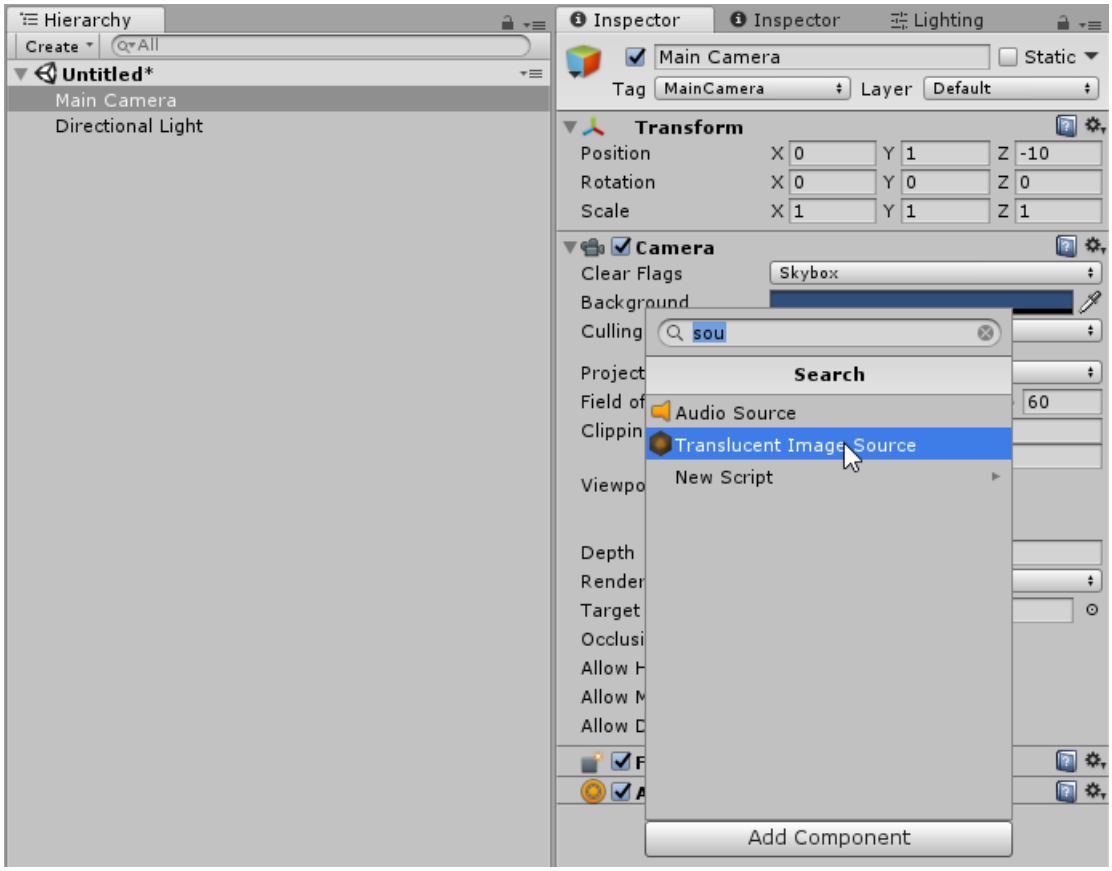
Table of Contents

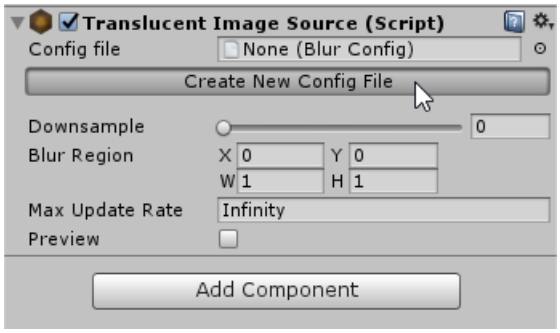
- Articles
  - Getting Started
  - Customize
  - Controls Translucent Image from script
  - Universal Render Pipeline
  - Blurring other UI elements
  - World space UI
  - FAQ
  - Support
- Api Documentation
  - LeTai.Asset.TranslucentImage
    - BlurAlgorithmType
    - BlurConfig
    - Extensions
    - IBlurAlgorithm
    - ScalableBlur
    - ScalableBlurConfig
    - ShaderId
    - TranslucentImage
    - TranslucentImageSource
  - LeTai.Asset.TranslucentImage.UniversalRP
    - BlurAlgorithmType
    - IBlurAlgorithm
    - ScalableBlur
    - ShaderId
    - TranslucentImageBlurRenderPass
    - TranslucentImageBlurSource
    - Utilities

Getting Started

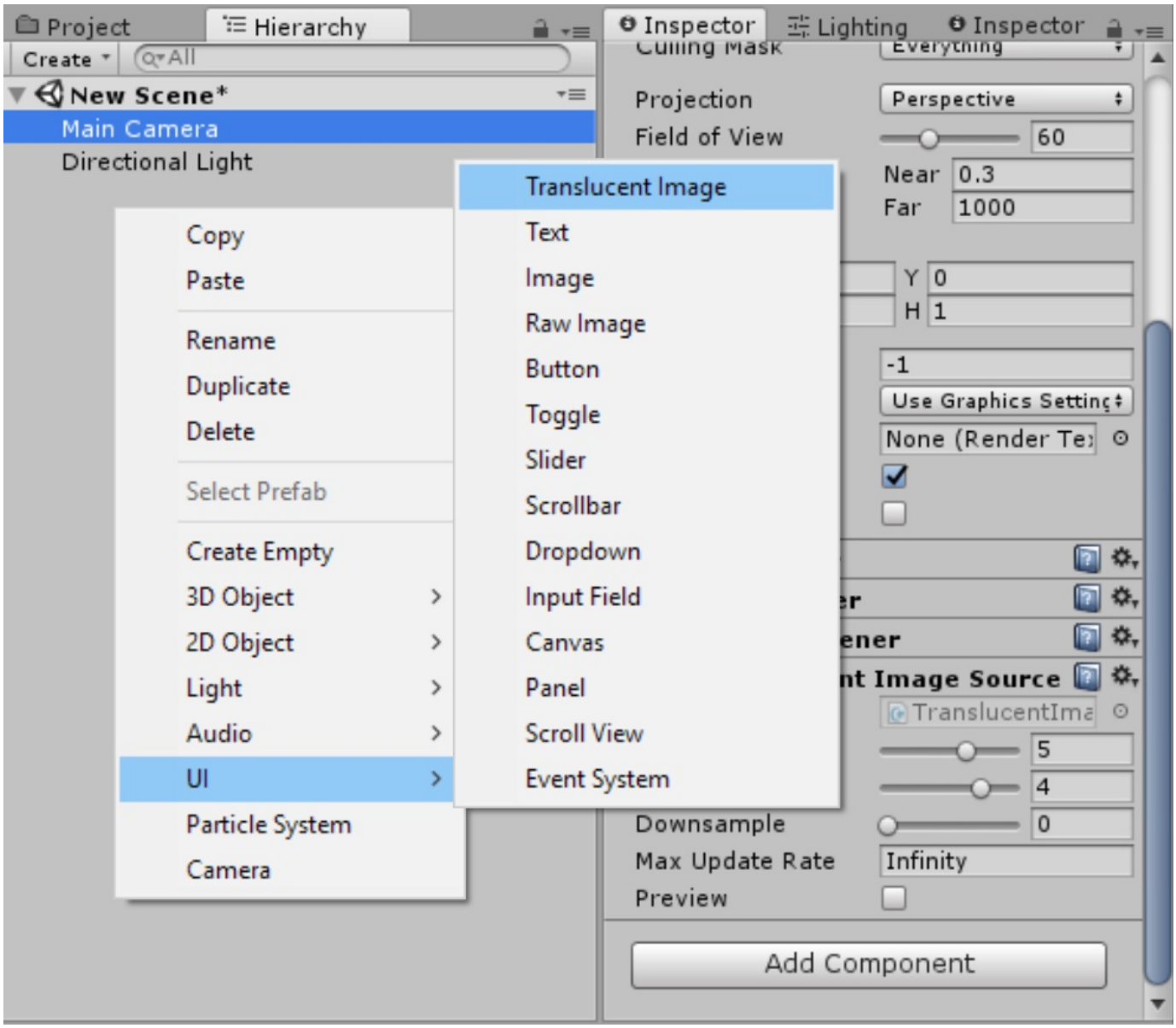
1. Add `Translucent Image Source` to your main camera.



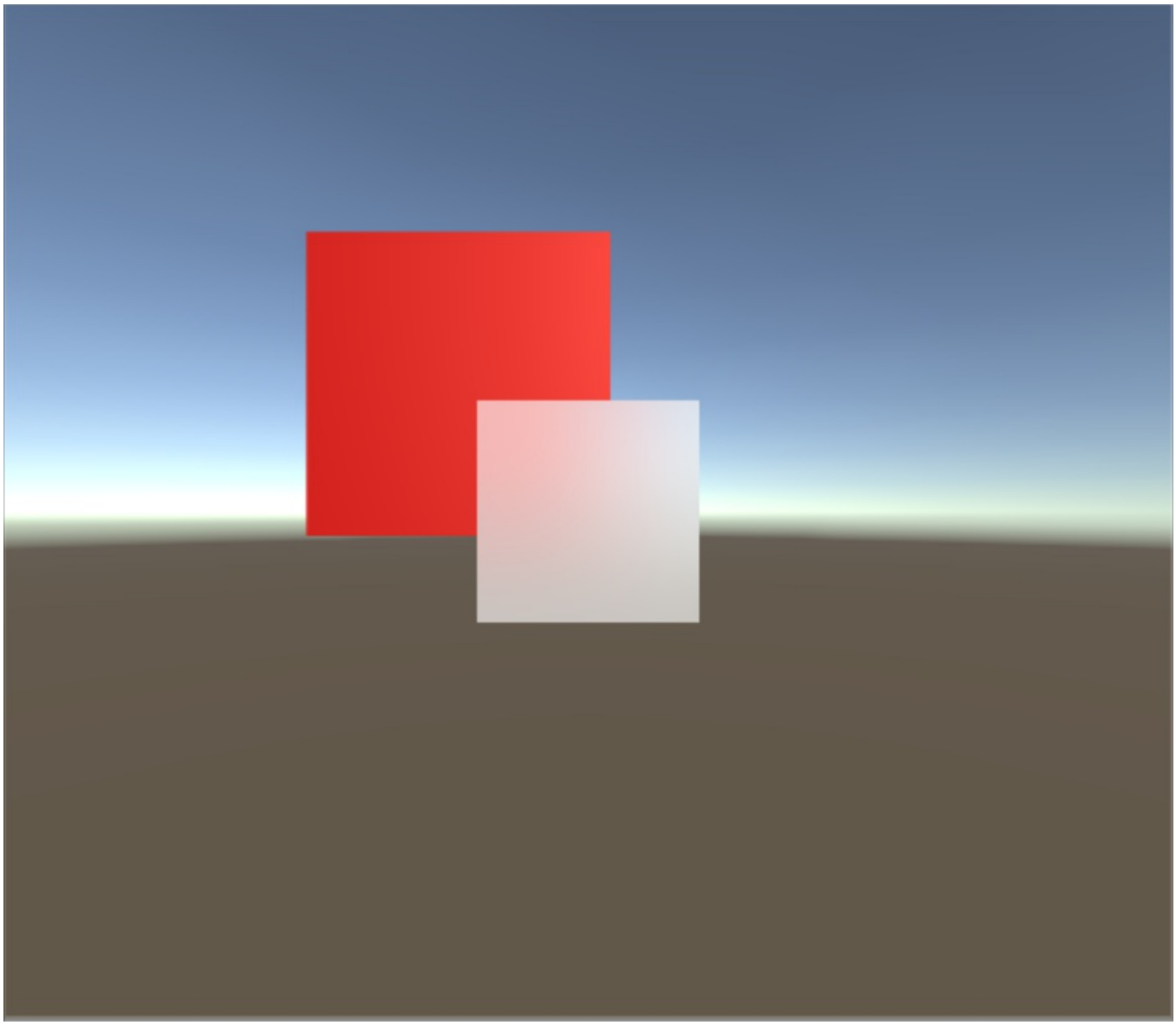
2. Create a ***Blur Config*** asset, or assign an existing one.



3. Create `UI -> Translucent Image` as you would with normal UI Image.


Project
Hierarchy
Inspector
Lighting
Inspector
Create
All
New Scene*
Main Camera
Directional Light
Culling Mask
Everything
Projection
Perspective
Field of View
60
Near
0.3
Far
1000
Y
0
H
1
-1
Use Graphics Setting
None (Render Te
Copy
Paste
Rename
Duplicate
Delete
Select Prefab
Create Empty
3D Object
2D Object
Light
Audio
UI
Particle System
Camera
Translucent Image
Text
Image
Raw Image
Button
Toggle
Slider
Scrollbar
Dropdown
Input Field
Canvas
Panel
Scroll View
Event System
ener
nt Image Source
TranslucentIma
5
4
Downsample
0
Max Update Rate
Infinity
Preview
Add Component

4. That's it!



⚠ WARNING

By default, Translucent Image will use a default Material. To make sure your Translucent Image look 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 willcontinue to show even when exit play mode.

Customize

> **NOTE**
>
> This asset was designed to be scalable. All properties that are said to affect performance below actually do so very little.

There are 2 components that form the effect, each with their own parameters that control how the effect look:

Translucent Image Source

This component offers two modes 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**: How much blurriness you want. Doesn't affect performance, but will look bad if the number too big. Also reduce flickering.
  - **Iteration**: Increase blur quality and blurriness when it is increased.
- **Max Depth**: Increase this property will:
  - Increase flickering when background moving
  - Increase blur level
  - Improve performance
- **Downsample**: Decrease the resolution before processing to increase performance. Side effect includes increased blurriness and flickering.
- **Blur Region**: Select the region of the screen to blur. If your UI does not span the entire screen, it might be a good idea to limit this to only the part that you use to increase performance and reduce power usage.

  > **TIP**
  >
  > It easier if you tune the `x` and `y` value before `w` and `h`

- **Max Update Rate**: How many times the effect updates itself each second. Use this property to increase performance and decrease power usage. Set to 0 to pause, this can reduce power usage/ prevent overheat when you don't need a dynamically updating background, for example, in a pause menu.
- **Preview**: preview the effect in full-screen without creating a Translucent Image

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, but they can batch dynamically to only take one 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**: Translucent Image Source component. This is where the image gets the blurred background. It will automatically be

set to the first one found, so you should make sure there one in your scene before creating any Translucent Image. You can always override this to change which camera will provide the blurred background.

- **Sprite Blending**: Mix between background and foreground image.

Shared settings

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

- **Vibrancy**: How colorful you want the background to be, 0 mean black and white, negative value will invert the color.
- **Brightness**: Brighten or darken the background.
- **Flatten**: Make your Translucent Image more contrast-y against the background. Useful when you can't predict the color of the background.

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

Controlling the blur settings is not very straightforward. The blur settings are stored as a ScriptableObject. The easiest way to access these settings is to create a public/serialized field in your script. You can then assign the setting asset you want to control to it in the inspector:

```
public ScalableBlurConfig blurSettings;
```

Alternatively, if you already 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 = 69;
}
```

Universal Render Pipeline

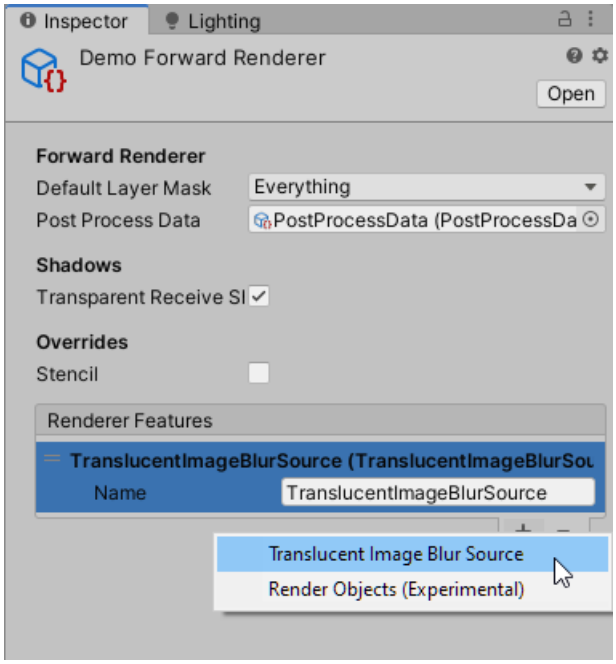
Requirements

The URP support package is tested on URP version 7.1.7 and Unity 2019.3.

The files required for URP support can be found by importing the unitypackage at: `TranslucentImage/UniversalRP support`. They are not included by default as they would produce errors for projects not using URP.

Tutorial

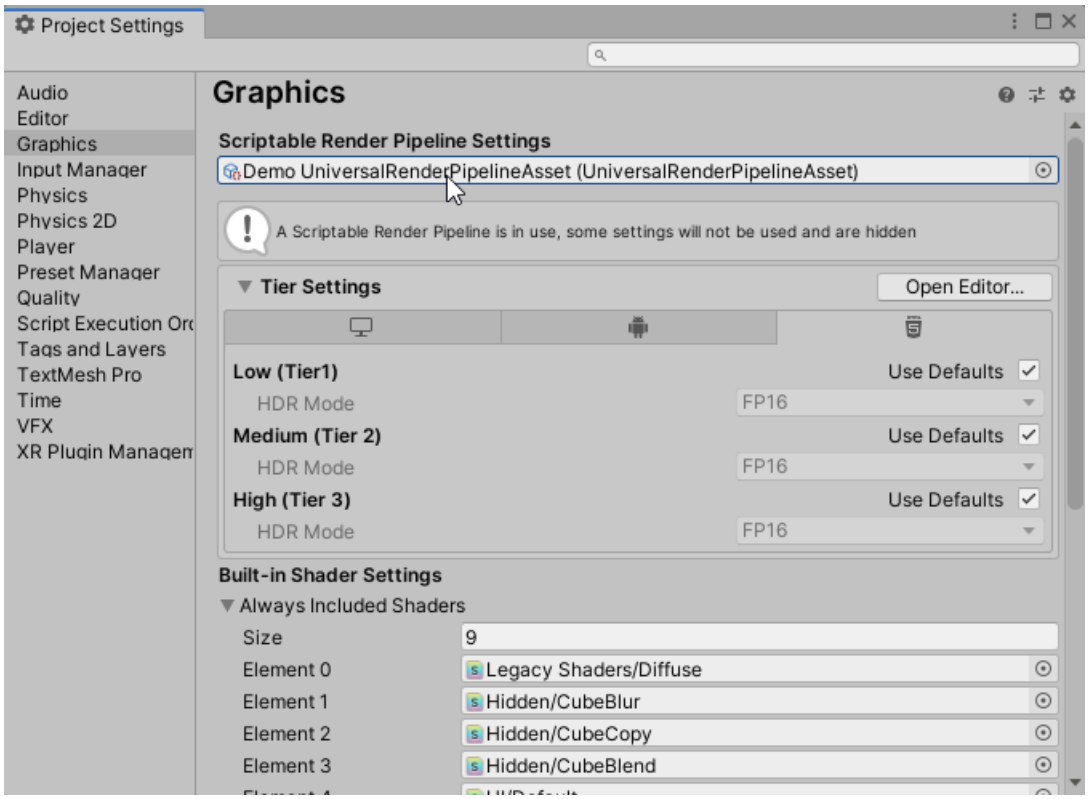
1. Import the package at `TranslucentImage/UniversalRP support`.
2. Find your **Forward Renderer Assets**, and add **TranslucentImageBlurSource** to the list of **Renderer Features**, you may have more than 1:



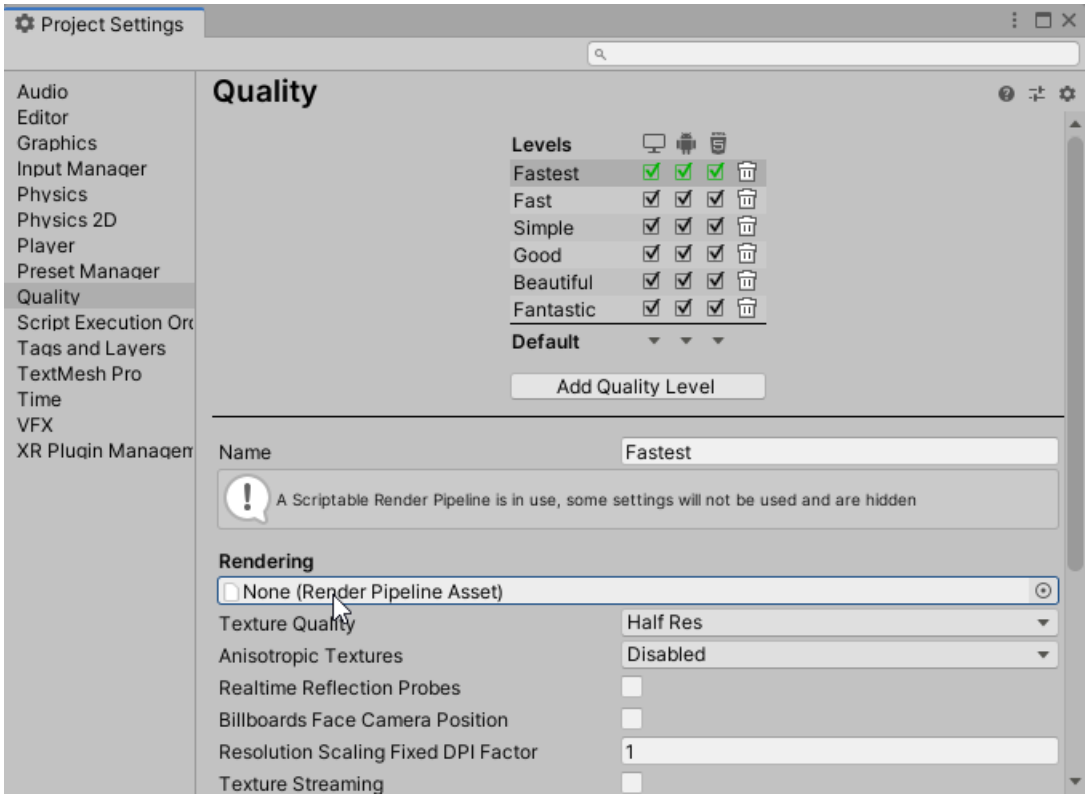
> **NOTE**
>
> You may have multiple **Forward Renderer Assets**. In which case you have to add the **Renderer Feature** to all of them.

Finding the Forward Renderer Assets

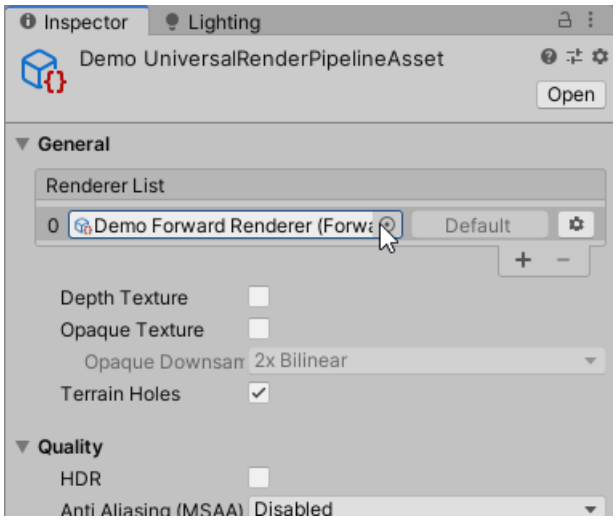
1. You can find the **Forward Renderer asset(s)** you're using by finding the **Render Pipeline Settings asset** in **Graphic Settings**:



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



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**:



Limitations

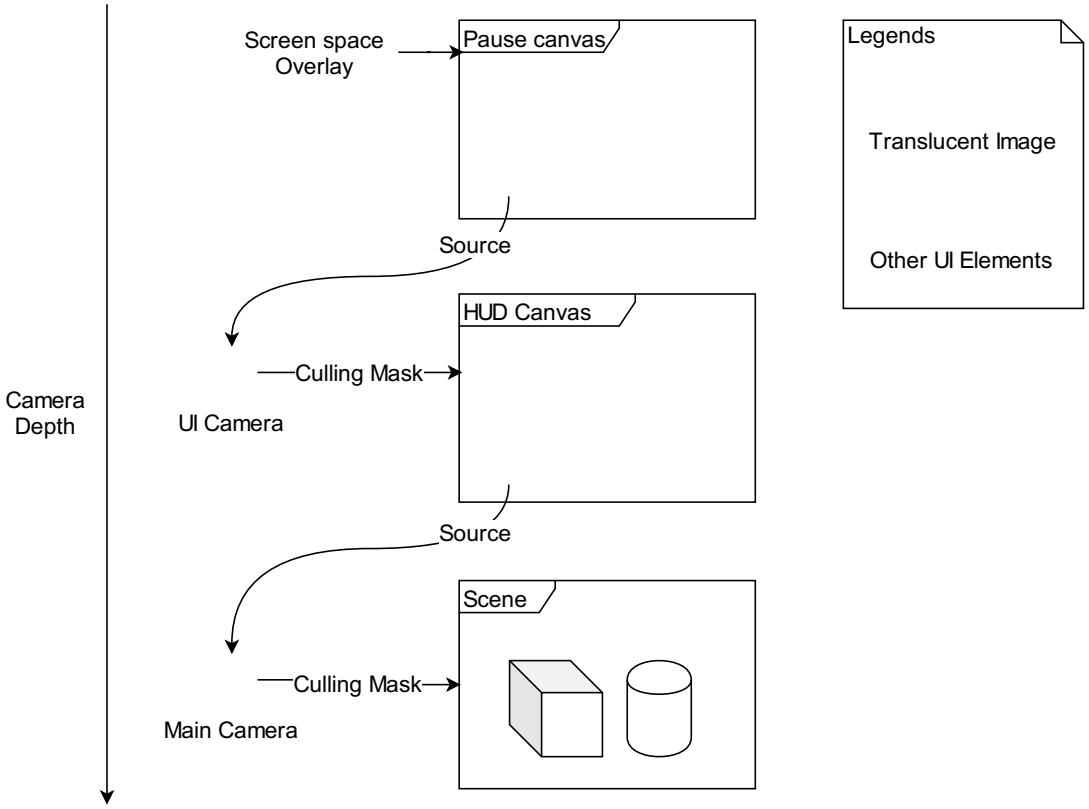
- URP does not yet support multiple cameras stacking on each other. Consequently, stacking multiple blur layers for Translucent Images is also not available. Unity does have this functionality planned.
- Some demo scene does not work with URP. There a demo scene dedicated for URP under `TranslucentImage/Demo/UniversalRP`, after you import the support package.

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 pause menu, where the pause menu UIs would be shown on top of the game HUD.

Translucent Image supports this on Standard Render Pipeline. LWRP is not supported for this use case, as the pipeline does not support multiple stacking cameras

The setup



Here is a typical setup:
Example setup

And here a more detailed description of the components you'll need and their settings:

- **Canvas**
  - *Overlay Canvas (i.e. pause menu)*
    - Layer: UI
    - Render Mode: Screen space - Overlay/Camera
    - Render Camera: Overlay Camera (optional)
  - *UI Canvas (i.e. HUD)*
    - Layer: UI
    - Render Mode: Screen space - Camera
    - Render Camera: UI Camera
- **Camera**
  - *Overlay Camera (optional)*
    - Clear Flags: Depth only
    - Culling Mask: UI
    - Depth: 3

- *UI Camera*
  - Clear Flags: Depth only
  - Culling Mask: UI
  - Depth: 2
- *Main Camera*
  - Clear Flags: <any>
  - Culling Mask: Everything but UI
  - Depth: 1

After you have the setup, add the Translucent Image Source component to *UI Camera* and *Main Camera*. Then:

- Assign the Translucent Images "Source" slot to the Source "below" them, like shown in the diagram
- Please note, Translucent Images that use different Sources should also use different Material.

Performance implication

You can add as many cameras and canvases as you like, and set them up similarly. 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 on top. In fact, both Windows 10 and macOS do this.

Windows 10 only use blur on the top-most UI

Why so complicated?

Many other blur solution you can find on the web can just blur whatever behind them. While this is convenient, it also very harmful for performance - each Image has to do their own blurring.

In Translucent Image, the blurring work is batched using Translucent Image Source(s). This massively reduces the amount of calculation the have to be done, especially when there's a lot of UIs.

The trade-off is when we want to blur other Translucent Images, we need to use extra "batches", which involve complex setup and more cost in performance.

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.

Note that this is not possible in LWRP/URP until Unity implements stacking Cameras there.

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 everything not too old.
- Android: There's 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: Apple A8 and later should hit 60FPS. A7 can hit 30FPS.

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

This usually due to one of the following:

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

Can I smoothly animate the blur level?

The strength property allows for mostly smooth change of blurriness, but not fully, there is still some abrupt jump that is noticeable when changing blurriness 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.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	

source

Source of blur for this image

Declaration

```
public TranslucentImageSource source
```

Field Value

TYPE	DESCRIPTION
TranslucentImageSource	

spriteBlending

Declaration

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

Field Value

TYPE	DESCRIPTION
System.Single	

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	

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

```
[ExecuteInEditMode]
[RequireComponent(typeof(Camera))]
[AddComponentMenu("Image Effects/Tai Le Assets/Translucent Image Source")]
[HelpURL("http://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

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	

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	

TYPE	NAME	DESCRIPTION
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[]	

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	