbox.**

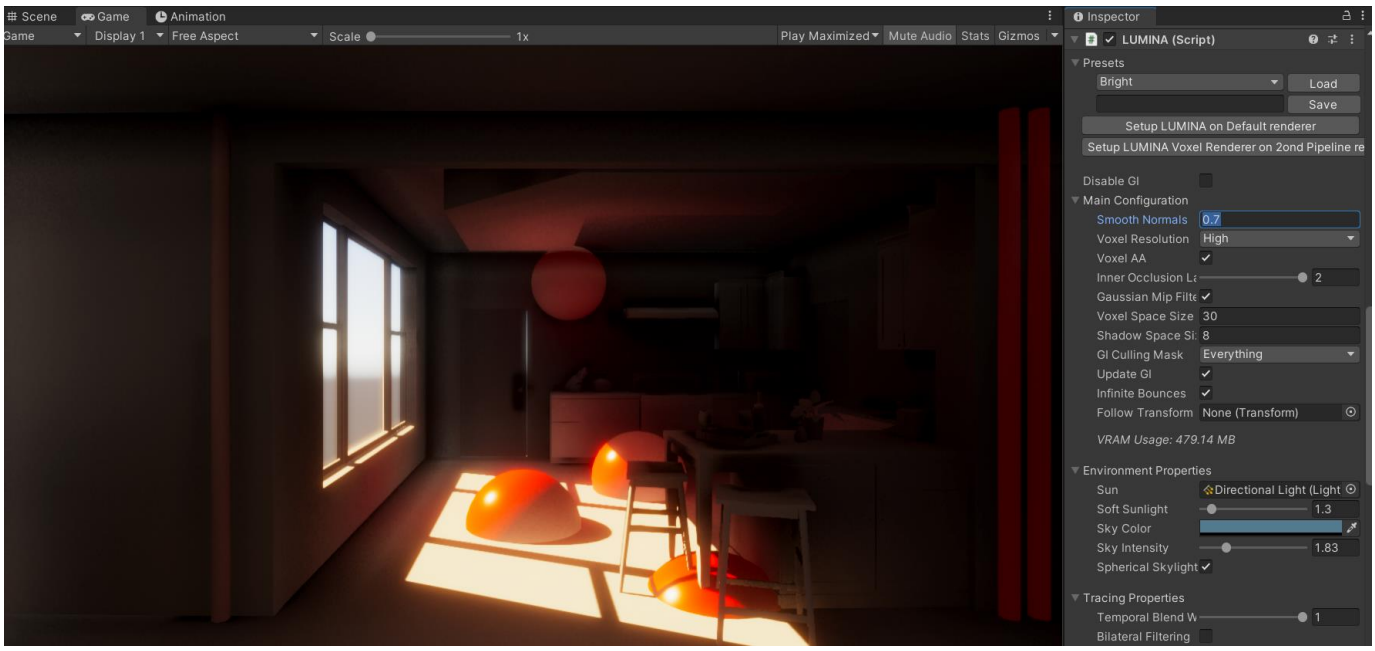
For Unity 6 LTS the best way to disable the effects in RenderGraph mode is to select a renderer without image effects in the camera. Also it is advised when working with the system in Unity editor to **add a renderer without any image effects in the pipeline and set it as Default**, such that the scene view camera does not use the LUMINA renderer.

12. Smooth normals new mode (v1.2)

The new variable should be set to one to configure the scene with the exact normals and can then be regulated further for artistic effects if needed.

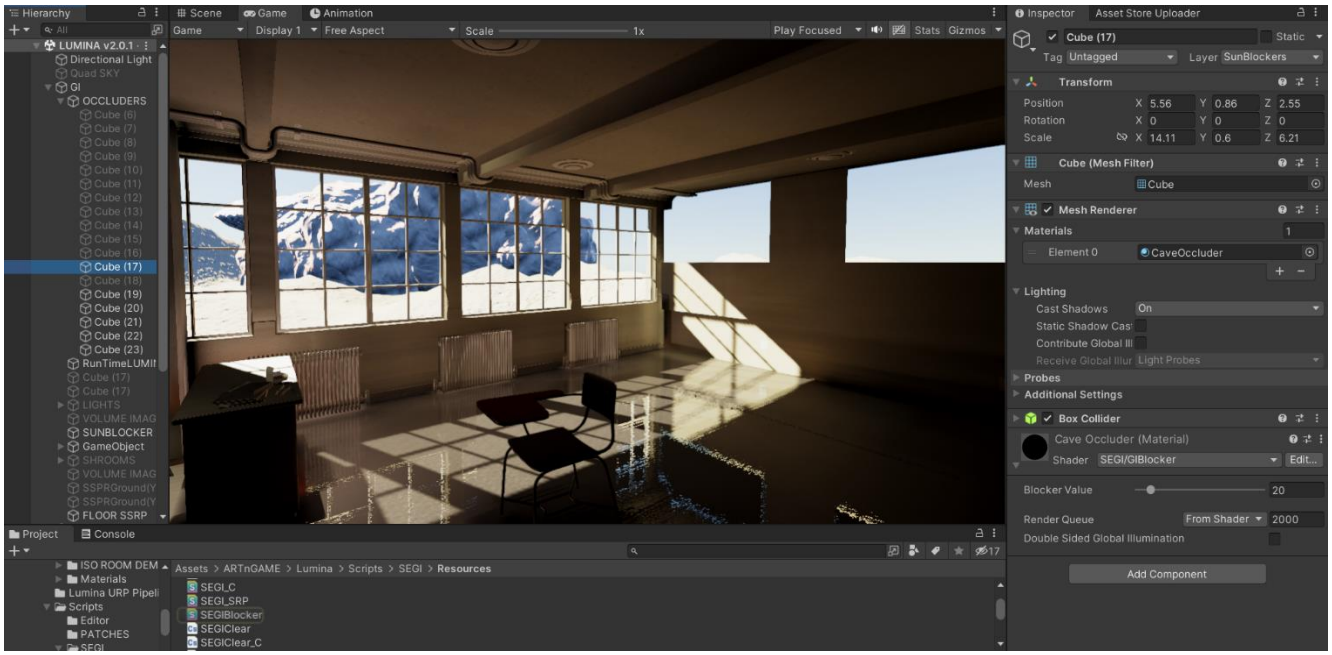
Below is shown a sample scene with exact normals in top image and with artistic shift of normals.





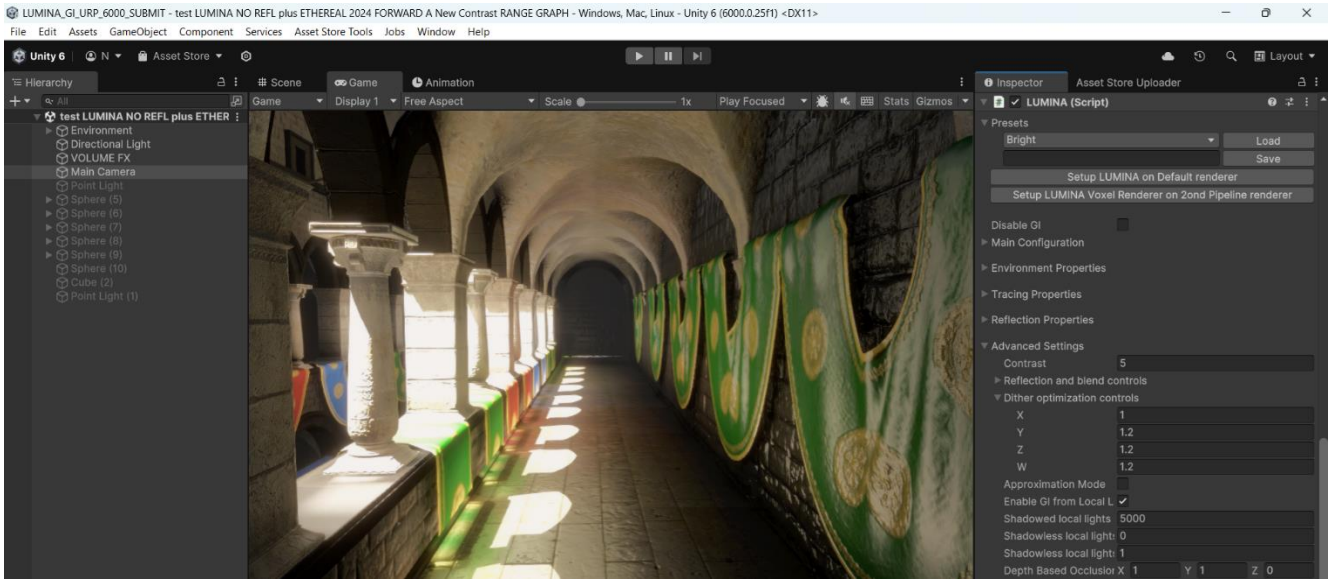
13. Light blocking methods

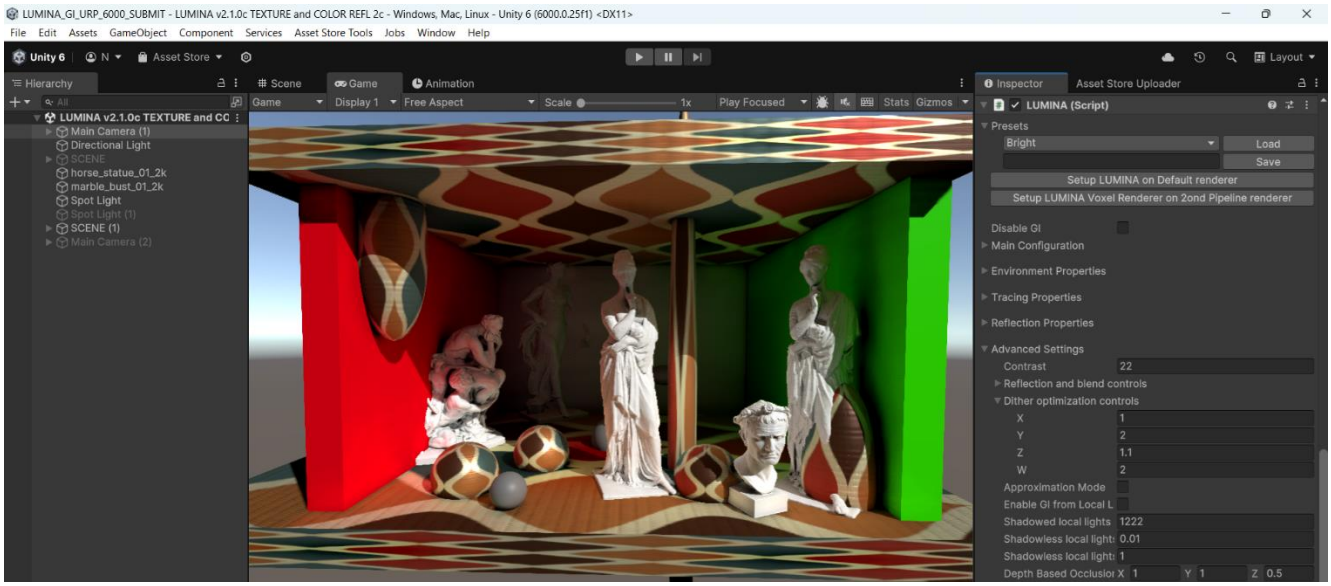
The system has a few methods to block light further than the meshes geometry. Meshes can be used with a material using a special shader that can block the GI light, so can create extra occusion and reduce light leaking. This material is not shown in the Main camera and is only visible in the GI solution. The shader is shown in the following image and can be set in a material and applied to any object that need to block the light and not be visible in the scene. Ideally those items should be added to a separate layer (e.g. Sunblocking) and the layer must be included in the LUMINA GI Layers.

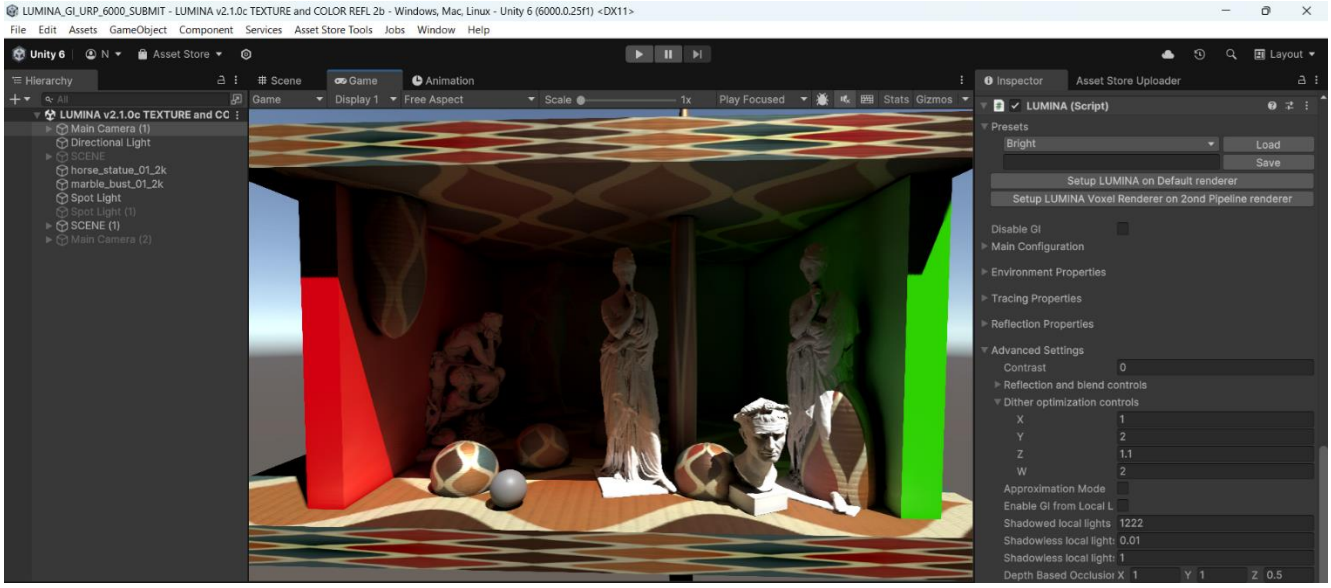


14. Contrast regulation

Below are some examples of the various options chosen in the Contrast modes and the relevant demo scenes to look at for the tweaking of the other LUMINA settings based on the Contrast value chosen.

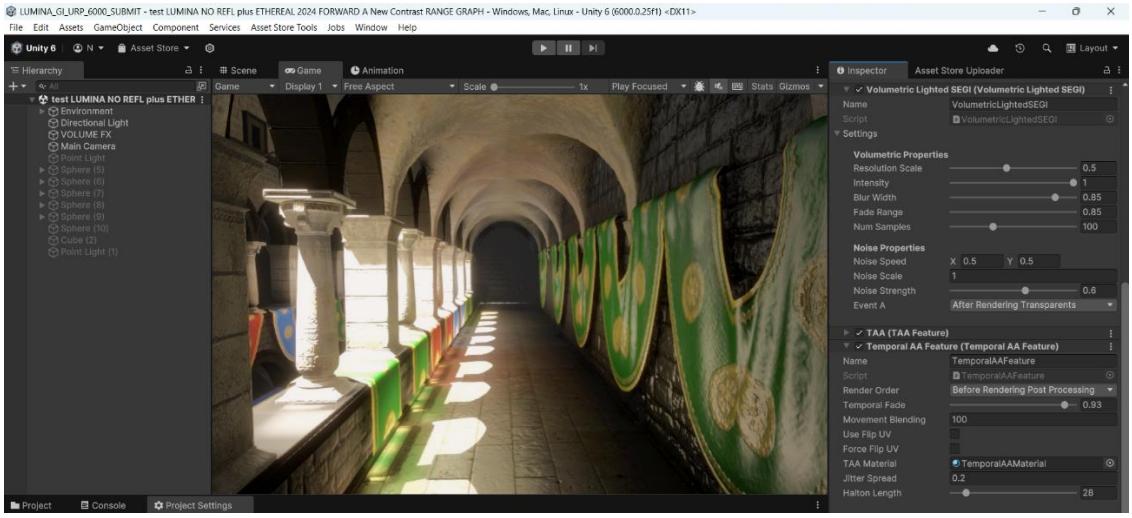




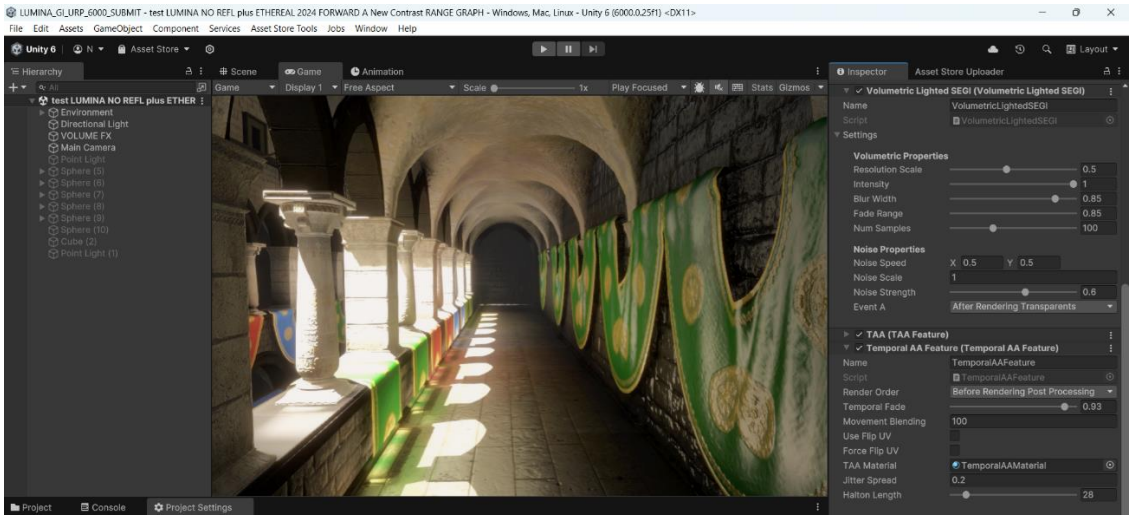


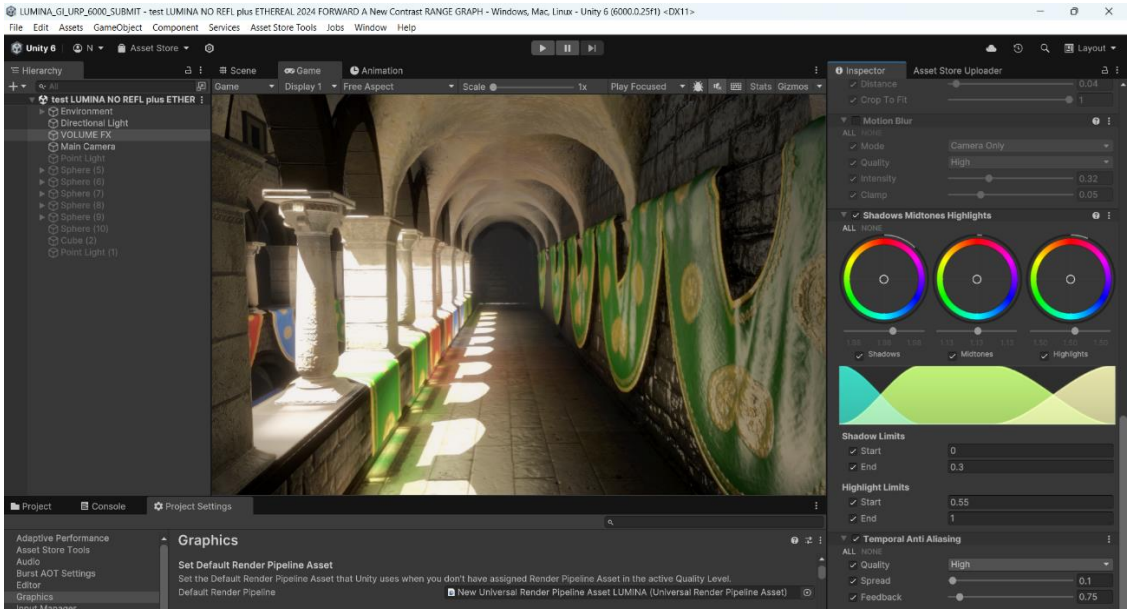
15. Temporal AA (Bonus feature for controllable Temporal AA)

The system includes two temporal AA implementations. **Temporal AA standalone** is implemented in “Temporal AA Feature” renderer feature, that can be added in the camera renderer as shown below.



Temporal AA in scene volume is implemented using the “TAA” renderer feature and must also have an entry in the scene volume, named “Temporal Anti Aliasing” to enable the effect.





16. Screen Space reflections (Extra feature for demo purposes)

The system includes the Lim screen space reflections module, that can be enabled by adding the “Lim SSR” renderer feature in the camera renderer. When used with the Pyramid provider renderer feature named “Depth Pyramid”, requires the renderer to be set to Deferred mode.