Table of Contents

Getting Started

1. Add **Translucent Image Source** to your main camera.

   

2. Create a ***Blur Config*** asset (or assign an existing one).

   

3. Create a ***UI > Translucent Image***, as you would with normal Image.

   

4. That's it!

   

Warning

By default, Translucent Image will use a default Material. To make sure your Translucent Image are not affected by asset update, create your own Material. See Customize section for more info.

Note

Sometime the effect does not shown up immediately. If that happen, just switch to play mode. The effect will continue to show even when exit play mode.

Customize

Translucent Image requires 2 components in a scene: 1 or more Source that control the blur amount, and multiple Translucent Image that replace the builtin Image component.

Translucent Image Source

Translucent Image Source generate the blurred background. Its let you controls how much blur is present, the quality of blur and thus the performance trade-off.

Blur Amount

There are too two methods of controlling the amount of blur, *Simple* and *Advanced*:

Simple:

- **Strength**. Using this single property, you can (kinda) smoothly change the blur amount at runtime.

Advanced:

- **Size**: Increase blur amount without affecting performance, but will look bad if set too high or too low. Reduce flickering when increased. Allow smoothly interpolating between blur amount, but can't go down to 0.
- **Iteration**: Increase blur quality and blurriness. Affect performance slightly. Animating this will create noncontinuous change in blur amount.

Performance:

These allow you to fine-tune the performance-quality trade-off. This is more useful for less capable mobile devices or high resolution tablet.

- **Max Depth**: Increasing this property will:
    - Improves performance
    - Increases blur amount if Iteration is high.
    - Cause flickering when the background moving.
- **Downsample**: Decrease the resolution before processing to increase performance. Side effects includes increased blurriness and flickering.
- **Blur Region**: Limit the blur effect to a region of the screen. If your UI does not span the entire screen, it is a good idea to use this to increase performance and reduce power usage.

    ##### Tip

    You can visualize and visually edit this by turning on preview. The number here work the same as the Camera component Viewport. It easier to weld if you change `x` and `y` before `w` and `h`.

- **Max Update Rate**: How many times the screen is blurred per second. Use this to improve performance and decrease power usage.

    ##### Tip

    Setting this to 0 will pause the effect completely. This can reduce power usage/ prevent overheat when you don't need a dynamically updating background, for example, in a pause menu.

- **Source Buffer**: Change if the effect does not work in URP 12+. URP 12+ use a double buffering approach, the content of the screen can be stored in either buffer at a given moment depend on rendering settings. There is no public API to retrieve the correct buffer, so this field is used as a workaround.
- **Preview**: Show the effect in full-screen without creating a Translucent Image. It also show the Blur Region as a resizable white rectangle.

Translucent Image

- **Source Image, Color, Raycast Target, Image Type**: same as built-in Image.
- **Material**: Multiple Translucent Images using the same material share some settings. They will batch dynamically into a single draw call.

    ##### Warning

    - Material used here must use the shader UI/TranslucentImage.

    - You should create your own Material instead of the default to avoid changes to look after asset updates.

- **Source**: a Translucent Image Source component. Will be automatically set to the first one found, so you should make sure there one in your scene before creating any Translucent Image. You can change this to select which camera will provide the background.
- **Sprite Blending**: Mix between Source Image property and the blurred background. This should be where you would use the alpha channel of the Color property in a normal Image component.

Shared settings

The following settings are shared across Translucent Images using the same material:

- **Vibrancy**: Colorful-ness of the background. 0 mean monochrome, and a negative value will invert the color.
- **Brightness**: Brighten or darken the background.
- **Flatten**: Reduce the background contrast. Useful when you can't predict the color of the background, but want to keep the content on top of it legible.

Controls Translucent Image from script

You can control all of the settings available in the inspector in C# through the exposed properties. See: TranslucentImage and TranslucentImageSource

The blur settings can't be accessed directly from the TranslucentImageSource class. They're are stored in a ScriptableObject. An easy way to access these settings is to create a public/serialized field. You can then assign the setting asset to it in the inspector:

```
public ScalableBlurConfig blurSettings;
```

Alternatively, if you have a reference to a TranslucentImageSource component, you can access the blur settings by casting it to a ScalableBlurConfig

```
void Start(){
   var theSource = FindObjectOfType<TranslucentImageSource>();
   var blurSettings = (ScalableBlurConfig) theSource.BlurConfig; // No other blur algorithms are available at the moment, so the cast will always success

   blurSettings.Strength = 42;
}
```

Universal Render Pipeline

Requirements

Only non-preview, non-beta version of URP and Unity will be supported.

The files required to support URP can be imported from the unitypackage at: TranslucentImage/UniversalRP support. They're not included by default, as that would produce errors in projects without URP.

Some demo scene does not work in URP. You can find demo scenes dedicated for URP under TranslucentImage/Demo/UniversalRP, after you import the support package.

Setup

1. Import the package at TranslucentImage/UniversalRP support.
2. Find your **Forward Renderer Assets**, and add **TranslucentImageBlurSource** to its list of **Renderer Features**:

Note

You may have multiple **Forward Renderer Assets**. In which case you have to add the **Renderer Feature** to all of them.

Finding the Forward Renderer Assets

1. You can find the **Forward Renderer asset(s)** you're using by finding the **Render Pipeline Settings asset** in **Graphic Settings**:

2. You may also have more **Quality Setting**. Be sure to check all Quality Levels that you use:

3. Double click the field under the cursor in the above images will take you to the **Render Pipeline Settings asset**, where you can find your **Forward Renderer asset(s)** in the list of **Renderer**:

Note for URP 12+

For URP 12+, the default settings may not work out of the box. Please toggle the Source Buffer option on Translucent Image Source to either A or B if there are weird artifact like flickering or blank background. Which option to select depends on the exact combination of rendering features used and is most easily determined through trial and error.

Blurring other UI elements

Sometimes, you not only want to blur the scene, but also other UI elements. A common use case for this is for popup. Translucent Image does not support blurring UI elements in arbitrary order for performance reason. However, by using 2 different Cameras and Canvases, Translucent Images from the upper Canvas can blur UIs in the lower Canvas.

TLDR

If you don't like reading, check out the included demo scene at:

- For Builtin Render Pipeline: Demo/_Scenes/Demo.unity
- For Universal Render Pipeline: Demo/UniversalRP/UniversalRP Demo UI Blur.unity

Desired outcome

To understand the required setup, it is useful to understand how Translucent Image work.

For performance reasons, Translucent Images does not perform any blurring. The Translucent Image Source component blurs the entire "screen" at once, and shares the result with multiple Translucent Images.

The Camera with the Source control what exactly is in the blurred background: what it renders is what is blurred. If the Camera "see" the Translucent Images, they will be included in their own background, causing them to quickly become opaque.

So, to blur UIs, what you want to do is to have 2 Canvases: one contains the UIs to be blurred, the other contains the Translucent Images. Then you make it so that the Camera with Translucent Image Source only "sees" the Canvas to be blurred.

The next section describes this setup in detail. You can also look at the demo scene mentioned above to quickly see how it work.

The setup

Here is a typical setup: *Example setup*

Here's what the hierarchy can look like:

```
Hierarchy
├── Popup Canvas
│   ├── Layer: UI
│   └── Render Mode: Screen Space - Overlay
│
├── Main Canvas
│   ├── Layer: UI
│   ├── Render Mode: Screen Space - Camera
│   └── Render Camera: UI Camera
│
├── UI Camera
│   ├── Clear Flags: Depth only
│   ├── Culling Mask: UI
│   └── Depth: 2
│
└── Main Camera
    ├── Clear Flags: any
    ├── Culling Mask: Everything but UI
    └── Depth: 1
```

Performance implication

You can stacks as many cameras and canvases as you like. However, with each extra Translucent Image Source you use, the GPU will have to do more work. A workaround is to disable the Source that is not the top-most. In fact, both Windows 10 and macOS do this: *Windows 10 only use blur on the top-most UI*

World Space UI

World Space UI face the same problem as blurring other UIs - if we want to batch blurring operations to achieve high performance, we cannot have Translucent Images interleaved between what they want to blur. If you simply put Translucent Images in world space, they will continuously blur themselves, causing an "overexposed" effect.

To work around this, use a separated Camera for Translucent Images, an example of this setup available at: Le Tai Asset/TranlucentImage/Demo/World Space UI. Particularly, the World UI Camera should:

- Have higher Depth than your Main Camera.
- Have Culling Mask set to UI layer only.
- Have Depth only clear Flags.
- Other properties should match your Main Camera setting.
- Be in the same position as your Main Camera - setting it as children with position and rotation of (0,0,0) is the simplest way.

Also, your Main Camera should have Culling Mask set to *exclude* UI layer.

Now, your Translucent Images should appear on top of scene geometry all the time, even if they are further away. While this is not ideal, it satisfies many use case, for example, world-space HUD, and allow for far better performance.

Frequently Asked Questions

Will this asset works well on my device?

The asset should run on any device. Performance-wise, it depends on your project's existing GPU consumption, but here some general rule of thumb:

- PC/Mac/Console: Should run well on almost everything except very old integrated GPU.
- Android: There're too many of them with too much difference in capability. The only way to know for sure is to test the demo on your target devices. On a Samsung Galaxy S7 Edge, the demo run at 60FPS with any setting.
- IOS:
    - Iphone: Apple A8 and later should hit 60FPS. A7 can hit 30FPS.
    - Ipad: Because of the higher pixel count, you'll need to use the resolution scale features to hit 60fps on A9 and below.

The blur does not work. UI in the demo scene just all white.

This usually due to one of the following:

- Sometime the effect only so up after switching to Play mode.
- If you're using URP, you have to do some setup, as detailed in the Universal Render Pipeline section first.
- Sometime Unity's import process break some random things. Try delete the whole folder and re-import the asset.

Can I smoothly animate the blur level?

The way the blur algorithm work make it difficult to smoothly animate the blur amount. The Strength property allows for mostly smooth change of blurriness, but there will still be some abrupt jump that is noticeable when interpolating slowly.

If you just need to fade in and out, you can use the alpha component of the Color property. You can also use Canvas Group as with normal Images.

Have another question?

Contact me

Support

If you need assistance regarding the asset or have a feature request, feel free to contact me by the form below or through my support email.

Support request Search Articles

Namespace LeTai.Asset.TranslucentImage

Classes

BlurConfig

Extensions

ScalableBlur

ScalableBlurConfig

ShaderId

TranslucentImage

Dynamic blur-behind UI element

TranslucentImageSource

Common source of blur for Translucent Images.

Interfaces

IBlurAlgorithm

Enums

BlurAlgorithmType

Enum BlurAlgorithmType

Namespace: LeTai.Asset.TranslucentImage

Assembly: LeTai.TranslucentImage.dll

Syntax

public enum BlurAlgorithmType

Fields

Name	Description
ScalableBlur	

Class BlurConfig

Inheritance

System.Object
UnityEngine.Object
UnityEngine.ScriptableObject
BlurConfig
ScalableBlurConfig

Inherited Members

UnityEngine.ScriptableObject.SetDirty()
UnityEngine.ScriptableObject.CreateInstance(System.String)
UnityEngine.ScriptableObject.CreateInstance(System.Type)
UnityEngine.ScriptableObject.CreateInstance<T>()
UnityEngine.Object.GetInstanceID()
UnityEngine.Object.GetHashCode()
UnityEngine.Object.Equals(System.Object)
UnityEngine.Object.Instantiate(UnityEngine.Object, UnityEngine.Vector3, UnityEngine.Quaternion)
UnityEngine.Object.Instantiate(UnityEngine.Object, UnityEngine.Vector3, UnityEngine.Quaternion, UnityEngine.Transform)
UnityEngine.Object.Instantiate(UnityEngine.Object)
UnityEngine.Object.Instantiate(UnityEngine.Object, UnityEngine.Transform)
UnityEngine.Object.Instantiate(UnityEngine.Object, UnityEngine.Transform, System.Boolean)
UnityEngine.Object.Instantiate<T>(T)
UnityEngine.Object.Instantiate<T>(T, UnityEngine.Vector3, UnityEngine.Quaternion)
UnityEngine.Object.Instantiate<T>(T, UnityEngine.Vector3, UnityEngine.Quaternion, UnityEngine.Transform)
UnityEngine.Object.Instantiate<T>(T, UnityEngine.Transform)
UnityEngine.Object.Instantiate<T>(T, UnityEngine.Transform, System.Boolean)
UnityEngine.Object.Destroy(UnityEngine.Object, System.Single)
UnityEngine.Object.Destroy(UnityEngine.Object)
UnityEngine.Object.DestroyImmediate(UnityEngine.Object, System.Boolean)
UnityEngine.Object.DestroyImmediate(UnityEngine.Object)
UnityEngine.Object.FindObjectsOfType(System.Type)
UnityEngine.Object.DontDestroyOnLoad(UnityEngine.Object)
UnityEngine.Object.DestroyObject(UnityEngine.Object, System.Single)
UnityEngine.Object.DestroyObject(UnityEngine.Object)
UnityEngine.Object.FindSceneObjectsOfType(System.Type)
UnityEngine.Object.FindObjectsOfTypeIncludingAssets(System.Type)
UnityEngine.Object.FindObjectsOfType<T>()
UnityEngine.Object.FindObjectOfType<T>()
UnityEngine.Object.FindObjectsOfTypeAll(System.Type)
UnityEngine.Object.FindObjectOfType(System.Type)
UnityEngine.Object.ToString()
UnityEngine.Object.name
UnityEngine.Object.hideFlags
System.Object.Equals(System.Object, System.Object)
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)

Namespace: LeTai.Asset.TranslucentImage

Assembly: LeTai.TranslucentImage.dll

Syntax

```
public class BlurConfig : ScriptableObject
```

Class Extensions

Inheritance

System.Object
Extensions

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: LeTai.Asset.TranslucentImage

Assembly: LeTai.TranslucentImage.dll

Syntax

public static class Extensions

Methods

BlitFullscreenTriangle(CommandBuffer, RenderTargetIdentifier, RenderTargetIdentifier, Material, Int32)

Declaration

public static void BlitFullscreenTriangle(this CommandBuffer cmd, RenderTargetIdentifier source, RenderTargetIdentifier destination, Material material, int pass)

Parameters

Type	Name	Description
UnityEngine.Rendering.CommandBuffer	cmd	
UnityEngine.Rendering.RenderTargetIdentifier	source	
UnityEngine.Rendering.RenderTargetIdentifier	destination	
UnityEngine.Material	material	
System.Int32	pass	

ToMinMaxVector(Rect)

Declaration

public static Vector4 ToMinMaxVector(this Rect self)

Parameters

Type	Name	Description
UnityEngine.Rect	self	

Returns

Type	Description
UnityEngine.Vector4	

ToVector4(Rect)

Declaration

public static Vector4 ToVector4(this Rect self)

Parameters

Type	Name	Description
UnityEngine.Rect	self	

Returns

Type	Description
UnityEngine.Vector4	

Interface IBlurAlgorithm

Namespace: LeTai.Asset.TranslucentImage

Assembly: LeTai.TranslucentImage.dll

Syntax

public interface IBlurAlgorithm

Methods

Blur(RenderTexture, Rect, ref RenderTexture)

Declaration

void Blur(RenderTexture source, Rect sourceCropRegion, ref RenderTexture blurredTexture)

Parameters

Type	Name	Description
UnityEngine.RenderTexture	source	
UnityEngine.Rect	sourceCropRegion	
UnityEngine.RenderTexture	blurredTexture	

Init(BlurConfig)

Declaration

void Init(BlurConfig config)

Parameters

Type	Name	Description
BlurConfig	config	

Class ScalableBlur

Inheritance

System.Object
ScalableBlur

Implements

IBlurAlgorithm

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: LeTai.Asset.TranslucentImage

Assembly: LeTai.TranslucentImage.dll

Syntax

public class ScalableBlur : IBlurAlgorithm

Methods

Blur(RenderTexture, Rect, ref RenderTexture)

Declaration

public void Blur(RenderTexture source, Rect sourceCropRegion, ref RenderTexture blurredTexture)

Parameters

Type	Name	Description
UnityEngine.RenderTexture	source	
UnityEngine.Rect	sourceCropRegion	
UnityEngine.RenderTexture	blurredTexture	

BlurAtDepth(Int32, ref RenderTexture, ref RenderTexture)

Declaration

protected virtual void BlurAtDepth(int depth, ref RenderTexture baseTexture, ref RenderTexture target)

Parameters

Type	Name	Description
System.Int32	depth	
UnityEngine.RenderTexture	baseTexture	
UnityEngine.RenderTexture	target	

ConfigMaterial(Single, Vector4)

Declaration

protected void ConfigMaterial(float radius, Vector4 cropRegion)

Parameters

Type	Name	Description
System.Single	radius	
UnityEngine.Vector4	cropRegion	

Init(BlurConfig)

Declaration

public void Init(BlurConfig config)

Parameters

Type	Name	Description
BlurConfig	config	

Implements

IBlurAlgorithm

Class ScalableBlurConfig

Inheritance

System.Object
UnityEngine.Object
UnityEngine.ScriptableObject
BlurConfig
ScalableBlurConfig

Inherited Members

UnityEngine.ScriptableObject.SetDirty()
UnityEngine.ScriptableObject.CreateInstance(System.String)
UnityEngine.ScriptableObject.CreateInstance(System.Type)
UnityEngine.ScriptableObject.CreateInstance<T>()
UnityEngine.Object.GetInstanceID()
UnityEngine.Object.GetHashCode()
UnityEngine.Object.Equals(System.Object)
UnityEngine.Object.Instantiate(UnityEngine.Object, UnityEngine.Vector3, UnityEngine.Quaternion)
UnityEngine.Object.Instantiate(UnityEngine.Object, UnityEngine.Vector3, UnityEngine.Quaternion, UnityEngine.Transform)
UnityEngine.Object.Instantiate(UnityEngine.Object)
UnityEngine.Object.Instantiate(UnityEngine.Object, UnityEngine.Transform)
UnityEngine.Object.Instantiate(UnityEngine.Object, UnityEngine.Transform, System.Boolean)
UnityEngine.Object.Instantiate<T>(T)
UnityEngine.Object.Instantiate<T>(T, UnityEngine.Vector3, UnityEngine.Quaternion)
UnityEngine.Object.Instantiate<T>(T, UnityEngine.Vector3, UnityEngine.Quaternion, UnityEngine.Transform)
UnityEngine.Object.Instantiate<T>(T, UnityEngine.Transform)
UnityEngine.Object.Instantiate<T>(T, UnityEngine.Transform, System.Boolean)
UnityEngine.Object.Destroy(UnityEngine.Object, System.Single)
UnityEngine.Object.Destroy(UnityEngine.Object)
UnityEngine.Object.DestroyImmediate(UnityEngine.Object, System.Boolean)
UnityEngine.Object.DestroyImmediate(UnityEngine.Object)
UnityEngine.Object.FindObjectsOfType(System.Type)
UnityEngine.Object.DontDestroyOnLoad(UnityEngine.Object)
UnityEngine.Object.DestroyObject(UnityEngine.Object, System.Single)
UnityEngine.Object.DestroyObject(UnityEngine.Object)
UnityEngine.Object.FindSceneObjectsOfType(System.Type)
UnityEngine.Object.FindObjectsOfTypeIncludingAssets(System.Type)
UnityEngine.Object.FindObjectsOfType<T>()
UnityEngine.Object.FindObjectOfType<T>()
UnityEngine.Object.FindObjectsOfTypeAll(System.Type)
UnityEngine.Object.FindObjectOfType(System.Type)
UnityEngine.Object.ToString()
UnityEngine.Object.name
UnityEngine.Object.hideFlags
System.Object.Equals(System.Object, System.Object)
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)

Namespace: LeTai.Asset.TranslucentImage

Assembly: LeTai.TranslucentImage.dll

Syntax

```
[CreateAssetMenu(fileName = "New Scalable Blur Config", menuName = "Translucent Image/ Scalable Blur Config")]
public class ScalableBlurConfig : BlurConfig
```

Properties

Iteration

Half the number of time to process the image. It is half because the real number of iteration must alway be even. Using half also make calculation

simpler

Declaration

```
public int Iteration { get; set; }
```

Property Value

Type	Description
System.Int32	Must be non-negative

MaxDepth

Clamp the minimum size of the intermediate texture. Reduce flickering and blur

Declaration

```
public int MaxDepth { get; set; }
```

Property Value

Type	Description
System.Int32	Must larger than 0

Radius

Distance between the base texel and the texel to be sampled.

Declaration

```
public float Radius { get; set; }
```

Property Value

Type	Description
System.Single	

Strength

User friendly property to control the amount of blur

Declaration

```
public float Strength { get; set; }
```

Property Value

Type	Description
System.Single	Must be non-negative

Methods

SetAdvancedFieldFromSimple()

Calculate size and iteration from strength

Declaration

protected virtual void SetAdvancedFieldFromSimple()

Class ShaderId

Inheritance

System.Object
ShaderId

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: LeTai.Asset.TranslucentImage

Assembly: LeTai.TranslucentImage.dll

Syntax

public static class ShaderId

Fields

CROP_REGION

Declaration

public static readonly int CROP_REGION

Field Value

Type	Description
System.Int32	

RADIUS

Declaration

public static readonly int RADIUS

Field Value

Type	Description
System.Int32	

Class TranslucentImage

Dynamic blur-behind UI element

Inheritance

System.Object
UnityEngine.Object
UnityEngine.Component
UnityEngine.Behaviour
UnityEngine.MonoBehaviour
UnityEngine.EventSystems.UIBehaviour
UnityEngine.UI.Graphic
UnityEngine.UI.MaskableGraphic
UnityEngine.UI.Image
TranslucentImage

Implements

UnityEngine.UI.ICanvasElement
UnityEngine.UI.IClippable
UnityEngine.UI.IMaskable
UnityEngine.UI.IMaterialModifier
UnityEngine.ISerializationCallbackReceiver
UnityEngine.UI.ILayoutElement
UnityEngine.ICanvasRaycastFilter
UnityEngine.UI.IMeshModifier

Inherited Members

UnityEngine.UI.Image.s_ETC1DefaultUI
UnityEngine.UI.Image.DisableSpriteOptimizations()
UnityEngine.UI.Image.OnBeforeSerialize()
UnityEngine.UI.Image.OnAfterDeserialize()
UnityEngine.UI.Image.SetNativeSize()
UnityEngine.UI.Image.OnPopulateMesh(UnityEngine.UI.VertexHelper)
UnityEngine.UI.Image.UpdateMaterial()
UnityEngine.UI.Image.OnCanvasHierarchyChanged()
UnityEngine.UI.Image.CalculateLayoutInputHorizontal()
UnityEngine.UI.Image.CalculateLayoutInputVertical()
UnityEngine.UI.Image.IsRaycastLocationValid(UnityEngine.Vector2, UnityEngine.Camera)
UnityEngine.UI.Image.OnValidate()
UnityEngine.UI.Image.sprite
UnityEngine.UI.Image.overrideSprite
UnityEngine.UI.Image.type
UnityEngine.UI.Image.preserveAspect
UnityEngine.UI.Image.fillCenter
UnityEngine.UI.Image.fillMethod
UnityEngine.UI.Image.fillAmount
UnityEngine.UI.Image.fillClockwise
UnityEngine.UI.Image.fillOrigin
UnityEngine.UI.Image.eventAlphaThreshold
UnityEngine.UI.Image.alphaHitTestMinimumThreshold
UnityEngine.UI.Image.useSpriteMesh
UnityEngine.UI.Image.defaultETC1GraphicMaterial
UnityEngine.UI.Image.mainTexture
UnityEngine.UI.Image.hasBorder
UnityEngine.UI.Image.pixelsPerUnitMultiplier
UnityEngine.UI.Image.pixelsPerUnit
UnityEngine.UI.Image.multipliedPixelsPerUnit
UnityEngine.UI.Image.material
UnityEngine.UI.Image.minWidth
UnityEngine.UI.Image.preferredWidth
UnityEngine.UI.Image.flexibleWidth
UnityEngine.UI.Image.minHeight
UnityEngine.UI.Image.preferredHeight
UnityEngine.UI.Image.flexibleHeight
UnityEngine.UI.Image.layoutPriority
UnityEngine.UI.MaskableGraphic.m_ShouldRecalculateStencil
UnityEngine.UI.MaskableGraphic.m_MaskMaterial
UnityEngine.UI.MaskableGraphic.m_StencilValue

UnityEngine.UI.MaskableGraphic.GetModifiedMaterial(UnityEngine.Material)
UnityEngine.UI.MaskableGraphic.Cull(UnityEngine.Rect, System.Boolean)
UnityEngine.UI.MaskableGraphic.SetClipRect(UnityEngine.Rect, System.Boolean)
UnityEngine.UI.MaskableGraphic.SetClipSoftness(UnityEngine.Vector2)
UnityEngine.UI.MaskableGraphic.OnTransformParentChanged()
UnityEngine.UI.MaskableGraphic.RecalculateClipping()
UnityEngine.UI.MaskableGraphic.RecalculateMasking()
UnityEngine.UI.MaskableGraphic.UnityEngine.UI.IClippable.get_gameObject()
UnityEngine.UI.MaskableGraphic.onCullStateChanged
UnityEngine.UI.MaskableGraphic.maskable
UnityEngine.UI.MaskableGraphic.isMaskingGraphic
UnityEngine.UI.Graphic.s_DefaultUI
UnityEngine.UI.Graphic.s_WhiteTexture
UnityEngine.UI.Graphic.m_Material
UnityEngine.UI.Graphic.m_SkipLayoutUpdate
UnityEngine.UI.Graphic.m_SkipMaterialUpdate
UnityEngine.UI.Graphic.m_OnDirtyLayoutCallback
UnityEngine.UI.Graphic.m_OnDirtyVertsCallback
UnityEngine.UI.Graphic.m_OnDirtyMaterialCallback
UnityEngine.UI.Graphic.s_Mesh
UnityEngine.UI.Graphic.m_CachedMesh
UnityEngine.UI.Graphic.m_CachedUvs
UnityEngine.UI.Graphic.SetAllDirty()
UnityEngine.UI.Graphic.SetLayoutDirty()
UnityEngine.UI.Graphic.SetVerticesDirty()
UnityEngine.UI.Graphic.SetMaterialDirty()
UnityEngine.UI.Graphic.OnRectTransformDimensionsChange()
UnityEngine.UI.Graphic.OnBeforeTransformParentChanged()
UnityEngine.UI.Graphic.OnDestroy()
UnityEngine.UI.Graphic.OnCullingChanged()
UnityEngine.UI.Graphic.Rebuild(UnityEngine.UI.CanvasUpdate)
UnityEngine.UI.Graphic.LayoutComplete()
UnityEngine.UI.Graphic.GraphicUpdateComplete()
UnityEngine.UI.Graphic.UpdateGeometry()
UnityEngine.UI.Graphic.OnPopulateMesh(UnityEngine.Mesh)
UnityEngine.UI.Graphic.OnRebuildRequested()
UnityEngine.UI.Graphic.Reset()
UnityEngine.UI.Graphic.Raycast(UnityEngine.Vector2, UnityEngine.Camera)
UnityEngine.UI.Graphic.PixelAdjustPoint(UnityEngine.Vector2)
UnityEngine.UI.Graphic.GetPixelAdjustedRect()
UnityEngine.UI.Graphic.CrossFadeColor(UnityEngine.Color, System.Single, System.Boolean, System.Boolean)
UnityEngine.UI.Graphic.CrossFadeColor(UnityEngine.Color, System.Single, System.Boolean, System.Boolean, System.Boolean)
UnityEngine.UI.Graphic.CrossFadeAlpha(System.Single, System.Single, System.Boolean)
UnityEngine.UI.Graphic.RegisterDirtyLayoutCallback(UnityEngine.Events.UnityAction)
UnityEngine.UI.Graphic.UnregisterDirtyLayoutCallback(UnityEngine.Events.UnityAction)
UnityEngine.UI.Graphic.RegisterDirtyVerticesCallback(UnityEngine.Events.UnityAction)
UnityEngine.UI.Graphic.UnregisterDirtyVerticesCallback(UnityEngine.Events.UnityAction)
UnityEngine.UI.Graphic.RegisterDirtyMaterialCallback(UnityEngine.Events.UnityAction)
UnityEngine.UI.Graphic.UnregisterDirtyMaterialCallback(UnityEngine.Events.UnityAction)
UnityEngine.UI.Graphic.UnityEngine.UI.ICanvasElement.get_transform()
UnityEngine.UI.Graphic.defaultGraphicMaterial
UnityEngine.UI.Graphic.color
UnityEngine.UI.Graphic.raycastTarget
UnityEngine.UI.Graphic.useLegacyMeshGeneration
UnityEngine.UI.Graphic.depth
UnityEngine.UI.Graphic.rectTransform
UnityEngine.UI.Graphic.canvas
UnityEngine.UI.Graphic.canvasRenderer
UnityEngine.UI.Graphic.defaultMaterial
UnityEngine.UI.Graphic.materialForRendering
UnityEngine.UI.Graphic.workerMesh
UnityEngine.EventSystems.UIBehaviour.Awake()
UnityEngine.EventSystems.UIBehaviour.IsActive()
UnityEngine.EventSystems.UIBehaviour.OnCanvasGroupChanged()
UnityEngine.EventSystems.UIBehaviour.IsDestroyed()
UnityEngine.MonoBehaviour.IsInvoking()
UnityEngine.MonoBehaviour.CancelInvoke()
UnityEngine.MonoBehaviour.Invoke(System.String, System.Single)
UnityEngine.MonoBehaviour.InvokeRepeating(System.String, System.Single, System.Single)
UnityEngine.MonoBehaviour.CancelInvoke(System.String)
UnityEngine.MonoBehaviour.IsInvoking(System.String)
UnityEngine.MonoBehaviour.StartCoroutine(System.String)

UnityEngine.MonoBehaviour.StartCoroutine(System.String, System.Object)
UnityEngine.MonoBehaviour.StartCoroutine(System.Collections.IEnumerator)
UnityEngine.MonoBehaviour.StartCoroutine_Auto(System.Collections.IEnumerator)
UnityEngine.MonoBehaviour.StopCoroutine(System.Collections.IEnumerator)
UnityEngine.MonoBehaviour.StopCoroutine(UnityEngine.Coroutine)
UnityEngine.MonoBehaviour.StopCoroutine(System.String)
UnityEngine.MonoBehaviour.StopAllCoroutines()
UnityEngine.MonoBehaviour.print(System.Object)
UnityEngine.MonoBehaviour.useGUILayout
UnityEngine.MonoBehaviour.runInEditMode
UnityEngine.Behaviour.enabled
UnityEngine.Behaviour.isActiveAndEnabled
UnityEngine.Component.GetComponent(System.Type)
UnityEngine.Component.GetComponent<T>()
UnityEngine.Component.TryGetComponent(System.Type, UnityEngine.Component)
UnityEngine.Component.TryGetComponent<T>(T)
UnityEngine.Component.GetComponent(System.String)
UnityEngine.Component.GetComponentInChildren(System.Type, System.Boolean)
UnityEngine.Component.GetComponentInChildren(System.Type)
UnityEngine.Component.GetComponentInChildren<T>(System.Boolean)
UnityEngine.Component.GetComponentInChildren<T>()
UnityEngine.Component.GetComponentsInChildren(System.Type, System.Boolean)
UnityEngine.Component.GetComponentsInChildren(System.Type)
UnityEngine.Component.GetComponentsInChildren<T>(System.Boolean)
UnityEngine.Component.GetComponentsInChildren<T>(System.Boolean, System.Collections.Generic.List<T>)
UnityEngine.Component.GetComponentsInChildren<T>()
UnityEngine.Component.GetComponentsInChildren<T>(System.Collections.Generic.List<T>)
UnityEngine.Component.GetComponentInParent(System.Type)
UnityEngine.Component.GetComponentInParent<T>()
UnityEngine.Component.GetComponentsInParent(System.Type, System.Boolean)
UnityEngine.Component.GetComponentsInParent(System.Type)
UnityEngine.Component.GetComponentsInParent<T>(System.Boolean)
UnityEngine.Component.GetComponentsInParent<T>(System.Boolean, System.Collections.Generic.List<T>)
UnityEngine.Component.GetComponentsInParent<T>()
UnityEngine.Component.GetComponents(System.Type)
UnityEngine.Component.GetComponents(System.Type, System.Collections.Generic.List<UnityEngine.Component>)
UnityEngine.Component.GetComponents<T>(System.Collections.Generic.List<T>)
UnityEngine.Component.GetComponents<T>()
UnityEngine.Component.CompareTag(System.String)
UnityEngine.Component.SendMessageUpwards(System.String, System.Object, UnityEngine.SendMessageOptions)
UnityEngine.Component.SendMessageUpwards(System.String, System.Object)
UnityEngine.Component.SendMessageUpwards(System.String)
UnityEngine.Component.SendMessageUpwards(System.String, UnityEngine.SendMessageOptions)
UnityEngine.Component.SendMessage(System.String, System.Object)
UnityEngine.Component.SendMessage(System.String)
UnityEngine.Component.SendMessage(System.String, System.Object, UnityEngine.SendMessageOptions)
UnityEngine.Component.SendMessage(System.String, UnityEngine.SendMessageOptions)
UnityEngine.Component.BroadcastMessage(System.String, System.Object, UnityEngine.SendMessageOptions)
UnityEngine.Component.BroadcastMessage(System.String, System.Object)
UnityEngine.Component.BroadcastMessage(System.String)
UnityEngine.Component.BroadcastMessage(System.String, UnityEngine.SendMessageOptions)
UnityEngine.Component.transform
UnityEngine.Component.gameObject
UnityEngine.Component.tag
UnityEngine.Object.GetInstanceID()
UnityEngine.Object.GetHashCode()
UnityEngine.Object.Equals(System.Object)
UnityEngine.Object.Instantiate(UnityEngine.Object, UnityEngine.Vector3, UnityEngine.Quaternion)
UnityEngine.Object.Instantiate(UnityEngine.Object, UnityEngine.Vector3, UnityEngine.Quaternion, UnityEngine.Transform)
UnityEngine.Object.Instantiate(UnityEngine.Object)
UnityEngine.Object.Instantiate(UnityEngine.Object, UnityEngine.Transform)
UnityEngine.Object.Instantiate(UnityEngine.Object, UnityEngine.Transform, System.Boolean)
UnityEngine.Object.Instantiate<T>(T)
UnityEngine.Object.Instantiate<T>(T, UnityEngine.Vector3, UnityEngine.Quaternion)
UnityEngine.Object.Instantiate<T>(T, UnityEngine.Vector3, UnityEngine.Quaternion, UnityEngine.Transform)
UnityEngine.Object.Instantiate<T>(T, UnityEngine.Transform)
UnityEngine.Object.Instantiate<T>(T, UnityEngine.Transform, System.Boolean)
UnityEngine.Object.Destroy(UnityEngine.Object, System.Single)
UnityEngine.Object.Destroy(UnityEngine.Object)
UnityEngine.Object.DestroyImmediate(UnityEngine.Object, System.Boolean)
UnityEngine.Object.DestroyImmediate(UnityEngine.Object)
UnityEngine.Object.FindObjectsOfType(System.Type)

UnityEngine.Object.DontDestroyOnLoad(UnityEngine.Object)
UnityEngine.Object.DestroyObject(UnityEngine.Object, System.Single)
UnityEngine.Object.DestroyObject(UnityEngine.Object)
UnityEngine.Object.FindSceneObjectsOfType(System.Type)
UnityEngine.Object.FindObjectsOfTypeIncludingAssets(System.Type)
UnityEngine.Object.FindObjectsOfType<T>()
UnityEngine.Object.FindObjectOfType<T>()
UnityEngine.Object.FindObjectsOfTypeAll(System.Type)
UnityEngine.Object.FindObjectOfType(System.Type)
UnityEngine.Object.ToString()
UnityEngine.Object.name
UnityEngine.Object.hideFlags
System.Object.Equals(System.Object, System.Object)
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)

Namespace: LeTai.Asset.TranslucentImage

Assembly: LeTai.TranslucentImage.dll

Syntax

```
[HelpURL("https://leloctai.com/asset/translucentimage/docs/articles/customize.html#translucent-image")]
public class TranslucentImage : Image, ICanvasElement, IClippable, IMaskable, IMaterialModifier, ISerializationCallbackReceiver, ILayoutElement, ICanvasRaycastFilter, IMeshModifier
```

Fields

brightness

Brighten/darken them image

Declaration

```
[Tooltip("Brighten/darken them image")]
[Range(-1F, 1F)]
public float brightness
```

Field Value

Type	Description
System.Single	

flatten

Flatten the color behind to help keep contrast on varying background

Declaration

```
[Tooltip("Flatten the color behind to help keep contrast on varying background")]
[Range(0F, 1F)]
public float flatten
```

Field Value

Type	Description
System.Single	

m_spriteBlending

Declaration

```
[Tooltip("Blend between the sprite and background blur")]
[Range(0F, 1F)]
[FormerlySerializedAs("spriteBlending")]
public float m_spriteBlending
```

Field Value

Type	Description
System.Single	

source

Source of blur for this image

Declaration

public TranslucentImageSource source

Field Value

Type	Description
TranslucentImageSource	

vibrancy

(De)Saturate them image, 1 is normal, 0 is grey scale, below zero make the image negative

Declaration

```
[Tooltip("(De)Saturate them image, 1 is normal, 0 is black and white, below zero make the image negative")]
[Range(-1F, 3F)]
public float vibrancy
```

Field Value

Type	Description
System.Single	

Properties

spriteBlending

Declaration

public float spriteBlending { get; set; }

Property Value

Type	Description
System.Single	

Methods

ModifyMesh(Mesh)

Declaration

public virtual void ModifyMesh(Mesh mesh)

Parameters

Type	Name	Description
UnityEngine.Mesh	mesh	

ModifyMesh(VertexHelper)

Declaration

public virtual void ModifyMesh(VertexHelper vh)

Parameters

Type	Name	Description
UnityEngine.UI.VertexHelper	vh	

OnDidApplyAnimationProperties()

Declaration

protected override void OnDidApplyAnimationProperties()

Overrides

UnityEngine.UI.Image.OnDidApplyAnimationProperties()

OnDisable()

Declaration

protected override void OnDisable()

Overrides

UnityEngine.UI.Image.OnDisable()

OnEnable()

Declaration

protected override void OnEnable()

Overrides

UnityEngine.UI.Image.OnEnable()

Start()

Declaration

protected override void Start()

Overrides

UnityEngine.EventSystems.UIBehaviour.Start()

Implements

UnityEngine.UI.ICanvasElement
UnityEngine.UI.IClippable
UnityEngine.UI.IMaskable
UnityEngine.UI.IMaterialModifier
UnityEngine.ISerializationCallbackReceiver
UnityEngine.UI.ILayoutElement
UnityEngine.ICanvasRaycastFilter
UnityEngine.UI.IMeshModifier

Class TranslucentImageSource

Common source of blur for Translucent Images.

Inheritance

System.Object
UnityEngine.Object
UnityEngine.Component
UnityEngine.Behaviour
UnityEngine.MonoBehaviour
TranslucentImageSource

Inherited Members

UnityEngine.MonoBehaviour.IsInvoking()
UnityEngine.MonoBehaviour.CancelInvoke()
UnityEngine.MonoBehaviour.Invoke(System.String, System.Single)
UnityEngine.MonoBehaviour.InvokeRepeating(System.String, System.Single, System.Single)
UnityEngine.MonoBehaviour.CancelInvoke(System.String)
UnityEngine.MonoBehaviour.IsInvoking(System.String)
UnityEngine.MonoBehaviour.StartCoroutine(System.String)
UnityEngine.MonoBehaviour.StartCoroutine(System.String, System.Object)
UnityEngine.MonoBehaviour.StartCoroutine(System.Collections.IEnumerator)
UnityEngine.MonoBehaviour.StartCoroutine_Auto(System.Collections.IEnumerator)
UnityEngine.MonoBehaviour.StopCoroutine(System.Collections.IEnumerator)
UnityEngine.MonoBehaviour.StopCoroutine(UnityEngine.Coroutine)
UnityEngine.MonoBehaviour.StopCoroutine(System.String)
UnityEngine.MonoBehaviour.StopAllCoroutines()
UnityEngine.MonoBehaviour.print(System.Object)
UnityEngine.MonoBehaviour.useGUILayout
UnityEngine.MonoBehaviour.runInEditMode
UnityEngine.Behaviour.enabled
UnityEngine.Behaviour.isActiveAndEnabled
UnityEngine.Component.GetComponent(System.Type)
UnityEngine.Component.GetComponent<T>()
UnityEngine.Component.TryGetComponent(System.Type, UnityEngine.Component)
UnityEngine.Component.TryGetComponent<T>(T)
UnityEngine.Component.GetComponent(System.String)
UnityEngine.Component.GetComponentInChildren(System.Type, System.Boolean)
UnityEngine.Component.GetComponentInChildren(System.Type)
UnityEngine.Component.GetComponentInChildren<T>(System.Boolean)
UnityEngine.Component.GetComponentInChildren<T>()
UnityEngine.Component.GetComponentsInChildren(System.Type, System.Boolean)
UnityEngine.Component.GetComponentsInChildren(System.Type)
UnityEngine.Component.GetComponentsInChildren<T>(System.Boolean)
UnityEngine.Component.GetComponentsInChildren<T>(System.Boolean, System.Collections.Generic.List<T>)
UnityEngine.Component.GetComponentsInChildren<T>()
UnityEngine.Component.GetComponentsInChildren<T>(System.Collections.Generic.List<T>)
UnityEngine.Component.GetComponentInParent(System.Type)
UnityEngine.Component.GetComponentInParent<T>()
UnityEngine.Component.GetComponentsInParent(System.Type, System.Boolean)
UnityEngine.Component.GetComponentsInParent(System.Type)
UnityEngine.Component.GetComponentsInParent<T>(System.Boolean)
UnityEngine.Component.GetComponentsInParent<T>(System.Boolean, System.Collections.Generic.List<T>)
UnityEngine.Component.GetComponentsInParent<T>()
UnityEngine.Component.GetComponents(System.Type)
UnityEngine.Component.GetComponents(System.Type, System.Collections.Generic.List<UnityEngine.Component>)
UnityEngine.Component.GetComponents<T>(System.Collections.Generic.List<T>)
UnityEngine.Component.GetComponents<T>()
UnityEngine.Component.CompareTag(System.String)
UnityEngine.Component.SendMessageUpwards(System.String, System.Object, UnityEngine.SendMessageOptions)
UnityEngine.Component.SendMessageUpwards(System.String, System.Object)
UnityEngine.Component.SendMessageUpwards(System.String)
UnityEngine.Component.SendMessageUpwards(System.String, UnityEngine.SendMessageOptions)

UnityEngine.Component.SendMessage(System.String, System.Object)
UnityEngine.Component.SendMessage(System.String)
UnityEngine.Component.SendMessage(System.String, System.Object, UnityEngine.SendMessageOptions)
UnityEngine.Component.SendMessage(System.String, UnityEngine.SendMessageOptions)
UnityEngine.Component.BroadcastMessage(System.String, System.Object, UnityEngine.SendMessageOptions)
UnityEngine.Component.BroadcastMessage(System.String, System.Object)
UnityEngine.Component.BroadcastMessage(System.String)
UnityEngine.Component.BroadcastMessage(System.String, UnityEngine.SendMessageOptions)
UnityEngine.Component.transform
UnityEngine.Component.gameObject
UnityEngine.Component.tag
UnityEngine.Object.GetInstanceID()
UnityEngine.Object.GetHashCode()
UnityEngine.Object.Equals(System.Object)
UnityEngine.Object.Instantiate(UnityEngine.Object, UnityEngine.Vector3, UnityEngine.Quaternion)
UnityEngine.Object.Instantiate(UnityEngine.Object, UnityEngine.Vector3, UnityEngine.Quaternion, UnityEngine.Transform)
UnityEngine.Object.Instantiate(UnityEngine.Object)
UnityEngine.Object.Instantiate(UnityEngine.Object, UnityEngine.Transform)
UnityEngine.Object.Instantiate(UnityEngine.Object, UnityEngine.Transform, System.Boolean)
UnityEngine.Object.Instantiate<T>(T)
UnityEngine.Object.Instantiate<T>(T, UnityEngine.Vector3, UnityEngine.Quaternion)
UnityEngine.Object.Instantiate<T>(T, UnityEngine.Vector3, UnityEngine.Quaternion, UnityEngine.Transform)
UnityEngine.Object.Instantiate<T>(T, UnityEngine.Transform)
UnityEngine.Object.Instantiate<T>(T, UnityEngine.Transform, System.Boolean)
UnityEngine.Object.Destroy(UnityEngine.Object, System.Single)
UnityEngine.Object.Destroy(UnityEngine.Object)
UnityEngine.Object.DestroyImmediate(UnityEngine.Object, System.Boolean)
UnityEngine.Object.DestroyImmediate(UnityEngine.Object)
UnityEngine.Object.FindObjectsOfType(System.Type)
UnityEngine.Object.DontDestroyOnLoad(UnityEngine.Object)
UnityEngine.Object.DestroyObject(UnityEngine.Object, System.Single)
UnityEngine.Object.DestroyObject(UnityEngine.Object)
UnityEngine.Object.FindSceneObjectsOfType(System.Type)
UnityEngine.Object.FindObjectsOfTypeIncludingAssets(System.Type)
UnityEngine.Object.FindObjectsOfType<T>()
UnityEngine.Object.FindObjectOfType<T>()
UnityEngine.Object.FindObjectsOfTypeAll(System.Type)
UnityEngine.Object.FindObjectOfType(System.Type)
UnityEngine.Object.ToString()
UnityEngine.Object.name
UnityEngine.Object.hideFlags
System.Object.Equals(System.Object, System.Object)
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)

Namespace: LeTai.Asset.TranslucentImage

Assembly: LeTai.TranslucentImage.dll

Syntax

```
[ExecuteAlways]
[RequireComponent(typeof(Camera))]
[AddComponentMenu("Image Effects/Tai Le Assets/Translucent Image Source")]
[HelpURL("https://leloctai.com/asset/translucentimage/docs/articles/customize.html#translucent-image-source")]
public class TranslucentImageSource : MonoBehaviour
```

Remarks

It is an Image effect that blur the render target of the Camera it attached to, then save the result to a global read-only Render Texture

Fields

maxUpdateRate

Maximum number of times to update the blurred image each second

Declaration

public float maxUpdateRate

Field Value

Type	Description
System.Single	

preview

Render the blurred result to the render target

Declaration

public bool preview

Field Value

Type	Description
System.Boolean	

Properties

BlurAlgorithmSelection

Declaration

public BlurAlgorithmType BlurAlgorithmSelection { get; set; }

Property Value

Type	Description
BlurAlgorithmType	

BlurConfig

Declaration

public BlurConfig BlurConfig { get; set; }

Property Value

Type	Description
BlurConfig	

BlurredScreen

Result of the image effect. Translucent Image use this as their content (read-only)

Declaration

public RenderTexture BlurredScreen { get; set; }

Property Value

Type	Description
UnityEngine.RenderTexture	

BlurRegion

Define the rectangular area on screen that will be blurred.

Declaration

public Rect BlurRegion { get; set; }

Property Value

Type	Description
UnityEngine.Rect	Between 0 and 1

BlurRegionNormalizedScreenSpace

Declaration

public Rect BlurRegionNormalizedScreenSpace { get; set; }

Property Value

Type	Description
UnityEngine.Rect	

Downsample

The rendered image will be shrinked by a factor of 2^{{this}} before bluring to reduce processing time

Declaration

public int Downsample { get; set; }

Property Value

Type	Description
System.Int32	Must be non-negative. Default to 0

Methods

CreateNewBlurredScreen()

Declaration

protected virtual void CreateNewBlurredScreen()

OnBeforeBlur()

Declaration

public void OnBeforeBlur()

OnRenderImage(RenderTexture, RenderTexture)

Declaration

protected virtual void OnRenderImage(RenderTexture source, RenderTexture destination)

Parameters

Type	Name	Description
UnityEngine.RenderTexture	source	
UnityEngine.RenderTexture	destination	

shouldUpdateBlur()

Declaration

public bool shouldUpdateBlur()

Returns

Type	Description
System.Boolean	

Start()

Declaration

protected virtual void Start()

Namespace LeTai.Asset.TranslucentImage.UniversalRP

Classes

Extensions

ScalableBlur

ShaderId

TranslucentImageBlurRenderPass

TranslucentImageBlurSource

Utilities

Interfaces

IBlurAlgorithm

Enums

BlurAlgorithmType

Enum BlurAlgorithmType

Namespace: LeTai.Asset.TranslucentImage.UniversalRP

Assembly: LeTai.TranslucentImage.UniversalRP.dll

Syntax

```
[MovedFrom("LeTai.Asset.TranslucentImage.LWRP")]
public enum BlurAlgorithmType
```

Fields

Name	Description
ScalableBlur	

Class Extensions

Inheritance

System.Object
Extensions

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: LeTai.Asset.TranslucentImage.UniversalRP

Assembly: LeTai.TranslucentImage.UniversalRP.dll

Syntax

public static class Extensions

Methods

BlitProcedural(CommandBuffer, RenderTargetIdentifier, RenderTargetIdentifier, Material, Int32)

Declaration

public static void BlitProcedural(this CommandBuffer cmd, RenderTargetIdentifier source, RenderTargetIdentifier destination, Material material, int passIndex)

Parameters

Type	Name	Description
UnityEngine.Rendering.CommandBuffer	cmd	
UnityEngine.Rendering.RenderTargetIdentifier	source	
UnityEngine.Rendering.RenderTargetIdentifier	destination	
UnityEngine.Material	material	
System.Int32	passIndex	

Interface IBlurAlgorithm

Namespace: LeTai.Asset.TranslucentImage.UniversalRP

Assembly: LeTai.TranslucentImage.UniversalRP.dll

Syntax

```
[MovedFrom("LeTai.Asset.TranslucentImage.LWRP")]
public interface IBlurAlgorithm
```

Methods

Blur(CommandBuffer, RenderTargetIdentifier, Rect, RenderTexture)

Declaration

```
void Blur(CommandBuffer cmd, RenderTargetIdentifier src, Rect srcCropRegion, RenderTexture target)
```

Parameters

Type	Name	Description
UnityEngine.Rendering.CommandBuffer	cmd	
UnityEngine.Rendering.RenderTargetIdentifier	src	
UnityEngine.Rect	srcCropRegion	
UnityEngine.RenderTexture	target	

Init(BlurConfig)

Declaration

```
void Init(BlurConfig config)
```

Parameters

Type	Name	Description
BlurConfig	config	

Class ScalableBlur

Inheritance

System.Object
ScalableBlur

Implements

IBlurAlgorithm

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: LeTai.Asset.TranslucentImage.UniversalRP

Assembly: LeTai.TranslucentImage.UniversalRP.dll

Syntax

```
[MovedFrom("LeTai.Asset.TranslucentImage.LWRP")]
public class ScalableBlur : IBlurAlgorithm
```

Methods

Blur(CommandBuffer, RenderTargetIdentifier, Rect, RenderTexture)

Declaration

public void Blur(CommandBuffer cmd, RenderTargetIdentifier src, Rect srcCropRegion, RenderTexture target)

Parameters

Type	Name	Description
UnityEngine.Rendering.CommandBuffer	cmd	
UnityEngine.Rendering.RenderTargetIdentifier	src	
UnityEngine.Rect	srcCropRegion	
UnityEngine.RenderTexture	target	

BlurAtDepth(CommandBuffer, Int32, RenderTexture)

Declaration

protected virtual void BlurAtDepth(CommandBuffer cmd, int depth, RenderTexture baseTexture)

Parameters

Type	Name	Description
UnityEngine.Rendering.CommandBuffer	cmd	
System.Int32	depth	
UnityEngine.RenderTexture	baseTexture	

ConfigMaterial(Single, Vector4)

Declaration

protected void ConfigMaterial(float radius, Vector4 cropRegion)

Parameters

Type	Name	Description
System.Single	radius	
UnityEngine.Vector4	cropRegion	

Init(BlurConfig)

Declaration

public void Init(BlurConfig config)

Parameters

Type	Name	Description
BlurConfig	config	

Implements

IBlurAlgorithm

Class ShaderId

Inheritance

System.Object
ShaderId

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: LeTai.Asset.TranslucentImage.UniversalRP

Assembly: LeTai.TranslucentImage.UniversalRP.dll

Syntax

public static class ShaderId

Fields

intermediateRT

Declaration

public static int[] intermediateRT

Field Value

Type	Description
System.Int32[]	

MAIN_TEX

Declaration

public static readonly int MAIN_TEX

Field Value

Type	Description
System.Int32	

Methods

Init(Int32)

Declaration

public static void Init(int stackDepth)

Parameters

Type	Name	Description
System.Int32	stackDepth	

Class TranslucentImageBlurRenderPass

Inheritance

System.Object
UnityEngine.Rendering.Universal.ScriptableRenderPass
TranslucentImageBlurRenderPass

Inherited Members

UnityEngine.Rendering.Universal.ScriptableRenderPass.ConfigureTarget(UnityEngine.Rendering.RenderTargetIdentifier, UnityEngine.Rendering.RenderTargetIdentifier)
UnityEngine.Rendering.Universal.ScriptableRenderPass.ConfigureTarget(UnityEngine.Rendering.RenderTargetIdentifier[], UnityEngine.Rendering.RenderTargetIdentifier)
UnityEngine.Rendering.Universal.ScriptableRenderPass.ConfigureTarget(UnityEngine.Rendering.RenderTargetIdentifier)
UnityEngine.Rendering.Universal.ScriptableRenderPass.ConfigureTarget(UnityEngine.Rendering.RenderTargetIdentifier[])
UnityEngine.Rendering.Universal.ScriptableRenderPass.ConfigureClear(UnityEngine.Rendering.ClearFlag, UnityEngine.Color)
UnityEngine.Rendering.Universal.ScriptableRenderPass.Configure(UnityEngine.Rendering.CommandBuffer, UnityEngine.RenderTextureDescriptor)
UnityEngine.Rendering.Universal.ScriptableRenderPass.FrameCleanup(UnityEngine.Rendering.CommandBuffer)
UnityEngine.Rendering.Universal.ScriptableRenderPass.Blit(UnityEngine.Rendering.CommandBuffer, UnityEngine.Rendering.RenderTargetIdentifier, UnityEngine.Rendering.RenderTargetIdentifier, UnityEngine.Material, System.Int32)
UnityEngine.Rendering.Universal.ScriptableRenderPass.RenderPostProcessing(UnityEngine.Rendering.CommandBuffer, UnityEngine.Rendering.Universal.CameraData, UnityEngine.RenderTextureDescriptor, UnityEngine.Rendering.RenderTargetIdentifier, UnityEngine.Rendering.RenderTargetIdentifier, System.Boolean, System.Boolean)
UnityEngine.Rendering.Universal.ScriptableRenderPass.CreateDrawingSettings(UnityEngine.Rendering.ShaderTagId, UnityEngine.Rendering.Universal.RenderingData, UnityEngine.Rendering.SortingCriteria)
UnityEngine.Rendering.Universal.ScriptableRenderPass.CreateDrawingSettings(System.Collections.Generic.List<UnityEngine.Rendering.ShaderTagId>, UnityEngine.Rendering.Universal.RenderingData, UnityEngine.Rendering.SortingCriteria)
UnityEngine.Rendering.Universal.ScriptableRenderPass.renderPassEvent
UnityEngine.Rendering.Universal.ScriptableRenderPass.colorAttachments
UnityEngine.Rendering.Universal.ScriptableRenderPass.colorAttachment
UnityEngine.Rendering.Universal.ScriptableRenderPass.depthAttachment
UnityEngine.Rendering.Universal.ScriptableRenderPass.clearFlag
UnityEngine.Rendering.Universal.ScriptableRenderPass.clearColor
System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: LeTai.Asset.TranslucentImage.UniversalRP

Assembly: LeTai.TranslucentImage.UniversalRP.dll

Syntax

```
[MovedFrom("LeTai.Asset.TranslucentImage.LWRP")]
public class TranslucentImageBlurRenderPass : ScriptableRenderPass
```

Constructors

TranslucentImageBlurRenderPass()

Declaration

```
public TranslucentImageBlurRenderPass()
```

Properties

PreviewMaterial

Declaration

```
public Material PreviewMaterial { get; }
```

Property Value

Type	Description
UnityEngine.Material	

Methods

Execute(ScriptableRenderContext, ref RenderingData)

Declaration

public override void Execute(ScriptableRenderContext context, ref RenderingData renderingData)

Parameters

Type	Name	Description
UnityEngine.Rendering.ScriptableRenderContext	context	
UnityEngine.Rendering.Universal.RenderingData	renderingData	

Overrides

UnityEngine.Rendering.Universal.ScriptableRenderPass.Execute(UnityEngine.Rendering.ScriptableRenderContext, UnityEngine.Rendering.Universal.RenderingData)

Finalize()

Declaration

protected void Finalize()

Class TranslucentImageBlurSource

Inheritance

System.Object
UnityEngine.Object
UnityEngine.ScriptableObject
UnityEngine.Rendering.Universal.ScriptableRendererFeature
TranslucentImageBlurSource

Inherited Members

UnityEngine.Rendering.Universal.ScriptableRendererFeature.SetActive(System.Boolean)
UnityEngine.Rendering.Universal.ScriptableRendererFeature.isActive
UnityEngine.ScriptableObject.SetDirty()
UnityEngine.ScriptableObject.CreateInstance(System.String)
UnityEngine.ScriptableObject.CreateInstance(System.Type)
UnityEngine.ScriptableObject.CreateInstance<T>()
UnityEngine.Object.GetInstanceID()
UnityEngine.Object.GetHashCode()
UnityEngine.Object.Equals(System.Object)
UnityEngine.Object.Instantiate(UnityEngine.Object, UnityEngine.Vector3, UnityEngine.Quaternion)
UnityEngine.Object.Instantiate(UnityEngine.Object, UnityEngine.Vector3, UnityEngine.Quaternion, UnityEngine.Transform)
UnityEngine.Object.Instantiate(UnityEngine.Object)
UnityEngine.Object.Instantiate(UnityEngine.Object, UnityEngine.Transform)
UnityEngine.Object.Instantiate(UnityEngine.Object, UnityEngine.Transform, System.Boolean)
UnityEngine.Object.Instantiate<T>(T)
UnityEngine.Object.Instantiate<T>(T, UnityEngine.Vector3, UnityEngine.Quaternion)
UnityEngine.Object.Instantiate<T>(T, UnityEngine.Vector3, UnityEngine.Quaternion, UnityEngine.Transform)
UnityEngine.Object.Instantiate<T>(T, UnityEngine.Transform)
UnityEngine.Object.Instantiate<T>(T, UnityEngine.Transform, System.Boolean)
UnityEngine.Object.Destroy(UnityEngine.Object, System.Single)
UnityEngine.Object.Destroy(UnityEngine.Object)
UnityEngine.Object.DestroyImmediate(UnityEngine.Object, System.Boolean)
UnityEngine.Object.DestroyImmediate(UnityEngine.Object)
UnityEngine.Object.FindObjectsOfType(System.Type)
UnityEngine.Object.DontDestroyOnLoad(UnityEngine.Object)
UnityEngine.Object.DestroyObject(UnityEngine.Object, System.Single)
UnityEngine.Object.DestroyObject(UnityEngine.Object)
UnityEngine.Object.FindSceneObjectsOfType(System.Type)
UnityEngine.Object.FindObjectsOfTypeIncludingAssets(System.Type)
UnityEngine.Object.FindObjectsOfType<T>()
UnityEngine.Object.FindObjectOfType<T>()
UnityEngine.Object.FindObjectsOfTypeAll(System.Type)
UnityEngine.Object.FindObjectOfType(System.Type)
UnityEngine.Object.ToString()
UnityEngine.Object.name
UnityEngine.Object.hideFlags
System.Object.Equals(System.Object, System.Object)
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)

Namespace: LeTai.Asset.TranslucentImage.UniversalRP

Assembly: LeTai.TranslucentImage.UniversalRP.dll

Syntax

```
[MovedFrom("LeTai.Asset.TranslucentImage.LWRP")]
public class TranslucentImageBlurSource : ScriptableRendererFeature
```

Methods

AddRenderPasses(ScriptableRenderer, ref RenderingData)

Declaration

public override void AddRenderPasses(ScriptableRenderer renderer, ref RenderingData renderingData)

Parameters

Type	Name	Description
UnityEngine.Rendering.Universal.ScriptableRenderer	renderer	
UnityEngine.Rendering.Universal.RenderingData	renderingData	

Overrides

UnityEngine.Rendering.Universal.ScriptableRendererFeature.AddRenderPasses(UnityEngine.Rendering.Universal.ScriptableRenderer, UnityEngine.Rendering.Universal.RenderingData)

Create()

Declaration

public override void Create()

Overrides

UnityEngine.Rendering.Universal.ScriptableRendererFeature.Create()

RegisterSource(TranslucentImageSource)

When adding new Translucent Image Source to existing Camera at run time, the new Source must be registered here

Declaration

public void RegisterSource(TranslucentImageSource source)

Parameters

Type	Name	Description
TranslucentImageSource	source	

Class Utilities

Inheritance

System.Object
Utilities

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: LeTai.Asset.TranslucentImage.UniversalRP

Assembly: LeTai.TranslucentImage.UniversalRP.dll

Syntax

```
[MovedFrom("LeTai.Asset.TranslucentImage.LWRP")]
public static class Utilities
```

Methods

SimplePingPong(Int32, Int32)

Declaration

```
public static int SimplePingPong(int t, int max)
```

Parameters

Type	Name	Description
System.Int32	t	
System.Int32	max	

Returns

Type	Description
System.Int32	