Corgi SDF Outlines

ReadMe v1.3.1

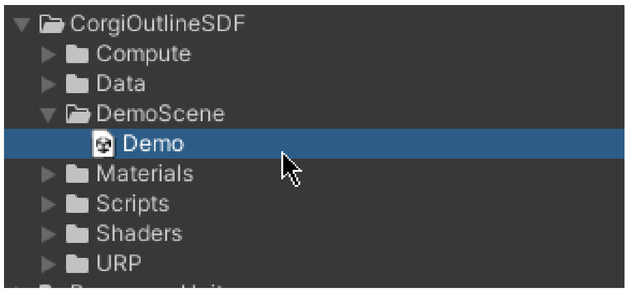
(latest documentation here)

Thanks for purchasing Corgi SDF Outlines! This document will help you get started and explain how everything works.

Getting Started

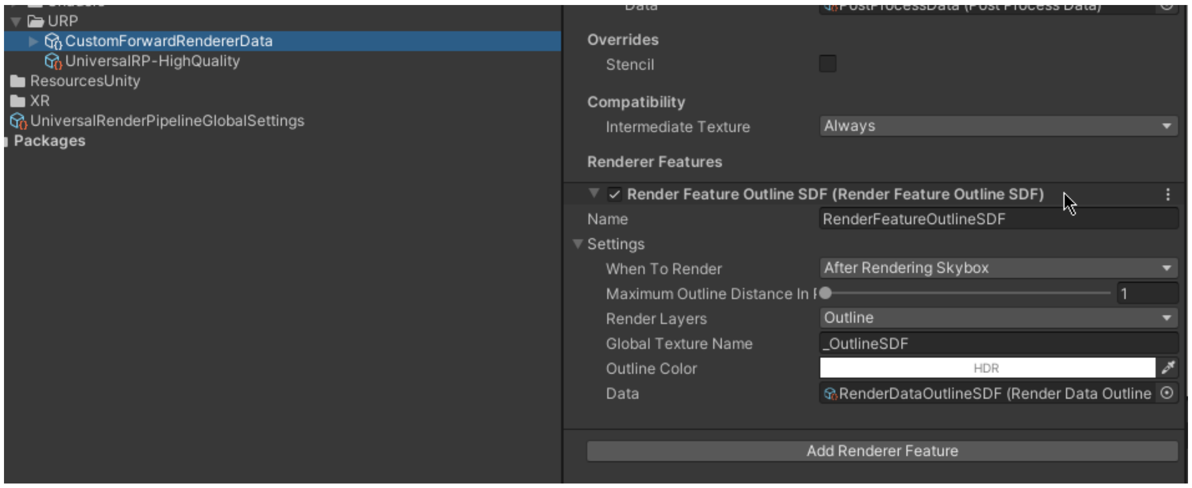
Corgi SDF Outlines requires the Universal Render Pipeline (URP) to be in use. If you do not have it setup in your project, this plugin will not work for you. To try out the demo scene, you'll need it installed (Window -> Package Manager -> Unity Registry -> Universal Render Pipeline).

The first thing you should do is navigate to the Demo scene. It's located in `CorgiOutlineSDF/DemoScene/Demo.scene`.



```
CorgiOutlineSDF
  Compute
  Data
  DemoScene
    Demo
  Materials
  Scripts
  Shaders
  URP
```

Once you are there, either assign the demo `RendererData` to your Graphics settings or simply add the `RenderFeatureOutlineSDF` to your current `RendererData`.



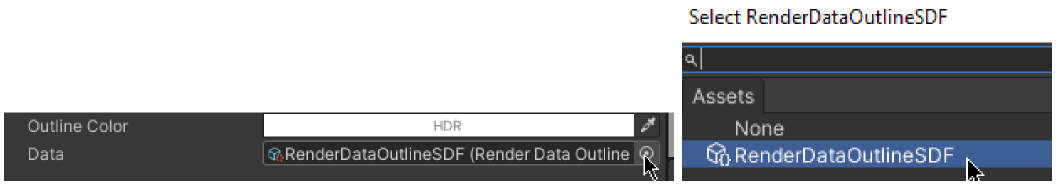
```
URP
  CustomForwardRendererData
  UniversalRP-HighQuality
ResourcesUnity
XR
UniversalRenderPipelineGlobalSettings
Packages
```

Overrides
Stencil

Compatibility
Intermediate Texture: Always

Renderer Features
✓ Render Feature Outline SDF (Render Feature Outline SDF)
Name: RenderFeatureOutlineSDF
Settings
When To Render: After Rendering Skybox
Maximum Outline Distance In I: 1
Render Layers: Outline
Global Texture Name: _OutlineSDF
Outline Color: HDR
Data: RenderDataOutlineSDF (Render Data Outline

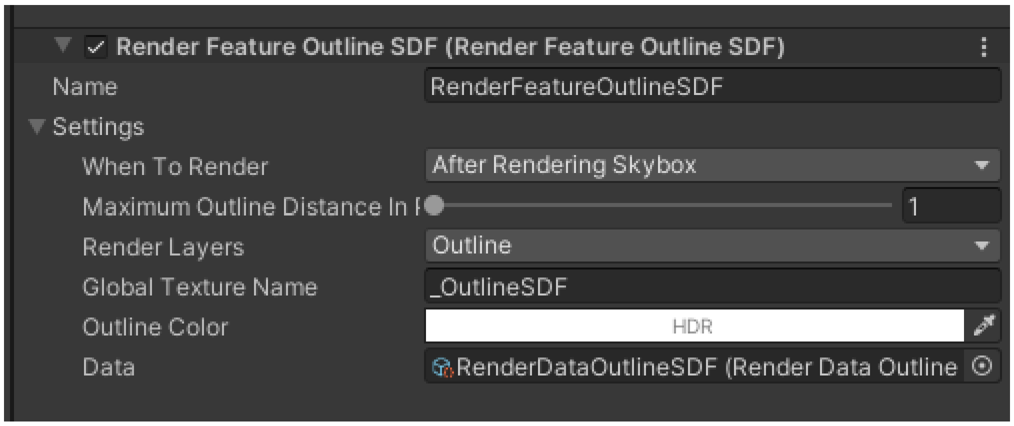
If assigning a new `RenderFeature`, you'll need to drop in the `Data` block. Just hit the circle, and there should only be one option.



Outline Color: HDR
Data: RenderDataOutlineSDF (Render Data Outline

Configurable Options

Now that you're set up, here are what the options do.

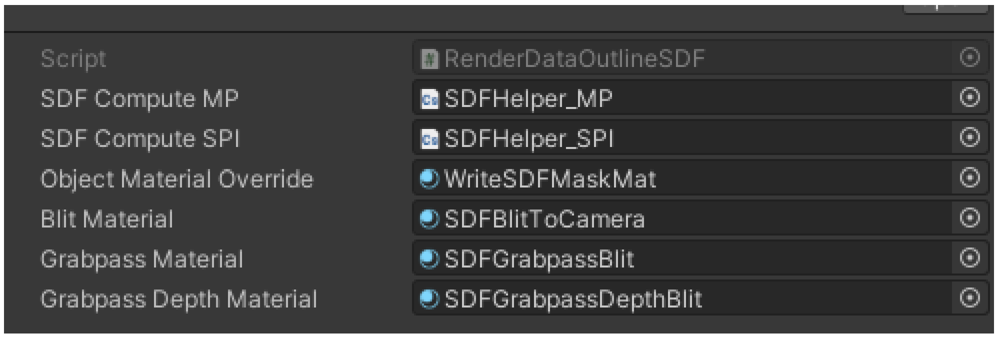


Render Feature Outline SDF (Render Feature Outline SDF)

Name	RenderFeatureOutlineSDF
Settings	
When To Render	After Rendering Skybox
Maximum Outline Distance In F	1
Render Layers	Outline
Global Texture Name	_OutlineSDF
Outline Color	HDR
Data	RenderDataOutlineSDF (Render Data Outline

- **WhenToRender -** This is where the RenderFeature will try to inject itself into the render pipeline. I recommend setting this to AfterRenderingSkybox, but you may have other reasons for changing this, so feel free.
- **MaximumOutllineDistanceInPixels -** The outline will try to reach this number of pixels around objects. Please keep in mind that performance scales with this value. 1 being the cheapest, 16 being the most expensive. If you do not need thicc outlines, I recommend leaving this at 1.
- **RenderLayers -** This is the Unity LayerMask that is used for rendering outlines. You can use any Layer you want. The demo uses "Outline" by default.
- **OutlineColor -** This is an HDR Color value for outlines. Alpha is used for lerping between the outline and the color under it. MAKE SURE YOUR ALPHA IS NOT ZERO! A lot of people make this mistake!
- **Data -** This is the data block used by the effect. It's a convenient scriptable object so you do not need to go looking for random references as a user of the plugin.

RenderDataOutlineSDF



Script	RenderDataOutlineSDF
SDF Compute MP	SDFHelper_MP
SDF Compute SPI	SDFHelper_SPI
Object Material Override	WriteSDFMaskMat
Blit Material	SDFBlitToCamera
Grabpass Material	SDFGrabpassBlit
Grabpass Depth Material	SDFGrabpassDepthBlit

This is the Data block in the RenderFeature. You can mostly ignore this, unless you really want to dive deep to customize how the outline behaves.

- **SDFComputeMP** / **SDFComputeSPI** - These are the compute shaders for the step process of the outline effect. It's split into MP (multi pass) and SPI (single pass instanced) because compute shaders do not currently support keywords and multi compiles.
- **ObjectMaterialOverride** is using the WriteMask shader, for the initial draw of objects from the outline layer into the outline mask texture.
- **BlitMaterial** is used for blitting the outline to the game texture.
- **Grabpass**[**Depth**]**Material** used for copying the game texture to an intermediate render texture so we can sample it.
-

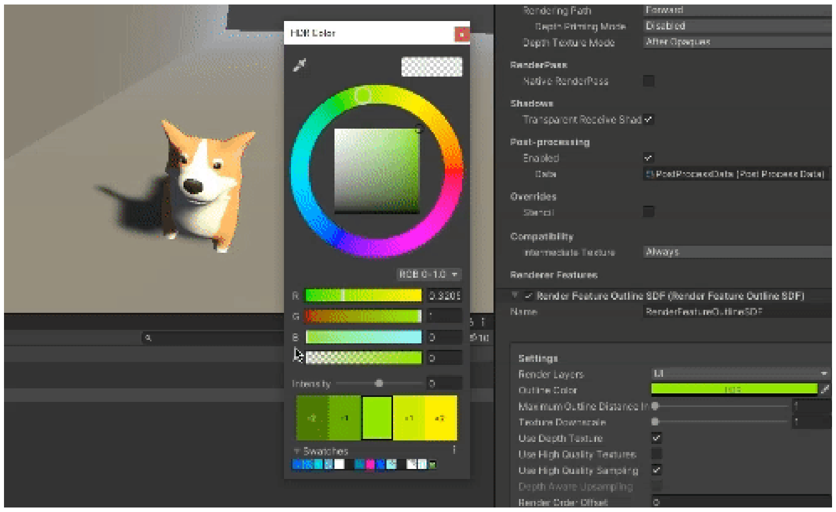
Common Problems

I’ve gotten a few emails since the release of the plugin. If the stuff written here doesn’t help resolve your issues, please contact me (contact info at the bottom of this document). I’ll be happy to help work through any issues.

“Help, no outline is showing!”

Of course, this is the most frequent kind of email.

- Ensure that the layer of the object you want to outline matches the outline RenderFeature’s set layer. If these layers do not match, nothing will be rendered in the outline buffer.
- If you’re building for a platform which does not support Compute Shaders, you may see a bunch of errors in the logs or even incorrect graphics on your device. This plugin requires Compute Shaders, so double check Unity’s documentation on whether or not the platform you’re targeting supports them!
- Double check the outline color. Very often, people will forget to assign an alpha. It defaults to zero - which means no visible outline. Set it to one!



R 0.3205
G 1
B 0
A 0
Intensity 0

Rendering Path: Forward
Depth Priming Mode: Disabled
Depth Texture Mode: After Opaques
RenderPass
Native RenderPass
Shadows
Transparent Receive Shad ✓
Post-processing
Enabled ✓
Data: PostProcessData (Post Process Data)
Overrides
Stencil
Compatibility
Intermediate Texture: Always
Renderer Features
✓ Render Feature Outline SDF (Render Feature Outline SDF)
Name: RenderFeatureOutlineSDF
Settings
Render Layers: UI
Outline Color: HDR
Maximum Outline Distance In: 1
Texture Downscale: 1
Use Depth Texture ✓
Use High Quality Textures
Use High Quality Sampling ✓
Depth Aware Upsampling
Render Order Offset: 0

How does this work?

This effect is otherwise known as a Stroke outline, or an SDF outline.

First, a mask is drawn to a render texture.
Compute shaders are then used to generate a distance field from any blank space to the nearest masked pixel. This is done by doing the following:

- Generate a uv from 0 to 1 in an SDF texture.
- For the thiccness of the outline, do the following in a loop:
    - For each pixel, check its neighbor. If it’s near a masked pixel, remember it’s uv and distance.
- Use a normal vert/frag shader to sample this SDF texture into an outline texture.
- Blit the outline texture to the screen.

If you need a deeper dive into this, check out the wiki page:
https://en.wikipedia.org/wiki/Jump_flooding_algorithm

Contact Me!

If anything is unclear or even if you just want to give feedback or suggestions: please contact me!

- You can email me at coty@wanderingcorgi.com
- Or you can hit me up on Discord (Coty#2845).
- There's also a support discord here: https://discord.gg/n23MtuE