

Keyhole Overlay Post Processing

By Joseph Y.

Overview

The keyhole overlay post-processing effect is a simple overlay post-processing that cutouts the black part of the texture and overlays it to the screen. It is designed to run on the Unity URP pipeline and Unity 6000.0.25f1. This effect utilizes URP's render feature to integrate it into the render pipeline and ShaderGraph to create the shader effect. This post processing is very similar to the transition post processing with modifications to only act as a simple overlay with color keying.

Since its similar to the transition effect, you can find some of the code to be based on Dan Moran's Shaders Case Study—Pokémon Battle Transitions

(<https://youtu.be/LnAoD7hgDxw?si=tCtTEOshaZdfLi6R>).

Installation

Before you begin:

1. Ensure you use the Universal Render Pipeline (URP) and Unity 6000.0.25f1.
2. Ensure you have installed the necessary packages, such as Post-Processing.
3. Ensure that you are using forward rendering in your universal renderer.
4. Ensure that the color space is set to linear in the project settings.
5. Ensure that under the project settings > graphics > Render Graph, you enable Compatibility Mode on (meaning you have Render Graph Disabled).

The effect is contained in a Unity package you can import into Unity. To do this, navigate to Assets > Import Packages, select the Unity package, and open it. A popup will appear showing the contents of the package. Choose to import everything from the package and click OK. They will be imported into the Unity project under the Keyhole folder.

After importing, you should see several specific items, including a folder named "textures" containing keyhole textures, 3 C# scripts (1 for the render feature, 1 for the render pass, and another for the volume component), 1 scene file containing an example level of the effect as a UI and post processing implementation, and the KeyholeOverlay shader.

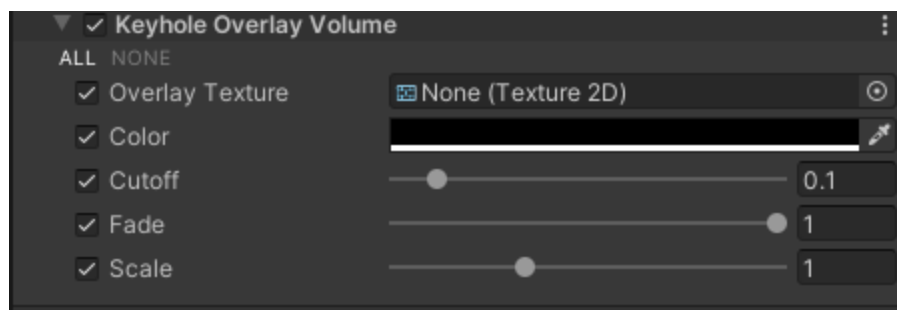
To apply the effect:

1. Navigate to the universal renderer being used in your project and locate the "Add Render Feature" button.

2. Click the "Add Render Feature" button and choose the keyhole overlay render feature.
3. Create a game object or go to the object containing your post-processing volume/volume.
4. Under the volume component, click on **Add Override** and select Keyhole Overlay (or overlay). If done correctly, the volume component should be updated with a new volume override that says Keyhole Overlay Volume.

You can adjust the render pass event by navigating to the Transition render feature in the universal renderer. Click on the dropdown beside the setting and change the event to your preference, such as "after opaques."

Using the effect



You can go to the volume game object with the transition effect to adjust it. You can use one of the provided textures in the textures folder or a custom one (refer to [Overlay Textures/Creating and Using custom textures](#)). Here are the details about the parameters:

1. **Overlay Texture:** The field containing our texture for the keyhole overlay. For more info, refer to [Overlay Textures/Creating and Using custom textures](#).
2. **Color:** This parameter affects the color at the end of the transition. Defaulted to black.
3. **Cutoff:** The parameter that affects the transition itself. 0 = the start of the transition and 1 = the end of the transition. To animate the effect, you will need to call the volume in a script and get the transition effect. Once you do so, you can lerp it from one value to another. Defaulted to 0.
4. **Fade:** Affects the color's alpha at the end of the transition. 1 = Opaque and 0 = Invisible. Defaulted to 1.
5. **Scale:** Affects how big or small the keyhole/texture is. Values over 1 means it will be larger while Values under 1 means it will be smaller.

Overlay Textures/Creating and Using custom textures

The post-processing includes a folder with some textures. These textures consist of two parts, one for UI purposes and one for the post processing shader. Similar to the transition post processing, it uses grayscale to figure out what to key out. In this case, the keyhole part is white while the background is black. The effect will key out the white part, leaving the black parts alone.

When you import the texture to your Unity project, there are certain settings to consider. One crucial setting to change is the wrap mode, which should be set to "clamp" instead of "repeat." If left on repeat, artifacts may appear on the edges of the screen during transitions.



If you look closely at the edges, you can see some gaps. It looks like this if you left it on repeat in the wrap mode. It can be avoided if you change it to clamp in wrap mode.

If you want smooth transition edges, use the trilinear or bilinear filter mode. For a slightly pixelated look, set the filter mode to point.



Point Filter Mode



Trilinear Filter Mode

Depending on the texture, you can adjust the texture compression level from none to high quality. This will impact the compression of the image in the build and project, as well as how it looks. Thus, It's important to be mindful of this. If you're unsure, it's best to keep the

compression at none, especially if the texture is already small enough to prevent it from becoming a significant concern.

You don't have to follow this strictly as different textures may have different needs or settings to be optimal.

You may need to use application software like Photoshop to create the textures. It is recommended to use black and white rather than using colors.

Update Log

V0.9: The base implementation of the post processing.

Credits

Some of the shader code is based from Dan Moran's Shaders Case Study—Pokémon Battle Transitions (<https://youtu.be/LnAoD7hgDxw?si=tCtTEOshaZdfLi6R>) YouTube video.

Example Screenshots



What it looks like to overlay the keyhole texture