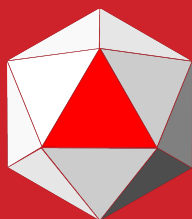


# Gradient Fog

Version 1.0  
User Manual  
April 10th, 2018

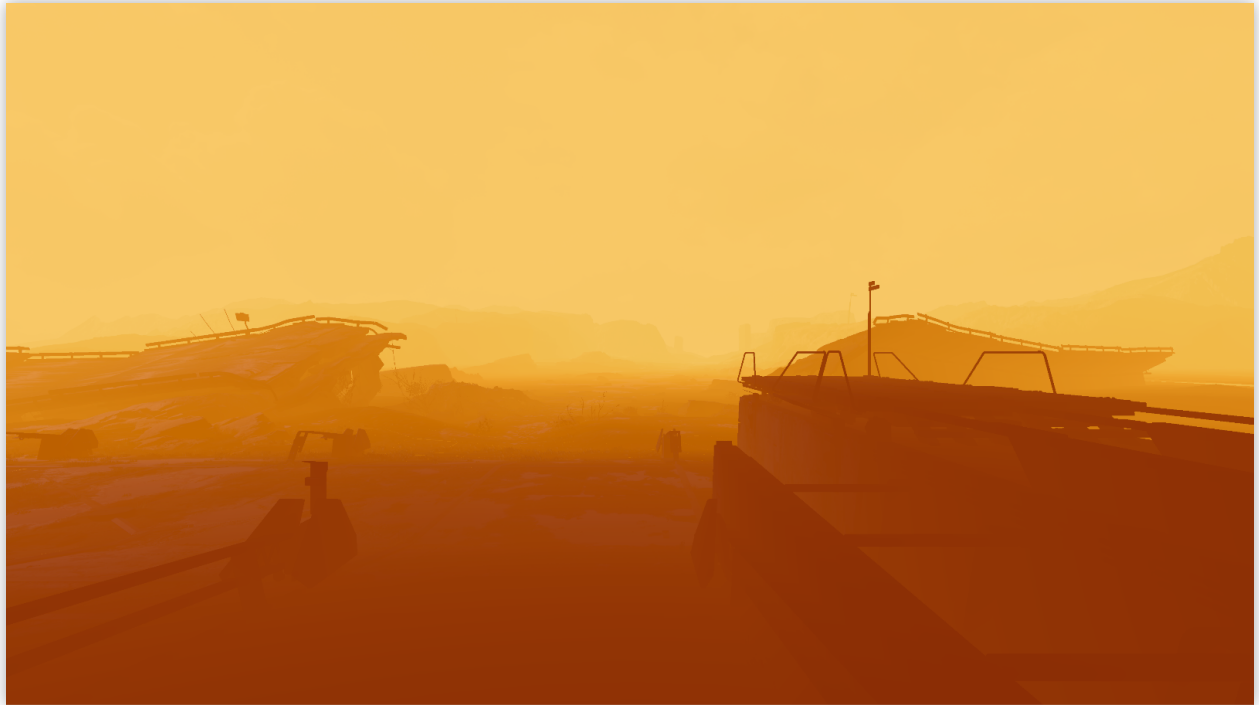


Ludogon Software

# Overview

---

**Gradient Fog** is a camera-based image effect that allows users to easily create colorful scene fog. With support of up to three different fog types, users are able to push visual boundaries by surpassing the limitations of Unity's built-in fog system.



Pictured above: an example of a scene using a brown-to-orange gradient.

## Features

---

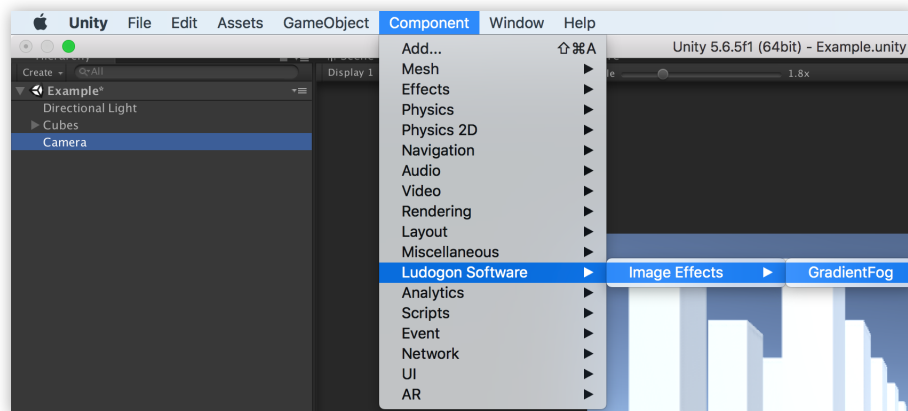
- Create gradients within the Unity Editor.
- VR ready. Single-pass and multi-pass stereo rendering support.
- Use up to three different fog types simultaneously: Distance Fog, Bottom Height Fog, and Top Height Fog.
- Works in both forward and deferred rendering.

# Getting Started

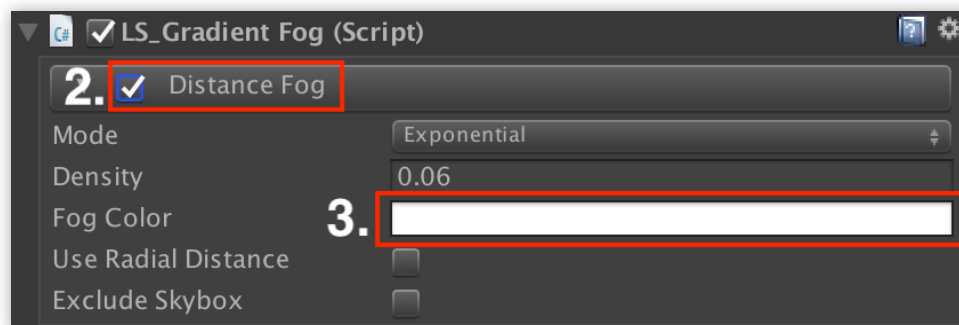
---

In order to properly see the effects of **Gradient Fog**, please select a scene with sufficient geometry. Apply **Gradient Fog** to a scene with the following steps:

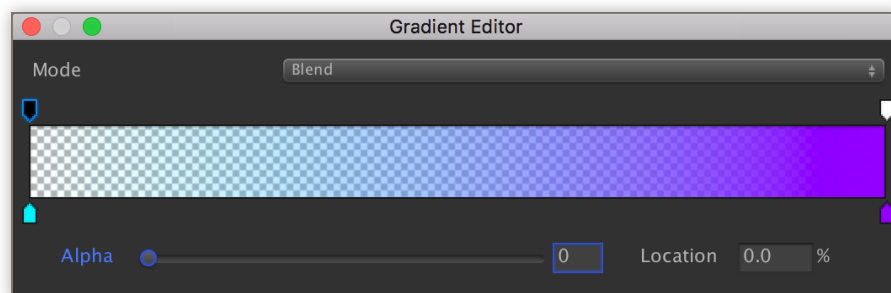
1. Select a camera within your scene. Add **Gradient Fog** to the camera by selecting from the Unity Editor menu: Component > Ludogon Software > Image Effects > Gradient Fog.



2. Enable Distance Fog from the **Gradient Fog** inspector.
3. Select Fog Color to open the [Gradient Editor](#). Fog Color is set to white by default.



4. Adjust Fog Color to your liking using the Gradient Editor. Adjusting the alpha values will affect the opacity of the fog.



5. Have fun exploring the other properties provided by the Gradient Fog inspector!

# Component Properties

## Distance Fog

▼

☒ Distance Fog

Mode

Linear

Start

0

End

20

Fog Color

Use Radial Distance

☐

Exclude Skybox

☐

Property	Function
Mode	Controls the mathematical function determining the way Fog Color is interpolated with distance.
Start	Push the initial Fog Color value away from the camera by this amount.
End	Controls the distance from the camera where the Fog Color will be completely interpolated.
Density	The degree to which the Fog Color is interpolated with distance.
Fog Color	The color of the fog drawn in the scene.
Use Radial Distance	Distance fog is based on radial distance from camera when checked.
Exclude Skybox	Exclude far plane pixels from distance-based fog? (Skybox or clear color).

## Bottom Height Fog

▼

☒ Bottom Height Fog

Mode

Linear

Start

0

End

1

Fog Color

Height

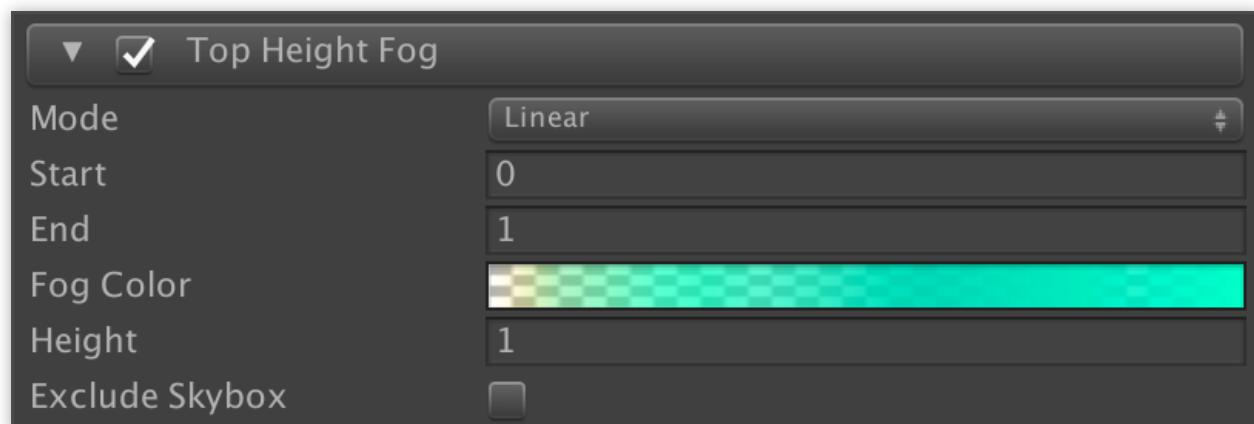
-1

Exclude Skybox

☐

Property	Function
<b>Mode</b>	Controls the mathematical function determining the way Fog Color is interpolated.
<b>Start</b>	Controls the Y position in which Fog Color interpolation begins relative to the Height parameter.
<b>End</b>	Controls the Y position in which Fog Color interpolation ends relative to the Height parameter.
<b>Density</b>	The degree to which the Fog Color is interpolated.
<b>Fog Color</b>	The color of the fog drawn in the scene.
<b>Height</b>	Fog bottom Y coordinate.
<b>Exclude Skybox</b>	Exclude far plane pixels from height-based fog? (Skybox or clear color).

## Top Height Fog



Property	Function
<b>Mode</b>	Controls the mathematical function determining the way Fog Color is interpolated.
<b>Start</b>	Controls the Y position in which Fog Color interpolation begins relative to the Height parameter.
<b>End</b>	Controls the Y position in which Fog Color interpolation ends relative to the Height parameter.
<b>Density</b>	The degree to which the Fog Color is interpolated.
<b>Fog Color</b>	The color of the fog drawn in the scene.
<b>Height</b>	Fog top Y coordinate.
<b>Exclude Skybox</b>	Exclude far plane pixels from height-based fog? (Skybox or clear color).

# Hardware Requirements

---

**Gradient Fog** requires a graphics card that supports depth textures.

## Limitations

---

### Transparent Objects

Since Unity's built-in transparent shaders do not write to the depth buffer, fog values can not be properly calculated for transparent objects. In order to somewhat alleviate this issue, **Gradient Fog** is rendered after transparent objects, effectively allowing transparent objects to "inherit" the depth values of opaque objects drawn before them. Unfortunately, this can still result in some unwanted behavior, such as transparent objects not receiving any fog whatsoever. For example, terrain objects drawn using Unity's Terrain Engine are rendered as transparent objects. Thus, terrain objects will not receive any fog in instances where there are no opaque objects behind them. In such instances, use Unity's built-in Fog settings in conjunction with **Gradient Fog**.



Left: Gradient Fog is not being applied to the trees in the background. Right: Unity's built-in Fog being used in conjunction with Gradient Fog.

### Anti-aliasing

Using **Gradient Fog** while MSAA is enabled will cause artifacts around edges. This is due to the depth buffer not being antialiased. In instances in which anti-aliasing is required, use an anti-aliasing image effect instead of MSAA. Don't forget to ensure your chosen anti-aliasing script executes after **Gradient Fog**.

### Unity 2018.1

**Gradient Fog** currently does not support the Lightweight Render Pipeline or High Definition Render Pipeline introduced in Unity 2018.1. Support for these render pipelines will be added in the near future.