

# UI Tutorial Fade

---

Tutorial Fade is an easy-to-implement solution for creating tutorials that allows you to highlight UI elements or 3D renderers, allowing the user to click only on the highlighted elements.

[Link](#) | [Documentation](#)

## Description:

---

**TutorialFadeImage** allows you to create full screen fade (similar with simple **Panel**), however custom shader will draw holes inside this fade.

Moreover, **TutorialFadeImage** will prevent clicking on any zone except these "holes", so you can force user to click entire button, for example.

## Getting Started:

---

1. Add **Tutorial Fade Image** game object to your canvas via GameObject->UI->Tutorial Fade Image.
2. Add to UI element or any Renderer on your canvas **Tutorial Highlight** component via AddComponent->UI->Tutorial Highlight
3. To enable/disable hole you can enable/disable either **Tutorial Highlight** component or game object this component attached to.
4. You're awesome! ♥

## Issues:

---

### Overlay canvases

While using **RendererTutorialHole** or just attaching **Tutorial Highlight** component to renderer (not UI element) it's recommended to not use Canvases with render type **Overlay**. Actually you can use **Overlay** canvas, but it's important to add camera to them. For that you can change canvas render mode to **Camera**, then select your **Main Camera** in appeared field and then change your canvas render mode back to **Overlay**.

### Max holes count

By default max holes size is set to 5. If you will try to create more holes, they won't appear and you will receive error message in console. However if you need, you can change this value. It is an advanced tip, because extending max holes count means changing source file, however if you lead instructions everything will be ok. First of all find file called **TutorialFadeImage.cs** and find there this line of code:

```
const int HolesSize = 5;
```

You can change this value to anything you want. Technically the value can be any (even 128 or higher for example). However it's not tested. Then you should find another file called **UITutorialFade.shader**, find there this line of code

```
float4 _Holes[5];
```

And then change it to the same value. All done!

## Properties:

---

### TutorialFadeImage

Property	Description
Source Image	This value could be used if you want to add texture to your fade image.
Color	Color of the fade. If Source Image property provided, the sprite will be multiplied to this color
Material	You can select your custom material here. Very advanced trick, it's not recommended to use this field.
Smoothness	Property that allows you to change smoothness of hole edges (i.e. hole size). Allowed values: from 0 to 1.
Raycast Target	This property allows you to pass click through <b>TutorialFadeImage</b> if it's unchecked. It can be used of you using <b>TutorialFadeImage</b> for visual effect, but not for click blocking.

## Tutorial Highlight

Property	Description
Tutorial Fade	<b>TutorialFadeImage</b> which will render hole for this object. The field is filled automatically, if there exists any <b>TutorialFadeImage</b> in the scene and you could not change this value to <b>None</b>

## UITutorialFade.shader

Property	Description
_MainTex	Base color texture.
_Aspect	Aspect ratio of the UITutorialFade's quad.
_Smoothness	Property that allows you to change smoothness of hole edges (i.e. hole size). Should be greater then 0.