Table of Contents

Thank you for buying the **Easy Performant Outline**, we hope it will help you in your project.

If you liked our product, do not forget to rate it in an Asset store: http://u3d.as/1EXT

Your feedback will help me improve the project.

If you have any suggestions or questions, you can write to this email:

pirate.parrot.software@gmail.com

**New information is marked in blue*

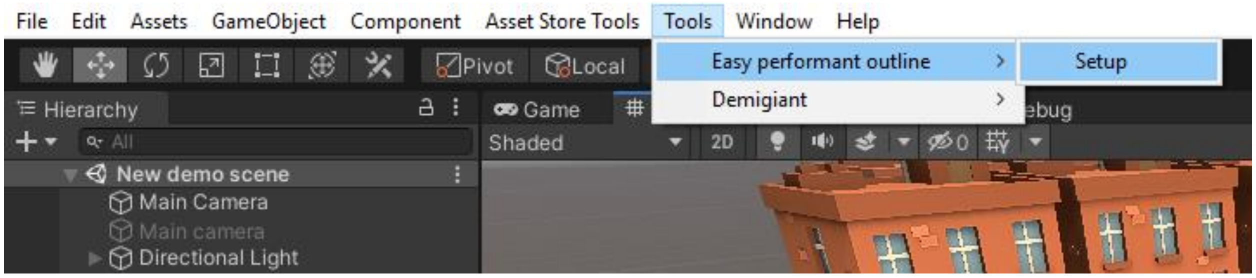
Promo Video

Tutorial Videos

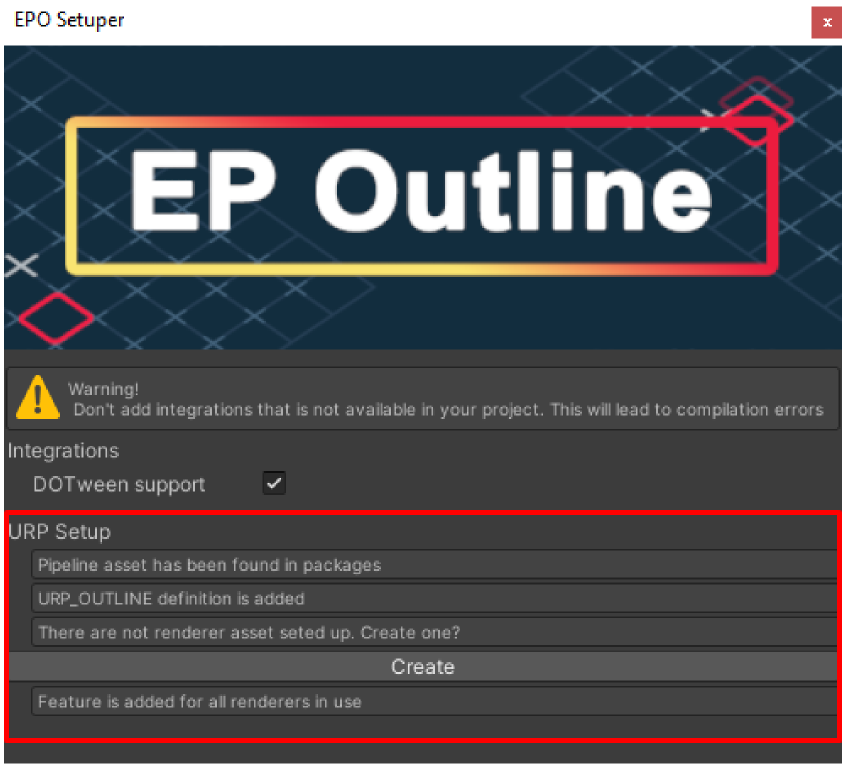
Easy Performant Outline 2.0 Setup

URP Setup

In order to set up the urp support, please click **Tools** - **Easy performant outline** - **Setup** menu item:



After follow the steps proposed by the wizard:



Warning!
Don't add integrations that is not available in your project. This will lead to compilation errors

Integrations

DOTween support ✓

URP Setup

Pipeline asset has been found in packages

URP_OUTLINE definition is added

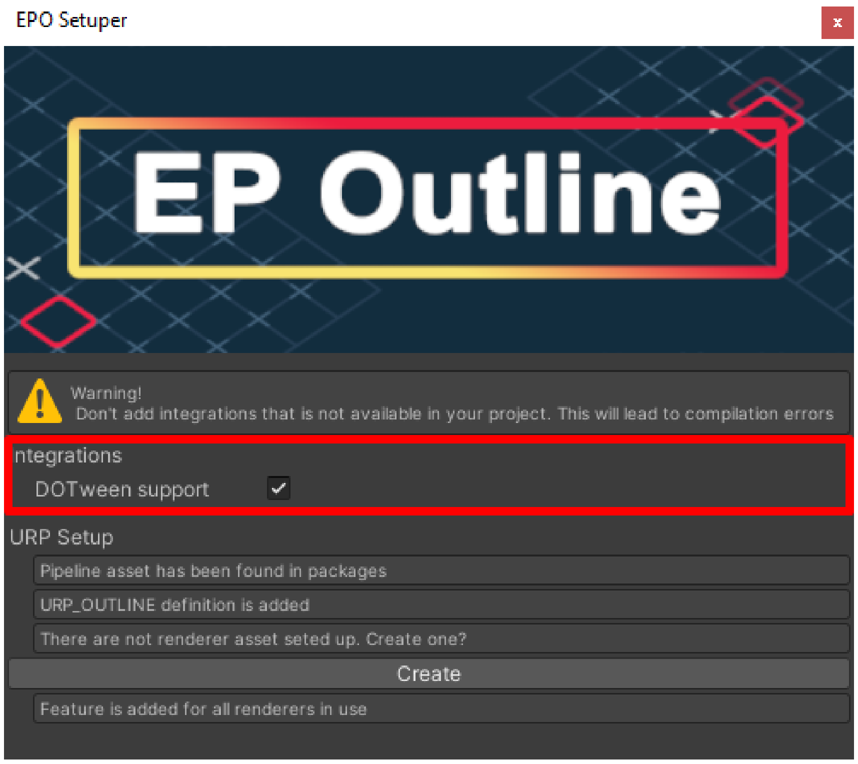
There are not renderer asset seted up. Create one?

Create

Feature is added for all renderers in use

DOTween Setup

You can enable **DOTween support** in the same menu under **Integrations** (*please enable it only if you have DOTween added to your project*):



EPO Setuper

Warning!
Don't add integrations that is not available in your project. This will lead to compilation errors

ntegrations
DOTween support ✓

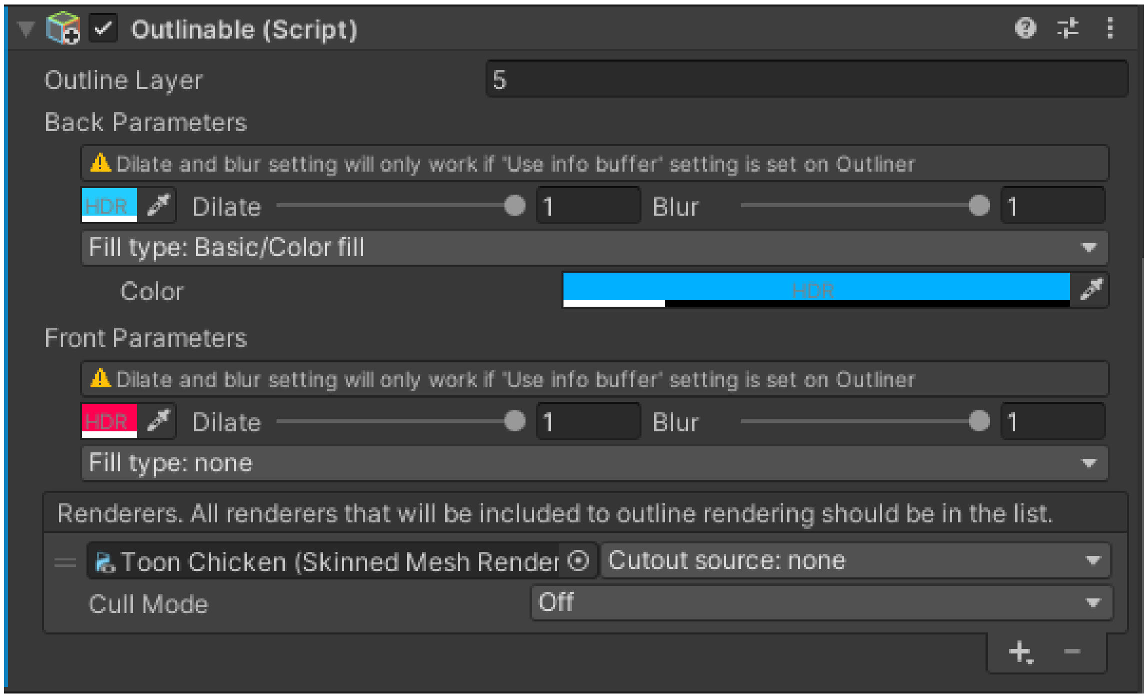
URP Setup
Pipeline asset has been found in packages
URP_OUTLINE definition is added
There are not renderer asset seted up. Create one?
Create
Feature is added for all renderers in use

Easy Performant Outline 2.0

Easy performant outline – is asset that provides high quality outlining for Unity. It supports various outline modes and features.

Components

1) The *Outlinable* component is the component which indicates the objects which should be outlined by the outline system.



Outlinable (Script)

Outline Layer: 5

Back Parameters

⚠ Dilate and blur setting will only work if 'Use info buffer' setting is set on Outliner

Dilate 1 Blur 1

Fill type: Basic/Color fill

Color

Front Parameters

⚠ Dilate and blur setting will only work if 'Use info buffer' setting is set on Outliner

Dilate 1 Blur 1

Fill type: none

Renderers. All renderers that will be included to outline rendering should be in the list.

Toon Chicken (Skinned Mesh Render | Cutout source: none

Cull Mode: Off

Parameters of Outlinable

- **Outline layer**

The layer of the outline. In case if you should have a separate outline which can overlap you can select different layer for the outlinables and add several Outliners (More on that later). In this case the outlines will not interfear with each other.

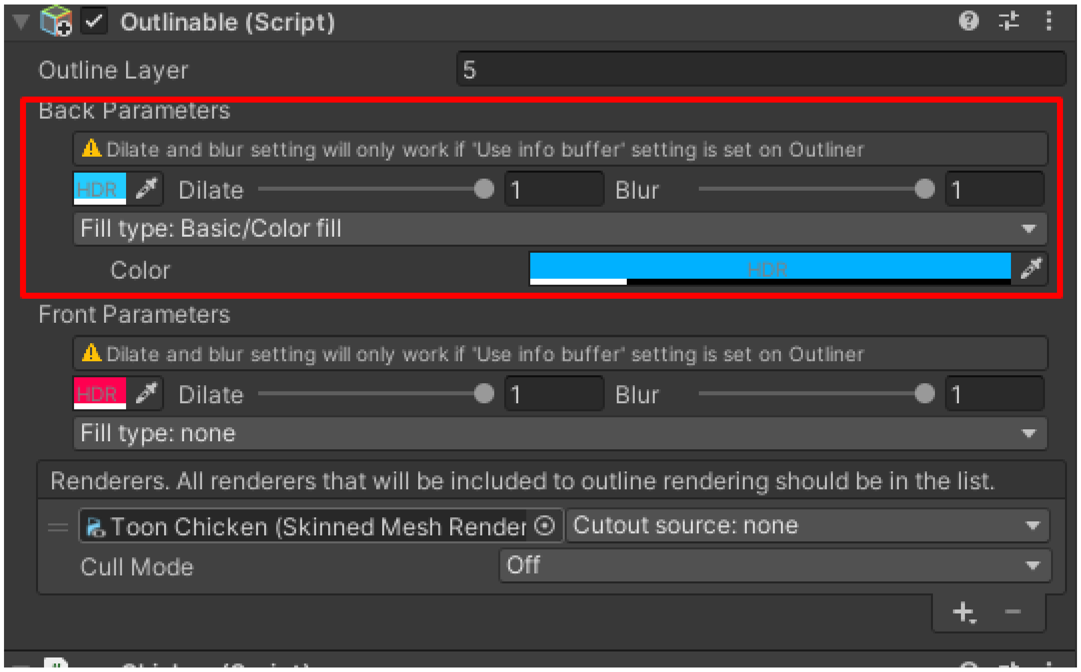
- **Back parameters**

The parameters which will apply to the outline when it is obstructed by somethins (Let's say behind the wall).

- **Front parameters**

The parameters which will apply to the outline which is directly visible (Not obstructed by anything).

Parameters details (both front and back):



Outlinable (Script)

Outline Layer 5

Back Parameters

⚠ Dilate and blur setting will only work if 'Use info buffer' setting is set on Outliner

HDR Dilate 1 Blur 1

Fill type: Basic/Color fill

Color HDR

Front Parameters

⚠ Dilate and blur setting will only work if 'Use info buffer' setting is set on Outliner

HDR Dilate 1 Blur 1

Fill type: none

Renderers. All renderers that will be included to outline rendering should be in the list.

Toon Chicken (Skinned Mesh Render | Cutout source: none

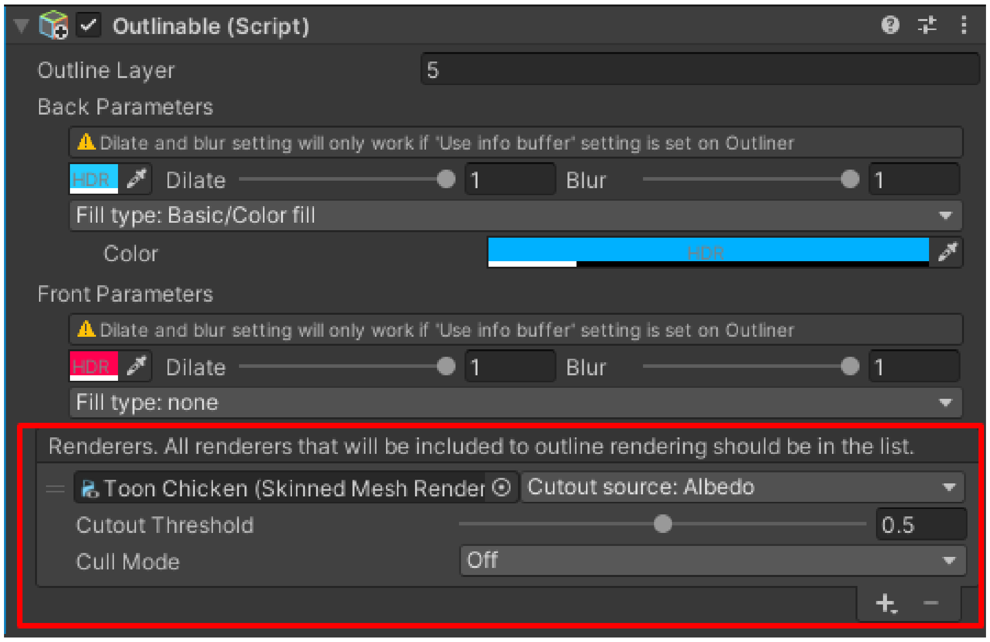
Cull Mode Off

Lets have a look at back parameters (The front works the same).

First is the color of the outline. It will change the outline color. On the right hand side you'll see dilate slider. If your outliner using info buffer (More on that later) you'll be able to change per object dilate shift. To the right of that you'll find a blur slider which is in case of using info buffer (More on that later) will change the amount of blur applied to each object.

Lower you'll find Fill type selector. It's a selector of the fill types available at the moment. Each of them might have different properties and describes how the fill should be rendered for each outlinable. If none is selected no fill will be applied.

- **Renderers**

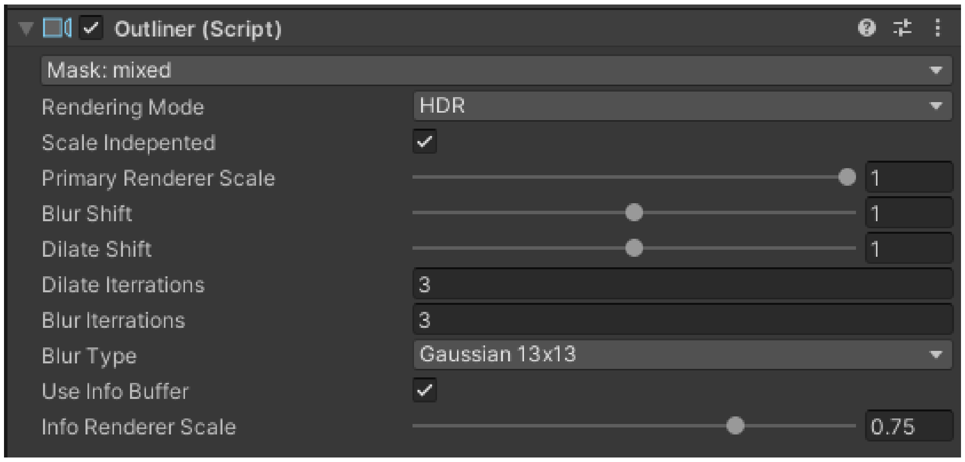

Outlinable (Script)
Outline Layer 5
Back Parameters
⚠ Dilate and blur setting will only work if 'Use info buffer' setting is set on Outliner
HDR Dilate 1 Blur 1
Fill type: Basic/Color fill
Color HDR
Front Parameters
⚠ Dilate and blur setting will only work if 'Use info buffer' setting is set on Outliner
HDR Dilate 1 Blur 1
Fill type: none
Renderers. All renderers that will be included to outline rendering should be in the list.
Toon Chicken (Skinned Mesh Render | Cutout source: Albedo
Cutout Threshold 0.5
Cull Mode Off

The list which describes which targets should be rendered to produce outline.

First parameter is the renderer which will be rendered in order to produce outline. Most of the time it should be all of the renderers included in your model/object. The disabled ones will not be rendered.

Cutout source is the paramter describing which of the texture the system should set as a cutout alpha source. In case if something is selecter there also will be Cutout threshould slider which is describing of how much alpha should the texture have in order to not be discarded.

Cull mode is the settings which is describing which cull mode should be used to render the outlinable target. Most of the time Back is the proper choise. In case of some geometry visible from both sides you should select Off. The Front setting should be used for extreme situations and is not commonly used but still exist.


Outliner (Script)
Mask: mixed
Rendering Mode HDR
Scale Independented ✓
Primary Renderer Scale 1
Blur Shift 1
Dilate Shift 1
Dilate Iterrations 3
Blur Iterrations 3
Blur Type Gaussian 13x13
Use Info Buffer ✓
Info Renderer Scale 0.75

2) The ***Outliner*** component

- **Mask** – which layers should the outline render. If you selected some layers on the outlinables you can discard it from rendering by this exact outlier (Althrough you can add several of them to the same camera with a different layers).

- **Rendering mode** – Should outliner use HDR or LDR for the outline (In case of postprocessing like bloom it's important to use HDR in other casese it's a waiste of resources). Please note that HDR will not be used if the camera itself is not allowed to use HDR or the system is not supporting HDR.

- **Scale independent** – if set to true will try keep the same size inf you change Pimary renderer scale property. If set to false will increase the Blur/Dilate size if making primary renderer scale smaller (Might be intendent behaviour).

- **Primary renderer scale** – if the screen resolution is lets say 800:600 than in case of primary renderer scale is set to 0.5 will render outline in 400:300 resolution. It's important to set the setting as low as possible while keeping intendent look of the outline.
The lower the value the better performance you gonna get.

- **Blur shift** – How much blur should affect the final image. In case of values larger than 1 might produce artifacts.

- **Dilate sift** – How much dilate should affect the final image. In case of values larger than 1 might produce artifacts.

- **Dilate iterations** – how many dilate effect iterations should be applied. The smaller the value the better performance you'll get. The larger the value the stronger dilate effect you'll get.

- **Blur iterations** – how many blur effect iterations should be applied. The smaller the value the better performance you'll get. The larger the value the stronger blur effect you'll get.

- **Blur type** – determines which of the blur types to apply. The most performant is BoxBlur, the most beautiful is Gaussian13x13.

- **Info buffer scale** – how should info buffer will be scaled down. 1.0 – the same side as the primary buffer. 0.1 – 1/10th of the primary buffer size. The smaller the value the better performance you'll get.

Coding API

In order to access the script in the script assigned to the same object as Outlinable:

```
using EPOOutline;
using UnityEngine;

public class MyScript : MonoBehaviour
{
    private void Start()
    {
        var outlinableToUse = GetComponent<Outlinable>();
    }
}
```

In order to add the outlinable in runtime you should use:

```
using EPOOutline;
using UnityEngine;

public class MyScript : MonoBehaviour
{
    private void Start()
    {
        var outlinableToUse = gameObject.AddComponent<Outlinable>();
    }
}
```

The Outlinable has multiple properties that can be changed throught scripting:

```
outlinable.RenderStyle = RenderStyle.FrontBack; // Now Outlinable will have a settings for both behind object rendering and in front
outlinable.RenderStyle = RenderStyle.Single; // Now Outlinable will render with the same options no matter obscured it or not

outlinable.DrawingMode = OutlinableDrawingMode.Normal; // The Outlinable will be rendered as usual
outlinable.DrawingMode = OutlinableDrawingMode.ZOnly; // The Outlinable will not be rendered at all but will render into ZBuffer which allow you to use obscured rendering for objects which is usually doesn't write depth. (Transparents, Sprites etc)
outlinable.DrawingMode = OutlinableDrawingMode.Normal | OutlinableDrawingMode.ZOnly; // The outlinable will draw both in ZBuffer and the outline itself

outlinable.OutlineLayer = 3; // Only Outliners with layer 3 enabled will render this outlinable;

outlinable.OutlineTargets.Add(new OutlineTarget(GetComponent<Renderer>())); // Now the Outlinable will render the Renderer we have on the gameObject we adding the Outlinable.
outlinable.OutlineTargets.Add(new OutlineTarget(GetComponent<Renderer>(), 1)); // Now the Outlinable will render the Renderer we have on the gameObject we adding the Outlinable. The submesh 1 will be added.


outlinable.OutlineParameters.Color = Color.green; // Setting green color for outline with outlinable.RenderStyle == RenderStyle.Single


outlinable.OutlineParameters.BlurShift = 0.5f; // Blur shift for outlinable.RenderStyle == RenderStyle.Single
outlinable.OutlineParameters.DilateShift = 0.5f; // Dilate shift for outlinable.RenderStyle == RenderStyle.Single
outlinable.OutlineParameters.Enabled = false; // Don't render outline at all for outlinable.RenderStyle == RenderStyle.Single
outlinable.OutlineParameters.FillPass.Shader = Resources.Load<Shader>("Easy performant outline/Shaders/Fills/ColorFill"); // Setting fill mode to Color fill for outlinable.RenderStyle == RenderStyle.Single
outlinable.OutlineParameters.FillPass.SetColor("_PublicColor", Color.yellow); // Setting yellow color for fill pass for outlinable.RenderStyle == RenderStyle.Single. Also has SetColor, SetVector, SetTexture for other types.

outlinable.FrontParameters.Color = Color.green; // Setting green color for front outline with outlinable.RenderStyle == RenderStyle.FrontBack

outlinable.FrontParameters.FillPass.Shader = Resources.Load<Shader>("Easy performant outline/Shaders/Fills/ColorFill"); // Setting fill mode to Color fill for front with outlinable.RenderStyle == RenderStyle.FrontBack
outlinable.FrontParameters.BlurShift = 0.5f; // Blur shift for front outline with outlinable.RenderStyle == RenderStyle.FrontBack
outlinable.FrontParameters.DilateShift = 0.5f; // Dilate shift for front outline with outlinable.RenderStyle == RenderStyle.FrontBack
outlinable.FrontParameters.Enabled = false; // Don't render front of the outline at all for outlinable.RenderStyle == RenderStyle.FrontBack
outlinable.FrontParameters.FillPass.Shader = Resources.Load<Shader>("Easy performant outline/Shaders/Fills/ColorFill"); // Setting fill mode for front outline to Color fill with outlinable.RenderStyle == RenderStyle.FrontBack
outlinable.FrontParameters.FillPass.SetColor("_PublicColor", Color.yellow); // Setting yellow color for front fill pass with outlinable.FrontBack == RenderStyle.Single. Also has SetColor, SetVector, SetTexture for other types.

outlinable.BackParameters.Color = Color.green; // Setting green color for back outline with outlinable.RenderStyle == RenderStyle.FrontBack

outlinable.BackParameters.FillPass.Shader = Resources.Load<Shader>("Easy performant outline/Shaders/Fills/ColorFill"); // Setting fill mode to Color fill for back with outlinable.RenderStyle == RenderStyle.FrontBack
outlinable.BackParameters.BlurShift = 0.5f; // Blur shift for back outline with outlinable.RenderStyle == RenderStyle.FrontBack
outlinable.BackParameters.DilateShift = 0.5f; // Dilate shift for back outline with outlinable.RenderStyle == RenderStyle.FrontBack
outlinable.BackParameters.Enabled = false; // Don't render back of the outline at all for outlinable.RenderStyle == RenderStyle.FrontBack
outlinable.BackParameters.FillPass.Shader = Resources.Load<Shader>("Easy performant outline/Shaders/Fills/ColorFill"); // Setting fill mode for back outline to Color fill with outlinable.RenderStyle == RenderStyle.FrontBack
outlinable.BackParameters.FillPass.SetColor("_PublicColor", Color.blue); // Setting blue color for back fill pass with outlinable.FrontBack == RenderStyle.Single. Also has SetColor, SetVector, SetTexture for other types.
```

Thank you for buying the **Easy Performant Outline**, we hope it will help you in your project.
If you liked our product, do not forget to rate it in an Asset store: http://u3d.as/1EXT

Your feedback will help me improve the project.

If you have any suggestions or questions, you can write to this email:

pirate.parrot.software@gmail.com