



UI Highlight Shader

v1.0
Documentation

Contents

1	Introduction	2
2	Documentation	3
2.1	Universal Render Pipeline (URP)	3
2.1.1	Setup	3
2.1.2	Shader Options	4
2.1.3	API	5
2.1.4	Controller Script	5
3	Support	7

Section 1

Introduction

The UI Highlight shader allows you to highlight UI elements by gradually moving an emissive wave along their surface. The wave's emission, sharpness (strength), thickness, speed, direction, and angle are fully customizable.

Use this shader to highlight any elements in the Canvas, as well as to add emission to your UI elements.

WARNING

This shader supports only Universal Render Pipeline (URP) on Unity 6.0+. Built-in Render Pipeline (BIR) and High-Definition Render Pipeline (HDRP) are not supported.

A controller script is included for dynamically changing the shader variables, creating dynamic effects, such as a rotating wave or a highlight effect with a changing color.

You can stack these effects on top of each other by adding several or all effects to the same element, creating unique UI effects.

Feel free to check out the demo scenes included in the asset for inspiration.

Section 2

Documentation

2.1 Universal Render Pipeline (URP)

2.1.1 Setup

The following paragraph presents the steps necessary to make the UI highlight shader work in your project. Visit <https://docs.project-gamedev.com/ui-highlight-shader/setup.html> for image-aided instructions.

The UI highlight shader is, in principle, the same as the regular unlit shader you would use in your Unity project, only reconstructed and expanded. The setup is thus the same as with any other unlit shader:

- Create a new material for your UI object(s).
- Select the material; then, select **Project G.A.M.E.D.E.V./Essential 2D Shaders/UI Highlight Shader** (or **Text Highlight Shader** for Legacy Text objects, see below) as the shader source for this material.
- Assign the new material to your object(s).

That's it! The shader is now active. Feel free to play around with the material properties to find a highlight effect that you are happy with.

Keep in mind that certain post-processing settings (most notably **Bloom** and **Tonemapping**) will have a profound effect on the UI Highlight effect.

In case you wish to animate the highlight effect, attach `UIHighlightController.cs` to an object in your scene. Consult with 2.1.4 for more information on the controller script.

NOTE

For **Text (Legacy)**, use **Text Highlight Shader** instead of **UI Highlight Shader** and attach `TextHighlightSetup.cs` to each Text object you wish to highlight.

i NOTE

When using the **Text Highlight Shader** for **Text (Legacy)** objects, the highlight effect will only be visible in **Play Mode**.

⚠ WARNING

Not all UI elements are supported by this shader. Please consult with the asset's description on the Unity Asset Store (<https://assetstore.unity.com/packages/slug/337896>) for an up-to-date list of supported UI elements.

2.1.2 Shader Options

Albedo Color An additional color multiplied with the one already present in the albedo texture. Use this to further customize the look of your sprite.

Alpha Cutoff This is a threshold value – all parts of the texture whose alpha value is less than the alpha cutoff will not be rendered.

Emission Color The emission color as a regular RGB color.

Emission Strength The strength of the emission. This allows you to make your emission stronger or weaker, regardless of color. Keep in mind that **different post-processing effects (most notably bloom)** will have a **profound** effect on the look of your emission.

Wave Controller The controller value for the wave effect. This value can be controlled via scripts (an example is provided) to create a smooth progressive wave along the image by lerp'ing the controller value, or to keep the wave stationary at a specific position by setting the controller to a constant value. Recommended values are 0 for the beginning of the wave effect, and 1 for its end. These values may need to be tweaked manually on larger UI elements or thicker highlight waves.

Wave Thickness The thickness of the emissive highlight wave. Customize the value to your liking.

Wave Strength The overall feel of the wave. Higher values make the wave thicker, brighter, and with sharper edges, whereas lower values make the wave lighter, with weaker emission, and smoother along the edges.

Angle The angle of the highlight wave in degrees of rotation on the z-axis. Leave at 0 or set to 180 for a horizontal wave moving left-to-right or right-to-left. Set to 90 or 270 for a vertical wave moving bottom-to-top or top-to-bottom. Set to any other value for custom angles.

2.1.3 API

Visit <https://docs.project-gamedev.com/ui-highlight-shader/api.html> for the UI Highlight Shader API documentation. This API allows you to modify all shader values in-code, from your own scripts, if necessary.

2.1.4 Controller Script

A controller script – `UIHighlightController.cs` – is included for indefinitely lerp-ing the UI highlight effect on a UI element. If you wish to set up repeating effects, attach this script to any object in your scene, and set it up with the elements and effects you wish to enable.

NOTE

The controller script accepts any `GameObject` elements and searches them for a `SpriteRenderer` or a `Graphic` component. This allows the effect to work both for **UI elements** (images, buttons, etc.) and for **sprites** (imported Photoshop files or images).

The controller script is split into the following sections, visible in the Editor:

General The General section contains a single setting – **Individual UI Elements**. Enable this setting to have all effects applied to the same UI elements. If you disable this option, you will need to specify (different or the same) UI elements for each effect you enable!

Main Controller This is the main highlight effect. Enabling this section will let you customize the movement of the highlight wave.

Further Effects Each shader property can be lerped independently by enabling its section. For example, you can set up rotating waves by enabling the Angle Controller effect, or make the wave change its color by setting up the Color Controller effect. Feel free to experiment with different effects and values for each effect.

WARNING

You can have multiple instances of the `UIHighlightController.cs` script in your scene, and different effects can target the same UI elements. However, ensure **different instances of the script do not target the same UI elements!**

For example, it is fine to have `UIHighlightController 1` target elements A and `UIHighlightController 2` target element B, but do not have both `UIHighlightController 1` and `UIHighlightController 2` target element A!

i NOTE

When enabling several effects at once, it is fine to target the same UI elements **from within the same** `UIHighlightController`.

For example, it is fine to have the Main Controller effect target elements A and B, and the Angle Controller effect target element A as well. Just make sure both effects are **inside the same** `UIHighlightController` script!

Section 3

Support

In case you experience any problems with the shader, can't manage to set it up properly, or would simply like to propose an improvement, don't hesitate to contact us at contact@project-gamedev.com, or fill out the form at <https://project-gamedev.com/contact.html>.

If you are satisfied with this product, please consider leaving a review on the Unity Asset Store page (<https://assetstore.unity.com/packages/slug/337896>).

Best of luck with your game development journey!