Table of Contents

Getting Started

1. Add `Translucent Image Source` component to your main camera.
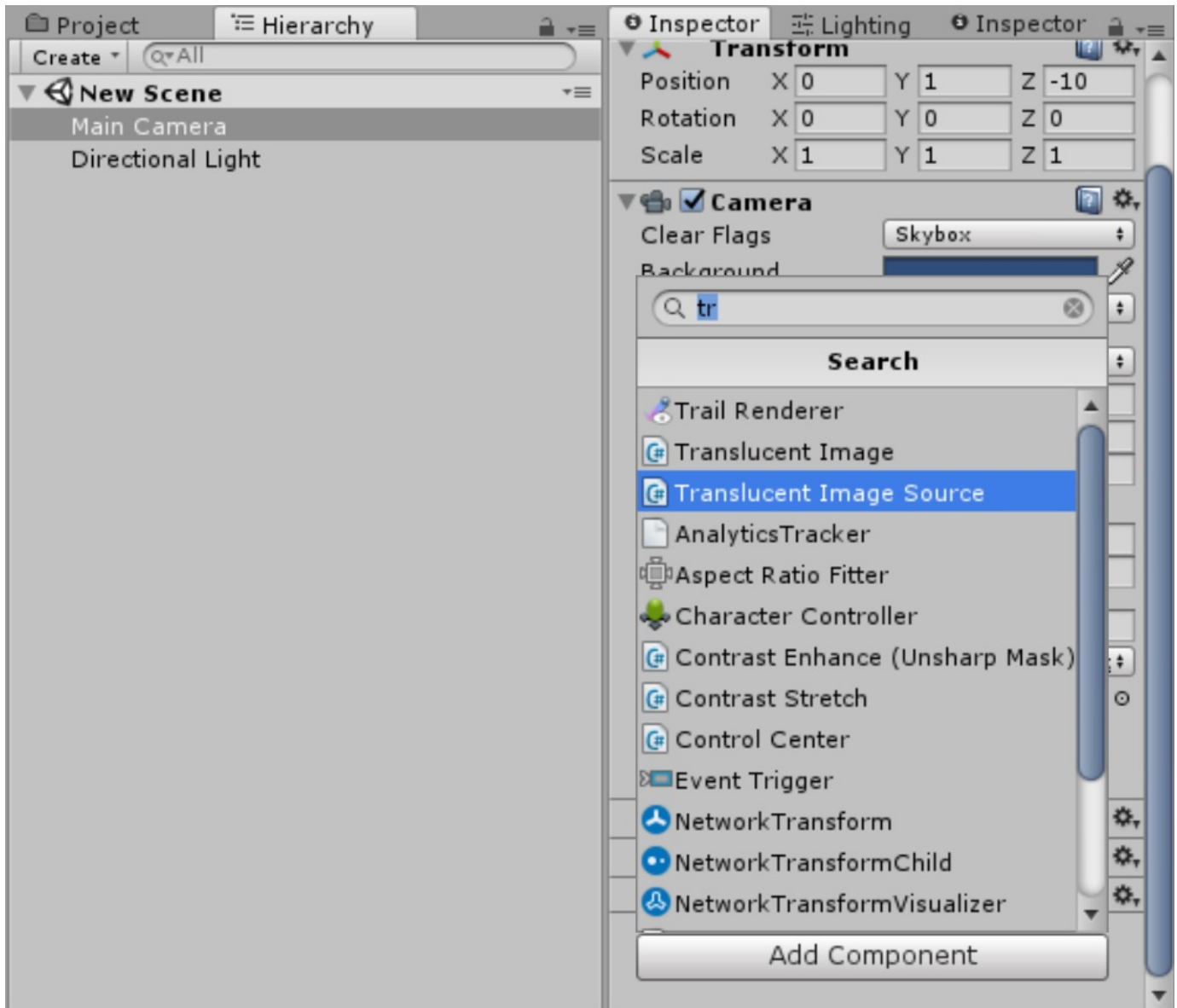
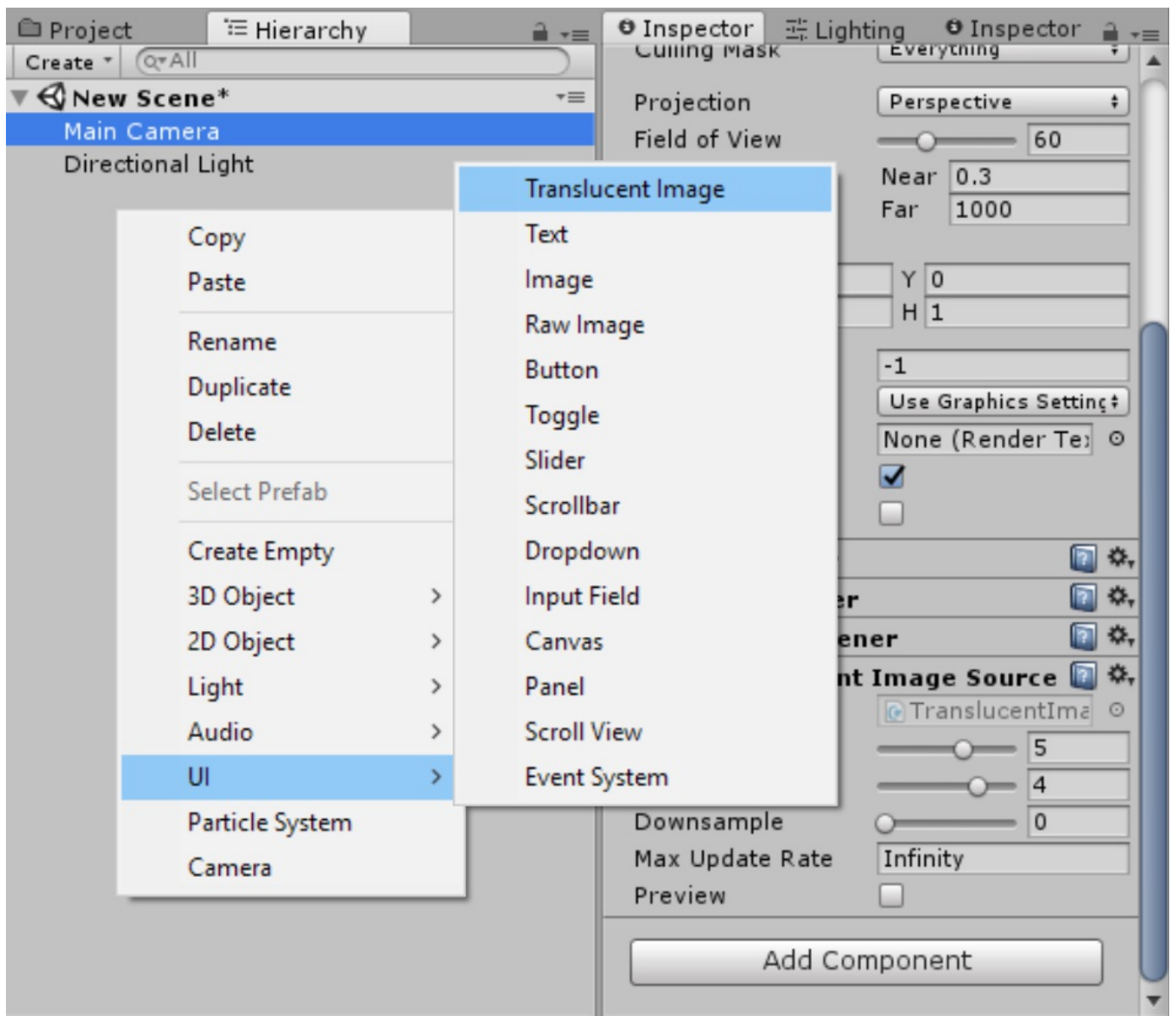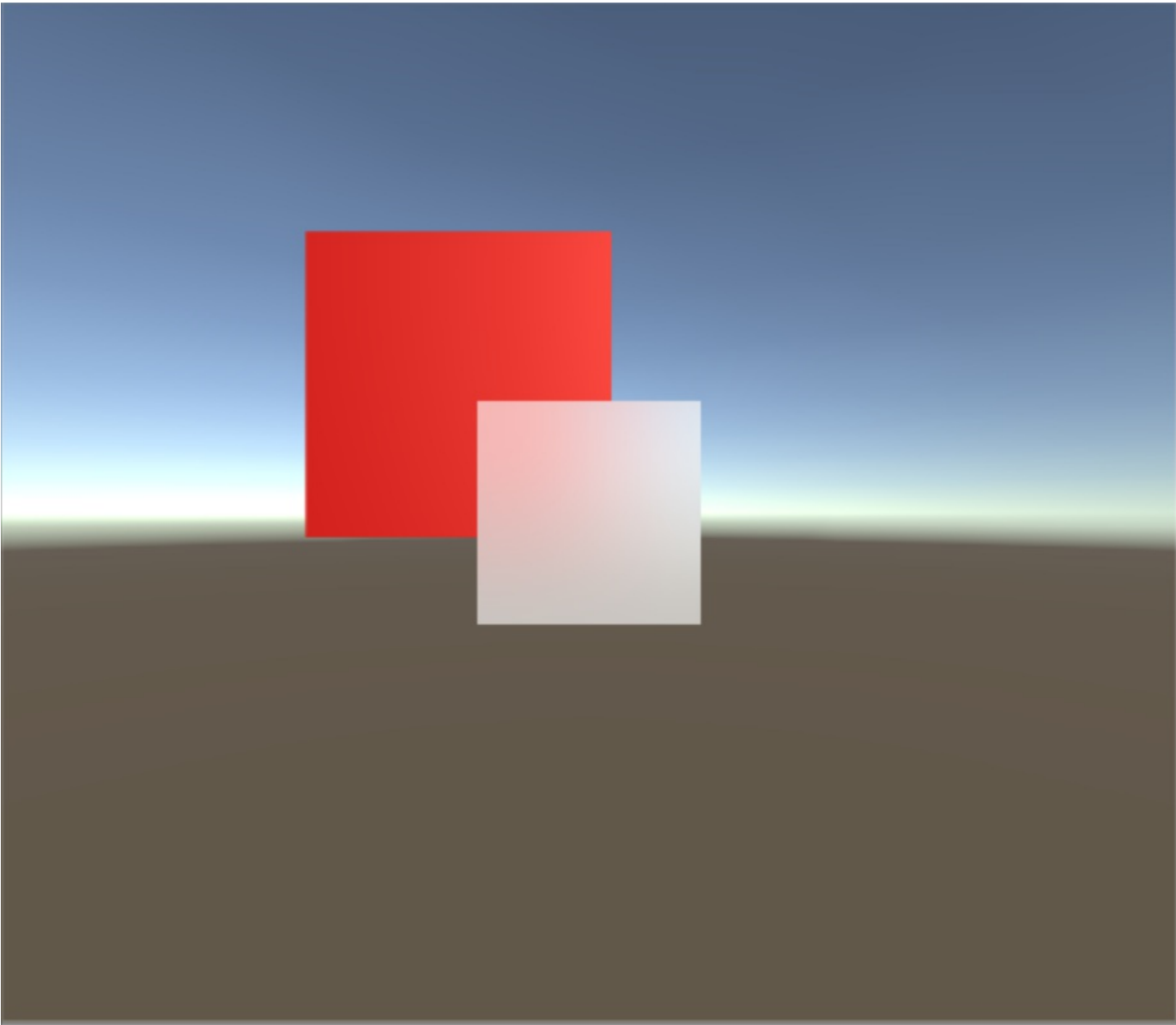2. Create `UI -> Translucent Image` as you would with built-in UI.

3. That this!

Customize

There are 2 components that form the effect, both with their own parameter that affect the look of the effect:

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.
    - **Downsample**: Decrease the resolution before processing to increase performance. Side effect include increase blurriness and flickering.

There are also other properties that are independant of mode:

- **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.

- **Max Depth**: Increase this property will:

    - Increase flickering when background moving
    - Increase blur level
    - Improve performance

- **Max Update Rate**: How many time the effect update itself per 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 dynamically updating background - like in a pause menu for example.

- **Preview**: preview the effect in full-screen without creating a Translucent Image

Translucent Image

- **Source Image**: The sprite to use for this image.

- **Material**: Multiple Translucent Image using the same material can only have different color, but they can batch dynamically to only take one draw call.

- **Color, Raycast Target, Image Type**: same as built-in Image.

- **Source**: Translucent Image Source component. This is where the image gets the blurred screen. It will automatically being 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 be blurred.

- **Vibrancy**: How colorful you want the background to be, 0 mean black and white, negative value will invert the color. This is great for enhancing the detail behind the image, or making death screen.

- **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.

Frequently Asked Questions

Will this asset works well on my device?

The asset should run on any device. Performance wise, it depend on your project, but here some general rule or thumb:

- PC/Mac/Console: Should run well on everything.
- Android: There's too many of them. Only way to know for sure is to test the demo on your target devices.
- IOS: Apple A8 and later should hit 60FPS. A7 can hit 30FPS.

Can I smoothly animate the blur level?

The strength property allow for some smoothness but not fully, no. If you just need to fade in and out, use the alpha value of the Color. You can also use Canvas Group as normal Images.

Can I blur other UI?

TL;DR: Kind of. See the demo.

Explanation

If the blur algorithm run once for each UI on the screen, it will get too slow very fast. In fact, if you use a Mac or Window 10 computer, you will notice the blur effect is disabled for windows that aren't in focus.

Therefore, the blurring is done once per camera. This give us much higher performance. However, every TranslucentImage share the same source will have the exact same background - which mean they will not "see" others below them.

Solution

If you need to blur UIs, for example, for a fully UI games, this can be done by setting up another *Camera* and *Canvas*.

The setup:

- *Main Camera*: Render the scene but NOT UIs
- *UI Camera*:
  - Render ONLY UIs.
  - Clear flag set to "Depth only".
  - Should have higher Depth than the *Main Camera*.
  - Add a TranslucentImageSource here.

- *Overlay Canvas*:
  - In Screen Space Overlay mode (Render on top of everything).
  - This is where we put our TranslucentImage (TI for short).
  - These TIs have source set to *UI Camera*.

- *Camera Canvas*:
  - In Screen Space Camera mode (Render Camera set to the *UI Camera*).
  - This is where we put all our normal UIs.

You can also have TIs in the *Camera Canvas*. Just add another TranslucentImageSource to the *Main Camera*, and set all the TIs in the *Camera Canvas* to that Source.

TL;DR: See the demo scene.

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

TranslucentImage

Dynamic blur-behind UI element

TranslucentImageSource

Common source of blur for Translucent Images.

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.OnBeforeSerialize()

UnityEngine.UI.Image.OnAfterDeserialize()

UnityEngine.UI.Image.SetNativeSize()

UnityEngine.UI.Image.OnPopulateMesh(UnityEngine.UI.VertexHelper)

UnityEngine.UI.Image.UpdateMaterial()

UnityEngine.UI.Image.CalculateLayoutInputHorizontal()

UnityEngine.UI.Image.CalculateLayoutInputVertical()

UnityEngine.UI.Image.IsRaycastLocationValid(UnityEngine.Vector2, UnityEngine.Camera)

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.defaultETC1GraphicMaterial

UnityEngine.UI.Image.mainTexture

UnityEngine.UI.Image.hasBorder

UnityEngine.UI.Image.pixelsPerUnit

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_IncludeForMasking

UnityEngine.UI.MaskableGraphic.m_ShouldRecalculate

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.OnValidate()

UnityEngine.UI.MaskableGraphic.OnTransformParentChanged()

UnityEngine.UI.MaskableGraphic.ParentMaskStateChanged()

UnityEngine.UI.MaskableGraphic.OnCanvasHierarchyChanged()

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.Graphic.s_DefaultUI

UnityEngine.UI.Graphic.s_WhiteTexture

UnityEngine.UI.Graphic.m_Material

UnityEngine.UI.Graphic.m_OnDirtyLayoutCallback

UnityEngine.UI.Graphic.m_OnDirtyVertsCallback

UnityEngine.UI.Graphic.m_OnDirtyMaterialCallback

UnityEngine.UI.Graphic.s_Mesh

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.Rebuild(UnityEngine.UI.CanvasUpdate)

UnityEngine.UI.Graphic.LayoutComplete()

UnityEngine.UI.Graphic.GraphicUpdateComplete()

UnityEngine.UI.Graphic.UpdateGeometry()

UnityEngine.UI.Graphic.OnFillVBO(System.Collections.Generic.List<UnityEngine.UIVertex>)

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.OnDestroy()

UnityEngine.EventSystems.UIBehaviour.IsActive()

UnityEngine.EventSystems.UIBehaviour.OnCanvasGroupChanged()

UnityEngine.EventSystems.UIBehaviour.IsDestroyed()

UnityEngine.MonoBehaviour.Invoke(System.String, System.Single)

UnityEngine.MonoBehaviour.InvokeRepeating(System.String, System.Single, System.Single)

UnityEngine.MonoBehaviour.CancelInvoke()

UnityEngine.MonoBehaviour.CancelInvoke(System.String)

UnityEngine.MonoBehaviour.IsInvoking(System.String)

UnityEngine.MonoBehaviour.IsInvoking()

UnityEngine.MonoBehaviour.StartCoroutine(System.Collections.IEnumerator)

UnityEngine.MonoBehaviour.StartCoroutine_Auto(System.Collections.IEnumerator)

UnityEngine.MonoBehaviour.StartCoroutine(System.String, System.Object)

UnityEngine.MonoBehaviour.StartCoroutine(System.String)

UnityEngine.MonoBehaviour.StopCoroutine(System.String)

UnityEngine.MonoBehaviour.StopCoroutine(System.Collections.IEnumerator)

UnityEngine.MonoBehaviour.StopCoroutine(UnityEngine.Coroutine)

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.GetComponent(System.String)

UnityEngine.Component.GetComponentInChildren(System.Type, System.Boolean)

UnityEngine.Component.GetComponentInChildren(System.Type)

UnityEngine.Component.GetComponentInChildren<T>()

UnityEngine.Component.GetComponentInChildren<T>(System.Boolean)

UnityEngine.Component.GetComponentsInChildren(System.Type)

UnityEngine.Component.GetComponentsInChildren(System.Type, System.Boolean)

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)

UnityEngine.Component.GetComponentsInParent(System.Type, System.Boolean)

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.SendMessageOptions)

UnityEngine.Component.SendMessage(System.String, System.Object)

UnityEngine.Component.SendMessage(System.String)

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.Component.rigidbody

UnityEngine.Component.rigidbody2D

UnityEngine.Component.camera

UnityEngine.Component.light

UnityEngine.Component.animation

UnityEngine.Component.constantForce

UnityEngine.Component.renderer

UnityEngine.Component.audio

UnityEngine.Component.guiText

UnityEngine.Component.networkView

UnityEngine.Component.guiElement

UnityEngine.Component.guiTexture

UnityEngine.Component.collider

UnityEngine.Component.collider2D

UnityEngine.Component.hingeJoint

UnityEngine.Component.particleEmitter

UnityEngine.Component.particleSystem

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.FindObjectsOfTypeAll(System.Type)

UnityEngine.Object.ToString()

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.FindObjectsOfType<T>()

UnityEngine.Object.FindObjectOfType<T>()

UnityEngine.Object.FindObjectOfType(System.Type)

UnityEngine.Object.name

UnityEngine.Object.hideFlags

System.Object.Equals(System.Object, System.Object)

System.Object.ReferenceEquals(System.Object, System.Object)

System.Object.GetType()

System.Object.MemberwiseClone()

Namespace: LeTai.Asset.TranslucentImage

Assembly: Assembly-CSharp.dll

Syntax

```
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	

correctShader

Declaration

```
Shader correctShader
```

Field Value

TYPE	DESCRIPTION
UnityEngine.Shader	

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	

oldBrightness

Declaration

```
float oldBrightness
```

Field Value

TYPE	DESCRIPTION
System.Single	

oldFlatten

Declaration

```
float oldFlatten
```

Field Value

TYPE	DESCRIPTION
System.Single	

oldVibrancy

Declaration

```
float oldVibrancy
```

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

LateUpdate()

Declaration

```
void LateUpdate()
```

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.Graphic.OnDidApplyAnimationProperties()

OnDisable()

Declaration

```
protected override void OnDisable()
```

Overrides

UnityEngine.UI.MaskableGraphic.OnDisable()

OnEnable()

Declaration

```
protected override void OnEnable()
```

Overrides

UnityEngine.UI.MaskableGraphic.OnEnable()

PrepShader()

Declaration

```
void PrepShader()
```

Start()

Declaration

```
protected override void Start()
```

Overrides

UnityEngine.EventSystems.UIBehaviour.Start()

SyncMaterialProperty(Int32, ref Single, ref Single)

Sync material property with instance

Declaration

```
void SyncMaterialProperty(int propId, ref float value, ref float oldValue)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	propId	material property id
System.Single	value	
System.Single	oldValue	

Update()

Declaration

```
void Update()
```

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.Invoke(System.String, System.Single)

UnityEngine.MonoBehaviour.InvokeRepeating(System.String, System.Single, System.Single)

UnityEngine.MonoBehaviour.CancelInvoke()

UnityEngine.MonoBehaviour.CancelInvoke(System.String)

UnityEngine.MonoBehaviour.IsInvoking(System.String)

UnityEngine.MonoBehaviour.IsInvoking()

UnityEngine.MonoBehaviour.StartCoroutine(System.Collections.IEnumerator)

UnityEngine.MonoBehaviour.StartCoroutine_Auto(System.Collections.IEnumerator)

UnityEngine.MonoBehaviour.StartCoroutine(System.String, System.Object)

UnityEngine.MonoBehaviour.StartCoroutine(System.String)

UnityEngine.MonoBehaviour.StopCoroutine(System.String)

UnityEngine.MonoBehaviour.StopCoroutine(System.Collections.IEnumerator)

UnityEngine.MonoBehaviour.StopCoroutine(UnityEngine.Coroutine)

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.GetComponent(System.String)

UnityEngine.Component.GetComponentInChildren(System.Type, System.Boolean)

UnityEngine.Component.GetComponentInChildren(System.Type)

UnityEngine.Component.GetComponentInChildren<T>()

UnityEngine.Component.GetComponentInChildren<T>(System.Boolean)

UnityEngine.Component.GetComponentsInChildren(System.Type)

UnityEngine.Component.GetComponentsInChildren(System.Type, System.Boolean)

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)

UnityEngine.Component.GetComponentsInParent(System.Type, System.Boolean)

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.SendMessageOptions)

UnityEngine.Component.SendMessage(System.String, System.Object)

UnityEngine.Component.SendMessage(System.String)

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.Component.rigidbody

UnityEngine.Component.rigidbody2D

UnityEngine.Component.light

UnityEngine.Component.animation

UnityEngine.Component.constantForce

UnityEngine.Component.renderer

UnityEngine.Component.audio

UnityEngine.Component.guiText

UnityEngine.Component.networkView

UnityEngine.Component.guiElement

UnityEngine.Component.guiTexture

UnityEngine.Component.collider

UnityEngine.Component.collider2D

UnityEngine.Component.hingeJoint

UnityEngine.Component.particleEmitter

UnityEngine.Component.particleSystem

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.FindObjectsOfTypeAll(System.Type)

UnityEngine.Object.ToString()

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.FindObjectsOfType<T>()

UnityEngine.Object.FindObjectOfType<T>()

UnityEngine.Object.FindObjectOfType(System.Type)

UnityEngine.Object.name

UnityEngine.Object.hideFlags

System.Object.Equals(System.Object, System.Object)

System.Object.ReferenceEquals(System.Object, System.Object)

System.Object.GetType()

System.Object.MemberwiseClone()

Namespace: LeTai.Asset.TranslucentImage

Assembly: Assembly-CSharp.dll

Syntax

```
[ExecuteInEditMode]
[RequireComponent(typeof(Camera))]
[AddComponentMenu("Image Effects/Tai Le Assets/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

blurRegion

Declaration

```
[SerializeField]
Rect blurRegion
```

Field Value

TYPE	DESCRIPTION
UnityEngine.Rect	

camera

Declaration

```
Camera camera
```

Field Value

TYPE	DESCRIPTION
UnityEngine.Camera	

downsample

Declaration

```
[SerializeField]
int downsample
```

Field Value

TYPE	DESCRIPTION
System.Int32	

iteration

Declaration

```
[SerializeField]
int iteration
```

Field Value

TYPE	DESCRIPTION
System.Int32	

lastBlurRegion

Declaration

```
Rect lastBlurRegion
```

Field Value

TYPE	DESCRIPTION
UnityEngine.Rect	

lastDownsample

Declaration

```
[SerializeField]
int lastDownsample
```

Field Value

TYPE	DESCRIPTION
System.Int32	

lastUpdate

Declaration

```
float lastUpdate
```

Field Value

TYPE	DESCRIPTION
System.Single	

TYPE	DESCRIPTION

material

Declaration

```
Material material
```

Field Value

TYPE	DESCRIPTION
UnityEngine.Material	

maxDepth

Declaration

```
[SerializeField]
int maxDepth
```

Field Value

TYPE	DESCRIPTION
System.Int32	

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

```
[Tooltip("Preview the effect on entire screen")]
public bool preview
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

previewMaterial

Declaration

Material previewMaterial

Field Value

TYPE	DESCRIPTION
UnityEngine.Material	

shader

Declaration

Shader shader

Field Value

TYPE	DESCRIPTION
UnityEngine.Shader	

size

Declaration

[SerializeField] float size

Field Value

TYPE	DESCRIPTION
System.Single	

strength

Declaration

[SerializeField] float strength

Field Value

TYPE	DESCRIPTION
System.Single	

Properties

BlurredScreen

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

Declaration

public RenderTexture BlurredScreen { get; }

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

Cam

The Camera attached to the same GameObject. Cached in field 'camera'

Declaration

```
Camera Cam
{
}
```

Property Value

TYPE	DESCRIPTION
UnityEngine.Camera	

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

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

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

MinUpdateCycle

Minimum time in second to wait before refresh the blurred image. If maxUpdateRate non-positive then just stop updating

Declaration

```
float MinUpdateCycle
{
}
```

Property Value

TYPE	DESCRIPTION
System.Single	

ScreenSize

A small number base on the smaller dimension of the camera render target. Used to retain the blur amount across screen size

Declaration

```
float ScreenSize
{
}
```

Property Value

TYPE	DESCRIPTION
System.Single	

Size

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

Declaration

```
public float Size { 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

CreateNewBlurredScreen()

Declaration

```
protected virtual void CreateNewBlurredScreen()
```

OnRenderImage(RenderTexture, RenderTexture)

Declaration

```
protected virtual void OnRenderImage(RenderTexture source, RenderTexture destination)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.RenderTexture	source	
UnityEngine.RenderTexture	destination	

ProgressiveBlur(RenderTexture)

Declaration

```
protected virtual void ProgressiveBlur(RenderTexture sourceRt)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.RenderTexture	sourceRt	

ProgressiveResampling(Int32, ref RenderTexture)

Resize the source texture then run it through a shader before assign to target texure

Declaration

```
protected virtual void ProgressiveResampling(int level, ref RenderTexture target)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	level	Resampling depth
UnityEngine.RenderTexture	target	

SetAdvancedFieldFromSimple()

Calculate size and iteration from strength

Declaration

```
protected virtual void SetAdvancedFieldFromSimple()
```

Start()

Declaration

```
protected virtual void Start()
```