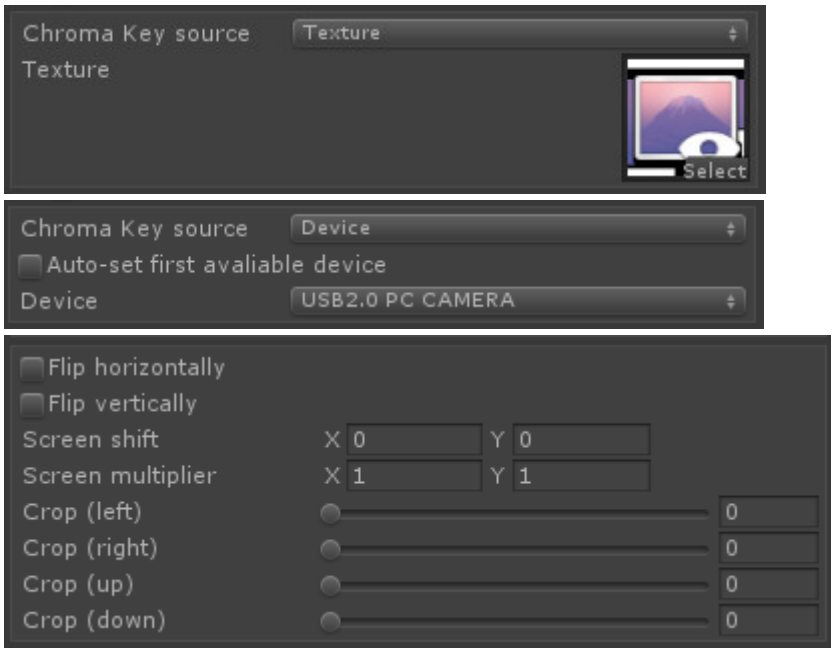
UChromaKey.

There's 3 core functions in UChromaKey asset:

- Fullscreen ChromaKey image effect
- Ability to apply ChromaKey to materials (ChromaKey shaders and source handling)
- Using ChromaKey to generate alpha-channel of your textures during runtime

If you want to use fullscreen effect or both fullscreen effect and materials, add UChromaKey component (Image effects -> U Chroma Key) to your camera. If you want to use chroma key effects only on your materials, use shaders from "UChromaKey/Direct source" branch.

Source handling:



UChromaKey component handles source for chroma-keyed images for both fullscreen effect and materials.

Chroma key source : what to use as a source of keyed image. Supports textures(including RenderTexture) and camera devices.

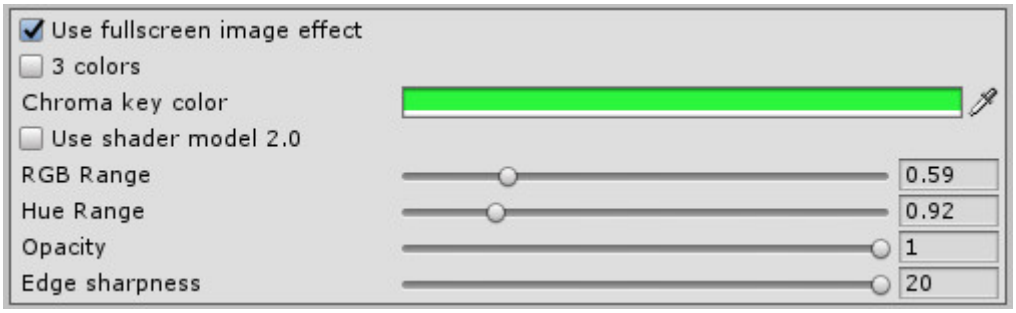
Auto-set first available device: if you're deploying on variety of platforms and devices, webcam naming will be different for each device. Use this toggle to tell UChromaKey to simply select first available device at the startup. Otherwise you'll need to handle it manually by setting DeviceName property of component. To get the list of available devices, get static WebCamTexture.devices property.

Although UChromaKey automatically manages web camera, you can access WebCamTexture via "webCamTexture" variable of UChromaKey component.

Flip horizontally and vertically toggles allow you to flip source picture. Some webcams provide inverted source, use this to fix that.

Screen shift and multiplier allow you to move and resize chroma-key source frame on your screen.

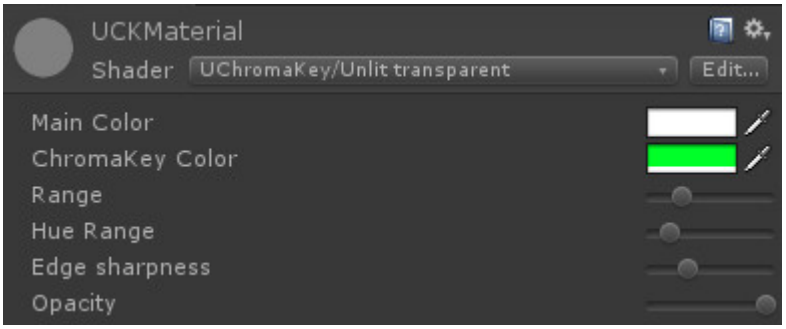
Fullscreen image effect.



To use fullscreen effect, enable “Use fullscreen image effect” checkbox. Note that UChromaKey component needs to be on GameObject with Camera component for it to work.

Chroma key color : Color that will be transparent on the result.
3 colors : When toggled, you can designate 3 chroma-keying colors instead of 1.
Use shader model 2.0 : When toggled, UCK will use simplified shaders – use this if you plan deploying on older mobile devices.
RGB Range : Range in RGB space that defines how big the color interval will be around selected color.
Hue Range : Range in hue that defines how big the color interval will be around selected color. Not available if you’re using shader model 2.0.
Opacity: defines opacity of keyed image.
Edge sharpness: How sharp the edges of keyed areas is.

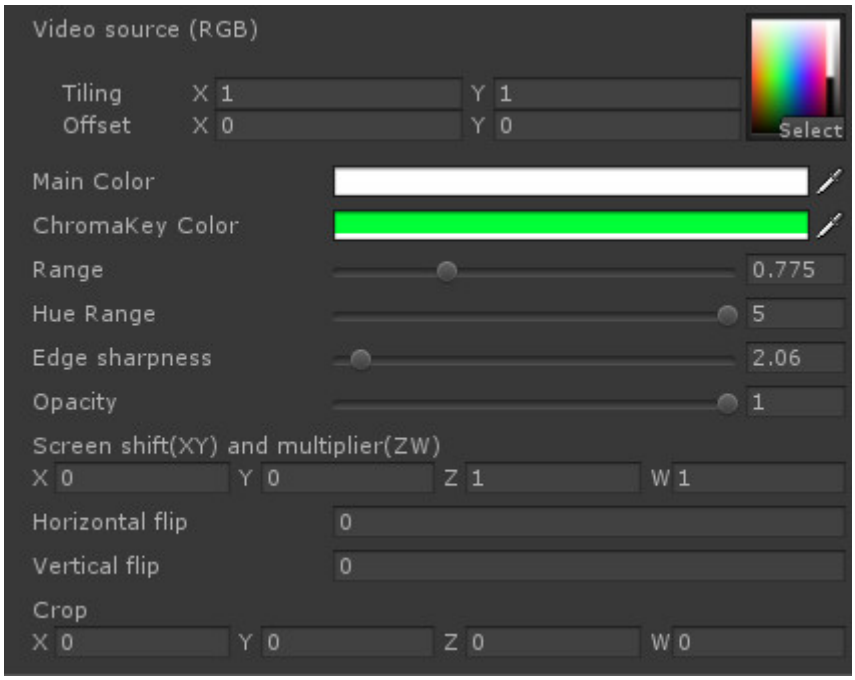
ChromaKey for material.



To use UChromaKey for Unity materials along with fullscreen effect, select shader from “UChromaKey” shader branch. Note that there’s no slot for main texture – UchromaKey component handles image input. Shader parameters are identical to fullscreen effect parameters.

Note that you can create multiple materials with shared source but different chroma-keying parameters.

Direct source shaders.



You can use direct source shaders if you don't want UChromaKey main component to be responsible for source texture of your material. These shaders are completely decoupled from other UCK components and can be used with any type of texture. Direct source shaders are also useful when integrating UCK with other plugins such as Vuforia etc.

All parameters are identical with fullscreen effect.

UChromaKey shader parameters

You can change your material parameters during runtime by using **Material.SetFloat**, **Material.SetColor** and **Material.SetTexture.** For more info see Unity Script Reference.

Parameter	Internal name for scripting purposes	Range	Default value
ChromaKey Color	_CKCol	Color	white
Range	_Range	0.0 – 2. 83	0.01
Hue Range	_HueRange	0.0 – 5.0	0.1
Edge sharpness	_EdgeSharp	1.0 – 20.0	20
Opacity	_Opacity	0.0 – 1.0	1
Source texture for non-direct-source	_UChromaKeyTex	Texture	-
Source texture for direct-source	_MainTex	Texture	-

Using UChromaKey to apply alpha-channel to your textures.

UChromaKey class contains static ChromaKeyAlphaBlit method that allows you to chroma-key your texture, with resulting transparency written into alpha-channel of RenderTexture, while color information remains unchanged.

Example of usage:

```
UChromaKey.ChromaKeyAlphaBlit(sourceTexture2D, outputRenderTexture,
Color.Green, 0.5f, 0.2f)
```

There is 2 overloaded methods, with second one providing more control over result:

```
public static void ChromaKeyAlphaBlit(Texture2D input, RenderTexture outp
ut, Color color, float range, float hueRange, float opacity = 1.0f, float
edgeSharpness = 20.0f)
```

```
public static void ChromaKeyAlphaBlit(Texture2D input, RenderTexture outp
ut, Color color, float range, float hueRange, float opacity , float
edgeSharpness, Vector2 shift, Vector2 multiplier,  bool flipH, bool flipV
, Vector4 crop)
```

Support:

Send your questions, suggestions and bug reports to: LeonidMV@gmail.com