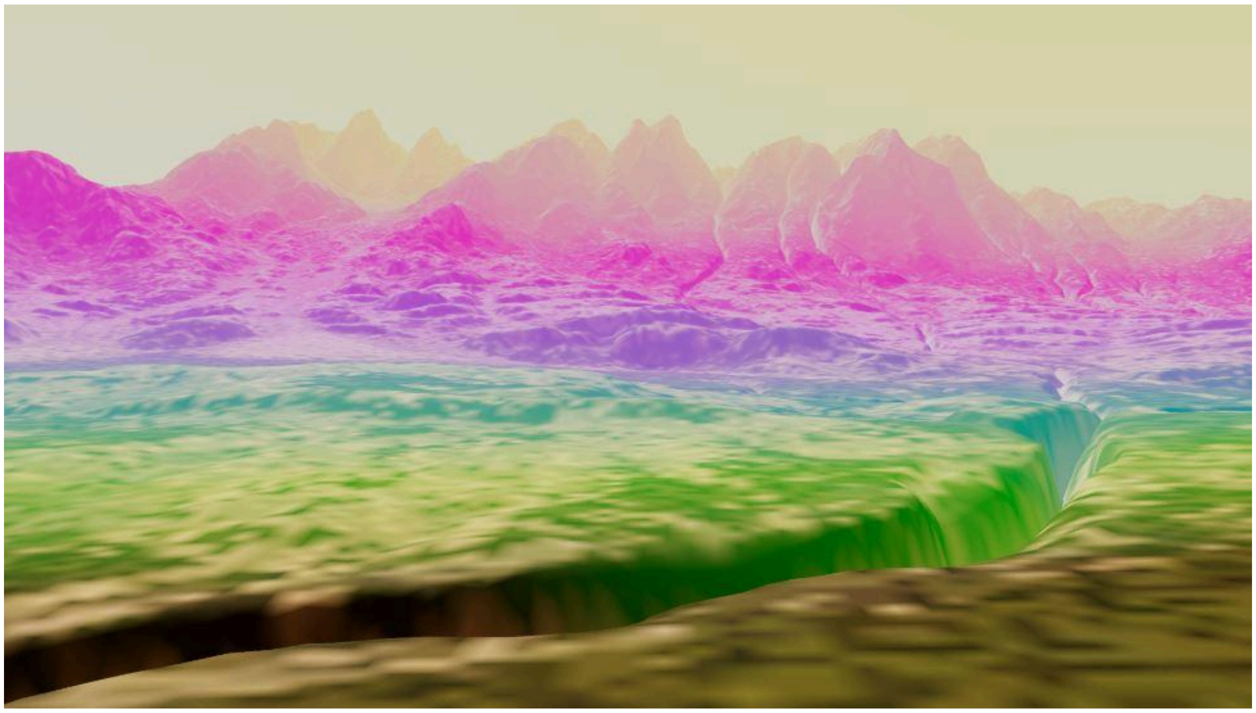


# Stylized Fog Post Processing

By Joseph Y.

## Overview

The stylized fog effect is a multi colored/gradient based fog post-processing that uses multiple colors or a gradient to create a fog like effect to the screen. It is based on Firewatch's gradient based stylized fog. This effect allows the use of 5 separate colors for each distance of the fog or a gradient (as a texture) for the whole fog. It is designed to run on the Unity URP pipeline and Unity 6, but theoretically, it could run on Unity 2022 but it is untested. This effect utilizes URP's render feature to integrate it into the render pipeline and ShaderGraph to create the shader effect.

The way the fog works is that we have 5 separate fog effects for each relative distance from closest to furthest based on the camera's depth texture, and then overlay each of the effects on top of each other from the order of closest to furthest to create the effect. Each fog effect has its own color and distance at which it is applied to. We include in betweens (close-mid and mid-far) to help make the fog look nicer and have better control over it. Since we have 5 effects for 5 common points of a gradient (beginning, beginning-midpoint in between, midpoint, midpoint-endpoint in between, and endpoint), the fog behaves as if it is multicolored or gradient based. The effect also has the capability to run it on a gradient rather than 5 separate colors, which you can enable or disable if you want.

---

## Installation

Before you begin:

1. Ensure you use the Universal Render Pipeline (URP) and Unity 6 (6000.0.25f1).
2. Ensure you have installed the necessary packages, such as Shader Graph.
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).

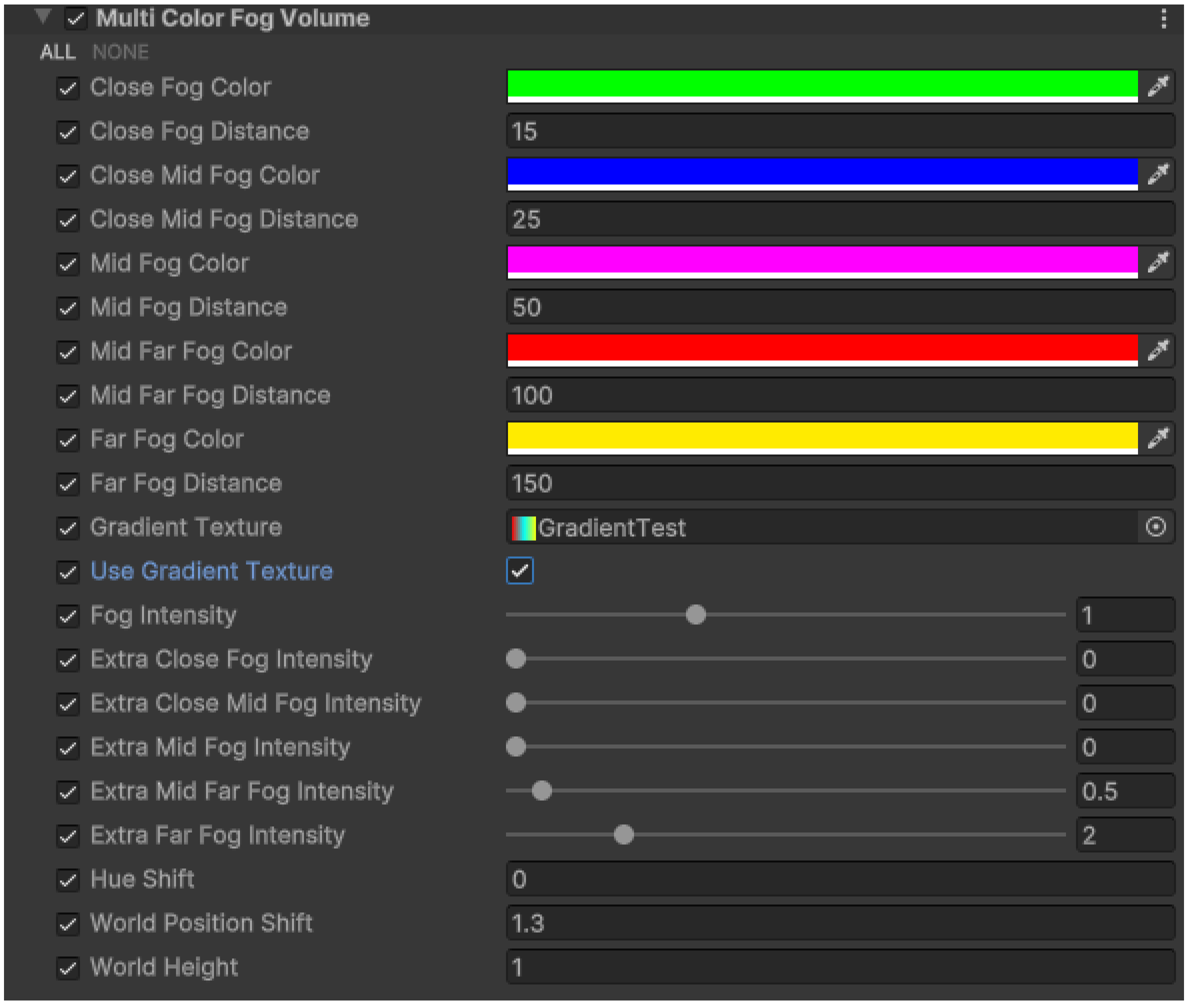
The effect is contained in a Unity package you can import into Unity. To do this, navigate to Assets > Import Packages, select the Unity package, and open it. A popup will appear showing the contents of the package. Choose to import everything from the package and click OK. They will be imported into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

After importing, you should see several specific items, including a gradient texture for testing and demonstration, 3 C# scripts (1 for the render feature, 1 for the render pass, and another for the volume component), the Multicolor Fog shadergraph, 2 sub graphs for the shader graph, and a material with the shader graph on it.
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 Multicolor Fog render feature.
3. Add the Mat_MulticolorFog material to the render feature under the material input field within the Multicolor Fog Render feature’s setting.
4. Create a game object or go to the object containing your post-processing volume/volume.
5. Under the volume component, click on **Add Override** and select Multicolor fog. If done correctly, the volume component should be updated with a new volume override that says Multicolor fog Volume. You can also use the material to adjust parameters if you are not using the volume component.

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

## Using the effect



✓ Multi Color Fog Volume

ALL NONE

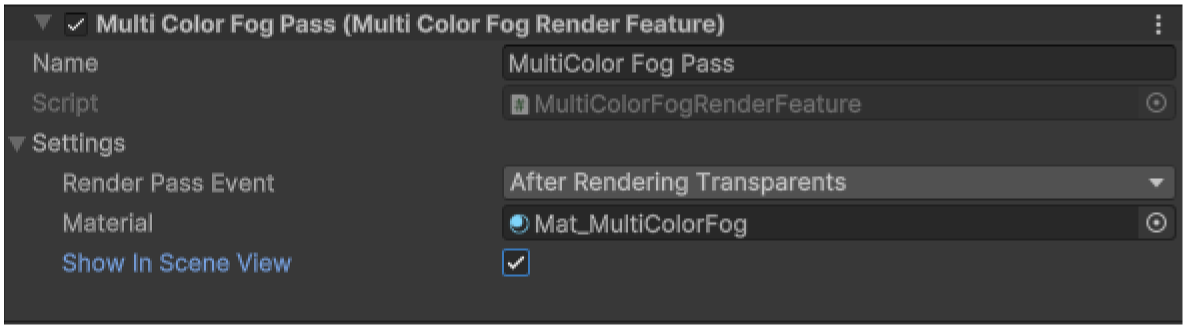
| | |
|---|---|
| ✓ Close Fog Color | 🟩 |
| ✓ Close Fog Distance | 15 |
| ✓ Close Mid Fog Color | 🟦 |
| ✓ Close Mid Fog Distance | 25 |
| ✓ Mid Fog Color | 🟪 |
| ✓ Mid Fog Distance | 50 |
| ✓ Mid Far Fog Color | 🟥 |
| ✓ Mid Far Fog Distance | 100 |
| ✓ Far Fog Color | 🟨 |
| ✓ Far Fog Distance | 150 |
| ✓ Gradient Texture | GradientTest |
| ✓ Use Gradient Texture | ✓ |
| ✓ Fog Intensity | 1 |
| ✓ Extra Close Fog Intensity | 0 |
| ✓ Extra Close Mid Fog Intensity | 0 |
| ✓ Extra Mid Fog Intensity | 0 |
| ✓ Extra Mid Far Fog Intensity | 0.5 |
| ✓ Extra Far Fog Intensity | 2 |
| ✓ Hue Shift | 0 |
| ✓ World Position Shift | 1.3 |
| ✓ World Height | 1 |

You can go to the volume game object with the Multicolor Fog effect to adjust it. You can either use the 5 separate color fields or use a gradient texture for the fog colors. You can use the provided texture or a custom one (refer to Creating and Using custom textures). Here are the details about the parameters:

1. **Fog Color [Close, Close-Mid, Mid, Mid-Far, Far]:** These are color fields for each fog effect part. For example, close fog color field affects the fog color of the fog closest to you. If the color is transparent (aka its alpha set to 0), it will not render as the effect also takes the color's alpha into account.
2. **Fog Distance [Close, Close-Mid, Mid, Mid-Far, Far]:** Affects at what distance each of the fog parts are applied.
3. **Use Gradient Texture:** Use the gradient texture instead of the 5 separate colors. By default, it is turned off.

4. **Gradient Texture:** The texture containing our gradient. Refer to Creating and Using custom textures for more info.
5. **Fog intensity:** Affects how intensive is the fog overall.
6. **Extra Fog intensity [Close, Close-Mid, Mid, Mid-Far, Far]:** Affects how intensive is the fog at each part of the fog. For example, extra close fog intensity affects the fog intensity of the fog closest to you.
7. **Hue Shift:** shifts the color of the fog by a based amount
8. **World Position Shift & World Height:** Affects the world position and height of the fog effect to add in a small amount of height fog and to take account of the height. World Height is dependent on the world position which can be shifted by the world position shift. Default values are good enough for general use.

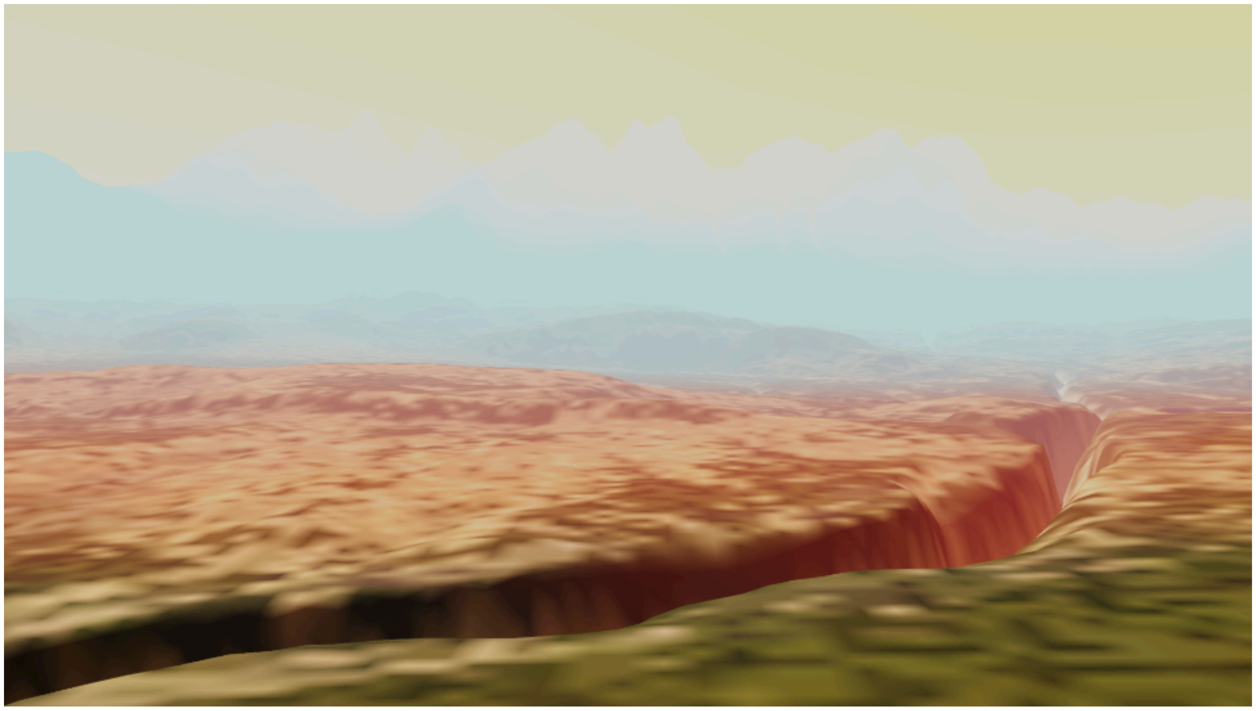
The effect doesn't affect the scene view's camera as it can get very annoying during development, so by default, it is turned off. You can enable it by toggling the Show In Scene View button on in the effect's render feature under its settings, as shown below.



Multi Color Fog Pass (Multi Color Fog Render Feature)

| | |
|---|---|
| Name | MultiColor Fog Pass |
| Script | MultiColorFogRenderFeature |
| Settings | |
| Render Pass Event | After Rendering Transparents |
| Material | Mat_MultiColorFog |
| Show In Scene View | ✓ |

Do note that there are some colors that may not work the best when it's blending with the skybox or overall at further distances, such as in the case with the far fog part. Increasing its fog intensity does help make it better only in some cases. But another way to help it blend better is to use it in combination with Unity's BuiltIn fog effect, allowing you to blend with the skybox better and be more similar to the look from Firewatch. You can find Unity's BuiltIn fog effect by going to the Lighting window > Environment > Other Settings section, which will contain an option for fog. In some templates, it will be turned on by default. Enabling it will look like this (see below).

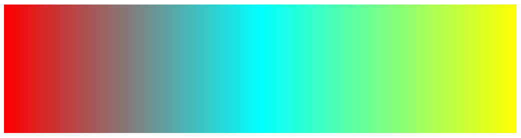


**The fog effect with Unity's BuiltIn fog effect on.**

You can play around with the fog setting to see which best suits you but typically, it is best to use a neutral color for the Unity fog to avoid influencing our fog effect too much to a particular color if you want your fog to be in a particular way.
But to fully showcase the effect, the screenshots that you see in the examples and at the top are all taken without Unity's BuiltIn Fog and at its default parameters.

## Creating and Using custom textures

The post-processing includes a gradient texture for testing and demonstration purposes. Our effect grabs the texture and uses its UV coords to obtain the colors in the gradient. To ensure that it gets the colors properly, the gradient should be on the x-axis, aka horizontally.



As mentioned, our effect uses the starting point, end point, mid point, and the inbetweens. So it is best to create gradients around those points. But due to a weird quirk, the point of the gradient is shifted by 0.01 to obtain the correct color, meaning for an example, our starting point is at 0.01 rather than 0 and our end point is at 0.99 rather than 1. This should not affect you when you create the gradients on your own but it's best to give a little bit of leg room at the beginning and the end so that it can pick up the colors it should be getting.

If your gradient texture doesn't have transparency, then the alpha of the colors will always be 1, which will be rendered into the fog. If you don't want to render fog at certain parts by using transparency, make sure that your gradient texture has transparency or a transparent background.

You can create the gradients via Adobe photoshop, an art software, or scripts in Unity. Unity has a gradient library that you can use to bake a gradient texture. A recommended asset is LKHGames' Gradient Texture Generator (https://assetstore.unity.com/packages/tools/utilities/gradient-texture-generator-216180), it is free and open source. You can create your own scripts if you prefer that. Do note that the effect does take alphas into account so if you have anything that is transparent, it will not render. This is great if you don't want specific parts of the fog to be colored or rendered.

## Known Issue

To enhance immersion, our fog effect can blend with background objects, particularly those that are farther away. However, because the skybox is considered extremely distant due to camera depth, the fog can sometimes overpower it, causing the skybox's color to shift to match that of the fog. To mitigate this effect, the alpha value of the far fog color is set to 0 by default.

Users have the option to adjust the alpha value of the far fog color to a setting greater than 0 to reintroduce the far fog feature. Fine-tuning the fog parameters to make the fog less apparent can help avoid these issues. Additionally, the fog can interfere with certain effects, such as fake volumetric spotlights. However, as mentioned, users can minimize these conflicts by fine-tuning the fog settings.

## Update Log

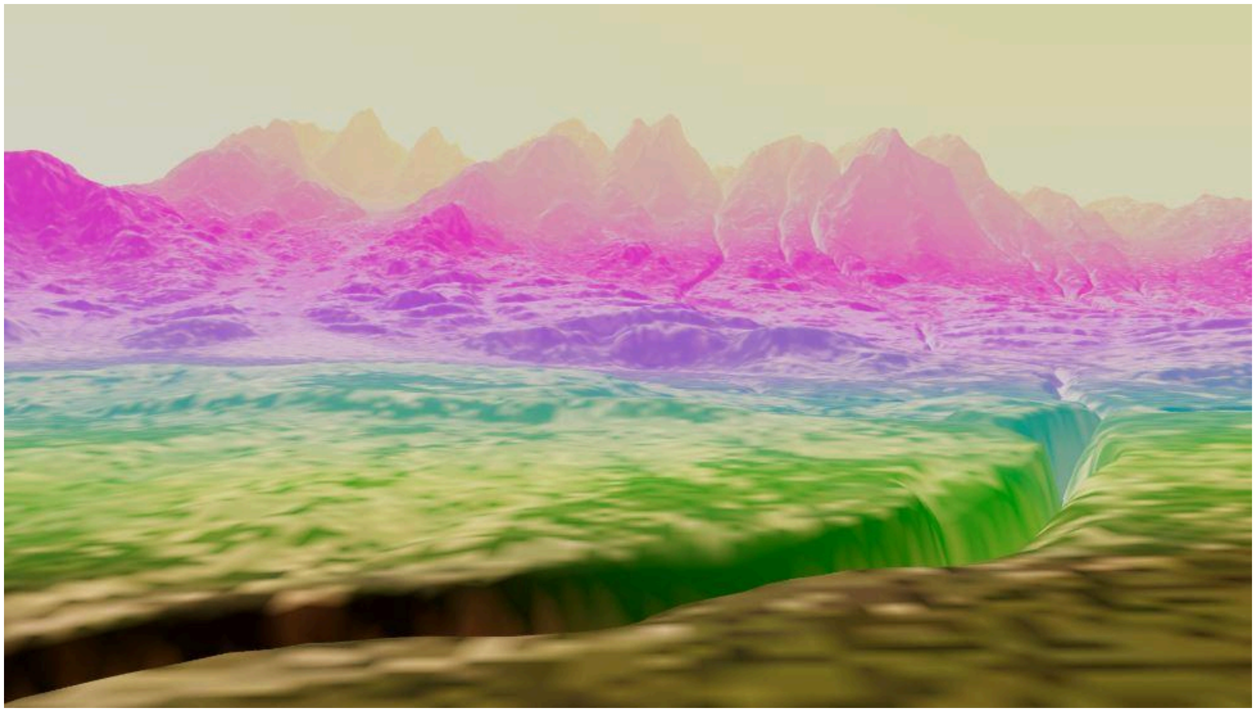
V0.9: The base implementation of the post processing.

## Credits

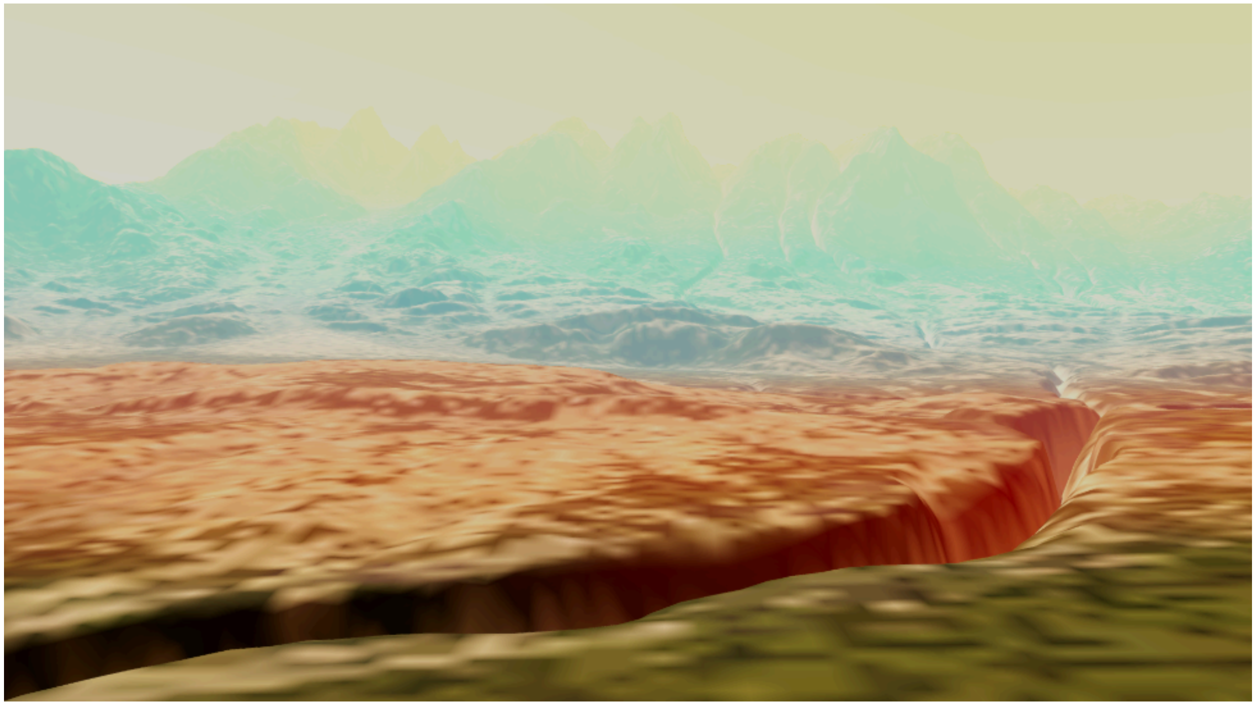
Some of the shadergraph code is based from Paulo Vilela's take on stylized multi-coloured fog in ShaderGraph Youtube Video (https://www.youtube.com/watch?v=YDT9s3nNVj0).

"Mossy/Grassy Landscape" (https://skfb.ly/6RYvL) by Šimon Ustal is licensed under Creative Commons Attribution (http://creativecommons.org/licenses/by/4.0/). No changes made. (The mountain landscape that you see in the screenshots for demonstration purposes)

### Example Screenshots



**What the fog looks like without using a gradient texture**



**What the fog looks like using the provided gradient texture**