

HIGHLIGHT PLUS 2D

STANDARD &
LWRP/HDRP Compatible

Kronnect

Contents

- Introduction3
- Quick Start and Demo Scene3
- How to use the asset in your project.....4
 - Option 1: Highlighting/customizing a SINGLE sprite4
 - Option 2: Highlighting/customizing ANY sprite automatically4
 - Option 3: Highlighting specific sprites using scripting.....4
 - Ignoring specific sprites from highlighting4
 - Highlighting single vs group of sprites4
 - Exclusive and grouped outline effects4
- Options and effects5
 - General options.....5
 - Highlight options5
 - Overlay6
 - Outline6
 - Glow6
 - Zoom Scale7
 - See-Through7
 - Shadow Options7
 - Hit FX.....8
- Advanced Topics and Notes.....9
 - Sorting layers.....9
 - Using scripting to add effects.....9
 - Events / reacting to selection.....9
 - 2D animation support9
 - License of the icons9

Introduction

Thank you for purchasing!

Highlight Plus 2D is a simple yet powerful package for adding outline, glow and other effects to your sprites.

We hope you find the asset easy and fun to use. Feel free to contact us for any enquiry.

Visit our Support Forum on <https://kronnect.com> for help and access to the latest beta releases.

Kronnect

Email: contact@kronnect.com

Support Forum: <https://www.kronnect.com/support>

Quick Start and Demo Scene

1. Import the asset into your project or create an empty project.
2. Go to Demo folder and run the demo scene to quickly test the asset effects.
3. Examine the code behind the script attached to the Demo game object.

The Demo scene contains two examples of how to render the effects (one or two ways can be used):

- *A HighlightPlus2DManager instance (the prefab) which detects when the pointer is over a sprite and triggers the effects. The manager is a single component that you can add from the top menu GameObject -> Effects -> Highlight Plus 2D -> Create Manager.*
- *A Potion sprite which has a Highlight Plus Effect and Highlight Plus Trigger component. The Highlight Plus Effect contains the effects settings while the Highlight Plus Trigger component is responsible for detecting when the pointer enters the sprite and enable those effects using the settings from the Highlight Plus Effect component.*

How to use the asset in your project

Option 1: Highlighting/customizing a SINGLE sprite

- Add HighlightEffect2D.cs script to any sprite. Customize the appearance options.
- Optionally add HighlightTrigger2D.cs script to the sprite GameObject. It will activate highlight on the sprite when mouse pass over it. Note that a collider must be present on the sprite. Note: adding a HighlightTrigger2D script to a sprite will automatically add a HighlightEffect2D component.

Option 2: Highlighting/customizing ANY sprite automatically

- Select top menu GameObject -> Effects -> Highlight Plus 2D -> Create Manager.
- Customize behaviour of Highlight Manager 2D. If a sprite already has a HighlightEffect component, it will use those settings instead.

Option 3: Highlighting specific sprites using scripting

After adding the Highlight Effect component to your sprite, use GetComponent<HighlightEffect2D> and set the "highlighted" property of the component to true.

Ignoring specific sprites from highlighting

Add a HighlightEffect2D component to the sprite and activate the "Ignore" checkbox under Highlight section.

Highlighting single vs group of sprites

Use the "Include" setting in the inspector to specify if you want to include the children sprites when this one is highlighted. By default, highlight plus will include all children but you can restrict this to only highlight this one.

Note that if any children has its own Highlight Plus 2D component, it won't be included with the parent.

Exclusive and grouped outline effects

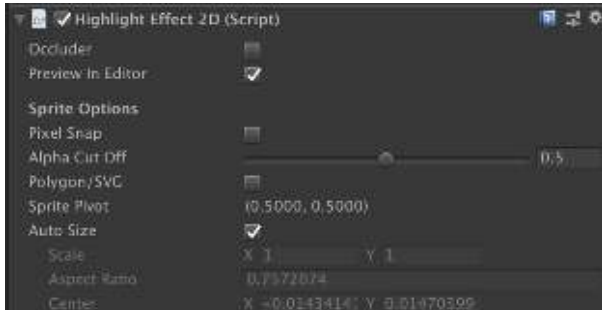
By default, Highlight Plus combines outline of overlapping sprites. If you want certain sprites to preserve their own outline you have two options:

- a) Enable the "Exclusive" toggle under the Outline section. That will make the outline to show independently for that sprite.
- b) Add a HighlightGroup2D component to the sprites in a group and assign a group number (ie. "1"). Group 0 does not do anything. The outlines of that group will not be combined with other sprites, only among sprites of the same group.

Options and effects

Tip: a tooltip with a short description will appear when you move the mouse over each option label in the inspector.

General options



- **Occluder:** enabling this option will trigger the see-through on other sprites with Highlight Plus 2D when passing behind this one.
- **Preview In Editor:** enabling this option will render the effects in Editor while not in play mode.
- **Pixel Snap:** render effects using pixel perfect adjustment.
- **Alpha Cut Off:** any pixel alpha below this value will be considered a transparent pixel.
- **Polygon/SVG:** enable this option if the sprite uses polygon packing or if it's a SVG sprite.
- **Auto Size:** enable if you want to override the size, aspect ratio or center for the effects.

Highlight options



- **Ignore:** enable to ignore any effect on this sprite. This option is useful if you have the Highlight Effect applied to a root object with the Include option set to Children. Using this “Ignore” option allows you to ignore this one but apply the effects to the others in the group.
- **Highlighted:** global setting for the highlight status.
- **Include:** determines which objects should be included when applying these effects. Currently the options are “Children”, “Only This Object” or “Root to Children”.
- **Exclusive:** by default, the outline or glow effect will be combined around the objects covered by a single Highlight Effect script. Enabling this option will force the outline and glow to render separately on this object.

Overlay

The overlay effect adds a transparent solid color layer on top of the sprite with different stylized options like custom blending and color animation.



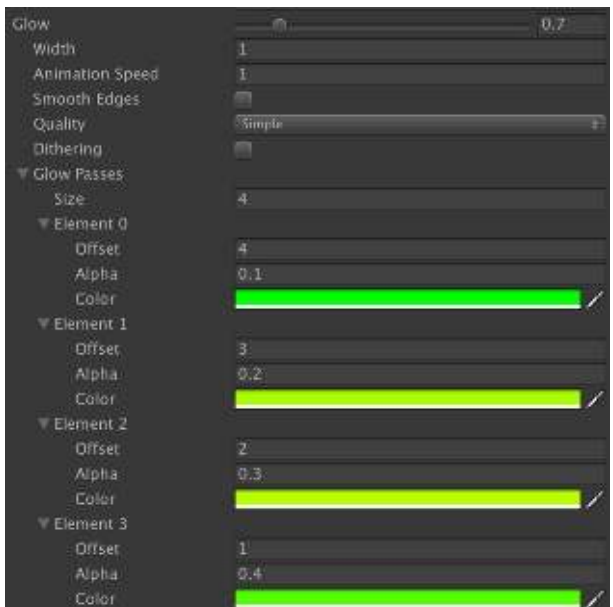
Outline

The outline effect adds a colored border to the sprite. Different quality levels are supported.



Glow

The glow effect adds a light “bloom” around the sprite. You can configure the number of glow passes (by default 4) and specify different appearance options for each pass to create infinite possibilities (ie. a rainbow or gradient-based bloom).



To change the glow color from script you can use the **SetGlowColor(color)** method of the HighlightPlus2D script. And that's all!

Or if you want to change the individual settings for each glow pass like offset, alpha and color as well, you can use this code:

```
HighlightEffect2D effect = GetComponent<HighlightEffect2D>();
GlowPassData[] glowPasses = effect.glowPasses;
glowPasses[0].color = Color.yellow;
glowPasses[0].alpha = 0.5f;
glowPasses[0].offset = 4;
glowPasses[1].color = Color.blue;
glowPasses[2].alpha = Color.blue;
...
effect.UpdateMaterialProperties();
```

By default the "effect.glowPasses" array length is 4, which you can modify in the inspector or by assigning a new array to the glowPasses property. Note that GlowPassData type of each entry in the array is a struct which contains the alpha, offset and color properties.

Zoom Scale

The zoom scale will modify the sprite scale when highlighted.



See-Through

The see-through or x-ray effect makes the sprites partially visible behind other solid objects.



Shadow Options

Sprites do not support shadow-casting by Highlight Plus 2D can add them! Use this section to add either 2D or 3D shadows to the sprite.



Hit FX

The Hit FX effect adds a quick flash to the sprite resembling a hit.

Call HitFX() method to execute a hit effect:

```
using HighlightPlus2D;  
...  
HighlightEffect2D effect = mySprite.GetComponent<HighlightEffect2D>();  
effect.HitFX(color, fadeOutDuration, initialIntensity);
```


Advanced Topics and Notes

Sorting layers

Highlight Plus 2D Effects are always rendered into Sorting Layer 0. If you have sprites that use a greater sorting layer, you may need to change them, so they use a lower sorting layer.

Using scripting to add effects

Use `GetComponent<HighlightEffect2D>()` to get a reference to the component of your sprite. Most properties shown in the inspector can be accessed through code, for example:

```
using HighlightPlus2D;
...

HighlightEffect2D effect = mySprite.GetComponent<HighlightEffect2D>();
effect.outline = true;
effect.outlineColor = Color.blue;
effect.Refresh();
```

Events / reacting to selection

When an object is highlighted the `HighlightStart` and `HighlightEnd` messages are invoked on all scripts attached to the sprite. You can also use the `OnObjectHighlightStart/OnObjectHighlightEvent` to react to highlight events and cancel the event depending on the situation.

Check `PotionHighlightEventSample.cs` script in the demo scene.

2D animation support

To enable support for Sprite Skin animations, please make sure that:

- 1) Polygon/SVG packing is enabled in the inspector.
- 2) Edit `HighlightEffect2D.cs` script and uncomment this line on top:

```
///#define USE_SPRITE_SKIN
```

License of the icons

All icons in the demo scene are Public Domain.