
SHINY
SHINY SSRR
Screen Space Raytraced Reflections for Universal Rendering Pipeline

USER GUIDE

Contents

Introduction

Thank you for purchasing!

Shiny SSRR brings Screen Space Raytraced Reflections to Universal Rendering Pipeline.

Reflections add a realistic touch to many surfaces. Use them wisely and in a subtle way to improve the quality of your scenes.

Setup

Requirements

- Unity 2019.4 LTS
- Universal Rendering Pipeline 7.5 or later (import the Universal RP package from the Package Manager).
- Windows / Mac desktop platforms.

Configuration

1) Make sure your project meets the above requirements.
2) Import Shiny SSRR asset into your project.
3) Enable Depth Texture option in the Universal Rendering Pipeline asset.
4) Select the Forward Renderer and add the “Shiny SSRR” render feature to the list.

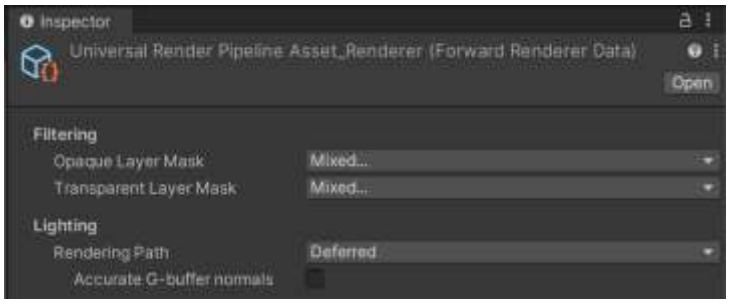
Demo Scene

If everything is properly setup, you should be able to see the reflections in the included demo scene.

How to add reflections to your scenes

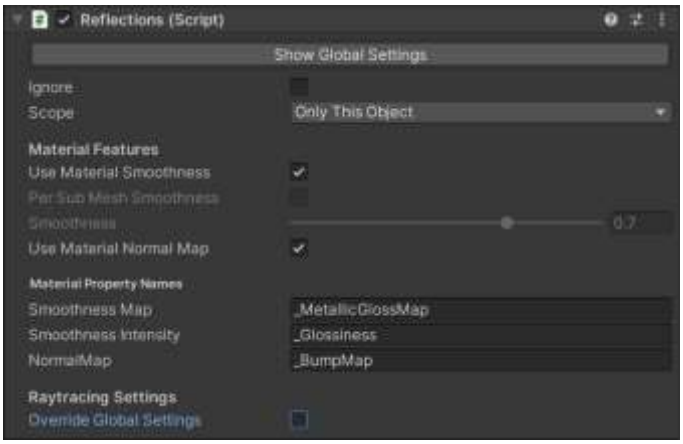
Depending on the URP version you have installed in your project, Shiny SSRR can be run in forward or deferred modes:

- In URP versions <12.0, this pipeline only supports forward rendering path. In this case, just jump to the next section (Forward rendering configuration).
- In URP 12.0 and up, the pipeline supports both forward and deferred. This rendering path can be configured from the renderer asset:



Forward rendering configuration

In this mode, add the Reflections script to any game object that you want to receive reflections. A single Reflections script can cover many game objects thanks to the “Scope” options shown in the inspector below:



When you add this script to the game object, it will start receiving reflections (assuming you have the Screen Space Reflections render feature correctly installed as described in the previous page).

The rest of options are:

- **Ignore**: enable this option to exclude this game object from getting any reflection. This is useful if you assign the Reflections script to the root of a group of objects, but you don’t want reflections on some of them.
- **Scope**: in addition to add reflections to this object, you can specify that reflections should also be added to the children of the object. And you can also specify some filters, like the layer, a name pattern or specific sub-meshes.

- **Use Material Smoothness**: if the game object uses a PBR material with _Smoothness property, the asset can use that value so changing the smoothness of the material will affect the reflections as well.
- **Per Sub Mesh Smoothness**: let you specify a different smoothness value per submesh.
- **Smothness**: if the "Use Material Smoothness" is not enabled or the material doesn't have a smoothness property, this value will be used instead.
- **Use Material Normal Map**: if the game object uses a material with a normal map, use it. If the material has no normal texture, this option will be ignored.
- **Material Property Names**: this are the default names for usual smoothness map textures, normal map and smoothness intensity values used in standard shader. Feel free to specify custom shader property names if required. The smoothness value is encoded in the alpha channel of the smoothness map so make sure that texture preserves the alpha value in Import settings.
- **Override Global Settings**: let you use different raytracing settings only for this object.

Hint: in forward rendering mode, you can also customize the reflection intensity modifying the alpha value of the color of that material.

Why is necessary to add a Reflections script on the objects?

Because forward renderer doesn't store (by default) normals and smoothness into a dedicated buffer which are required to compute reflections. The asset needs to compute the normal per object in order to produce correct reflections. In URP 12, this pipeline gets deferred support, and Shiny SSRR can leverage the g-buffers to automate the effect on the entire scene without requiring you to add the Reflections scripts.

However, there's a bonus for using this method: by adding and customizing Reflections per group of objects you have total control of how the effect renders on the screen. In order to make this process faster, there're global settings that are automatically applied to all reflections (see next section).

Also, because you're specifying which objects will require reflections, the overall result can render faster because the system doesn't need to process the entire screen – only those pixels of the objects that will receive reflections.

Deferred rendering configuration

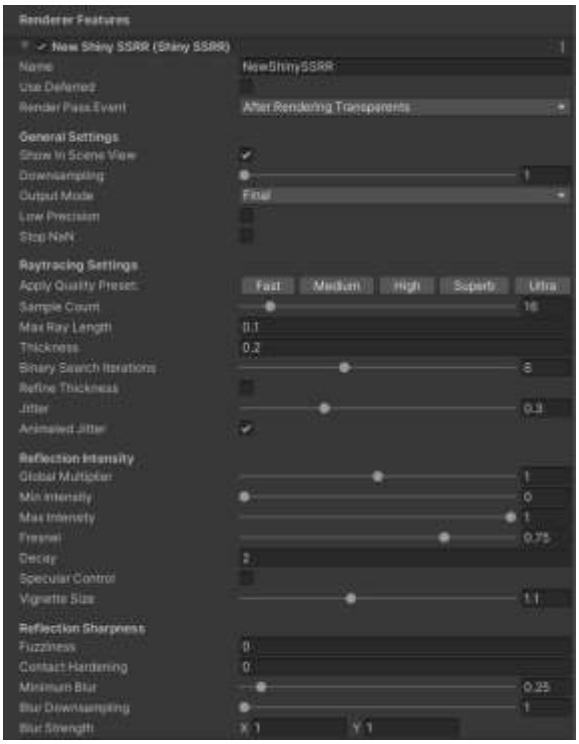
If you use URP 12.0 or later, you can switch to deferred rendering path in the renderer asset. Then enable the checkbox "Use Deferred" in the Shiny SSRR render feature (see global settings in the next section).

Benefits of deferred include a unified approach to reflections and more physically based results since smoothness is read per pixel. Also, you don't need to add the Reflections script to the objects that will receive reflections.

On the other hand, you lose a bit of flexibility since you can no longer customize the raytracing settings per object but globally. This can be tricky as sometimes you prefer to use less or more samples or different decay or jitter settings for example. Also, because deferred applies to entire screen, it can be a bit more expensive.

Global settings

The global settings are located in the Shiny SSRR render feature itself and provide options that affect all reflections. These settings are shown in the screenshot below:



General settings

- **Show In Scene View**: this option allows you to disable reflections while working in the SceneView.
- **Downsampling**: applies the effect to a reduced size image buffer improving performance in exchange of accuracy/quality. Depth Bias is used to clip reflections by depth buffer so objects on top of others do not get reflections. The Depth Bias parameter is used only when downsampling is greater than 1 and only if camera is rendering in forward rendering path.
- **Output Mode**: let you debug the result or show a side-by-side comparison between the original image and the image with reflections.
- **Low Precision**: uses LDR buffer (ARGB32) instead of HDR (ARGBhalf) buffer for reflection buffers.

Raytracing settings

- **Sample Count**: the number of steps used in the raytracer. The more samples, the more quality and extensive the reflections result.
- **Max Ray Length**: the maximum distance of reflections.
- **Thickness**: when the ray passes behind an object, it needs to figure if it's hitting the object or passing behind it. Since the back faces of objects are not rendered, the asset uses a fixed value. Increase this value to ensure all reflections are captured.
- **Binary Search Iterations**: increase to refine the hit position detected by the raytracer. Usually a value of 6 or 8 is enough in most situations.
- **Thickness Fine**: when this option is enabled, after the binary search has refined the reflection hit position, the algorithm will discard the reflection hit if the distance from the ray to the surface is greater than this value. Therefore, this is the actual thickness value used after binary search. Keep it as low as possible as long as reflections look bright and sharp.

- **Jitter**: reduces banding artefacts by adding a random offset to the starting position of the rays.
- **Animated jitter**: this option changes the jitter amount per pixel every frame redistributing the noise.

Reflection Intensity

- **Global Multiplier**: this is a global multiplier to all reflections intensity and let you tune down the effect globally.
- **Min Intensity**: determines the minimum reflection intensity on all surfaces. Can be used to force a very shiny environment.
- **Max Intensity**: caps the maximum reflection intensity on any surface.
- **Fresnel**: this setting reduces the reflection intensity based on the view angle to the surface. A value of 1 gives more realistic results.
- **Decay**: this setting reduces the reflection intensity based on the distance to the reflected point.
- **Specular Control**: reduces intensity of specular in the reflections. This will produce less flickering/shimmering in the scene caused by certain materials.
- **Vignette Size**: reflections are tuned down around the screen borders. This setting controls the intensity of this reduction.

Reflection Sharpness

- **Fuzzyness**: a blur multiplier which is also based on the distance to the reflected point so distant reflections look blurrier than contact reflections.
- **Contact Hardening**: increases the sharpness of contact (short distance) reflections.
- **Minimum Blur**: defines the minimum blur intensity for all reflections.
- **Blur Downsampling**: size of the blur buffers. Increasing only if you want to improve performance a bit.
- **Blur Strength**: let you control the blur spread along the x and y axis.

Support

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