

## **Advantages:**

- # 60% faster than Unity's built-in bloom, and specially optimized for mobile platforms.
- # This bloom can be applied to only intended specific objects. And its intensity and threshold can be controlled per-object.
- # This bloom can be applied to objects with random moving shapes (like glittering water surface under the sun).
- # Color grading is well integrated and costs little extra performance.

## **Limitations:**

- # Color of bloom is the same for all objects.
- # It cannot be used together with anti-aliasing.
- # It requires OpenGL ES 3.0
- # There's still potential performance cost. Be sure to profile your mobile device to check the increased performance time.

How to use: if you use HDR, read the following section. If you don't use HDR, jump to "No HDR" section. On top of that, if you're using translucent (semi-transparent) shader, check "Translucent" section. **If you're artist** and everything has been perfectly installed into your project, go to "How to tweak? For artists" section.

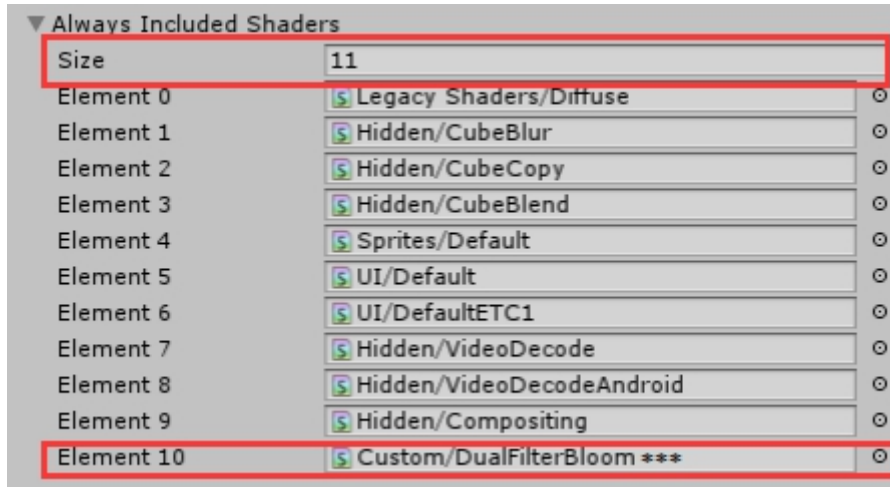
## **With HDR:**

- # Navigate through: File -> Build Settings -> switch platform to Android.
- # Navigate through: Edit -> Project Settings -> Quality, and ensure the anti-aliasing is disabled.

# Copy the following files into your project:

- ../Assets/Shaders/BloomHDR.shader
- ../Assets/Shaders/DualFilterBloomHDR.shader
- ../Assets/Scripts/ColorGradingBloomHDR.cs

# Navigate through: Edit -> Project Settings -> Graphics -> scroll down to Built-in Shader Settings / Always Included Shaders, and increase "Size" by 1, then click the circle on the right and find "DualFilterBloomHDR":



# For those objects on which you want to apply these effects, modify their shaders: Create a new shader, and copy the contents of your former shader into it. If you're using unity's built-in shaders, you can go to <https://unity3d.com/get-unity/download/archive> to download them from the drop-down menu:



Note that we can only modify vertex-fragment shaders, and we can't modify surface shaders using this method (because of using MRT).

Then modify the newly created shader according to BloomHDR.shader. Find the fragment function, and change its return type to mrtOutput, then add a definition of struct mrtOutput. After that, add the contents from "//MRT" to the end of the fragment function. **Here "hdrCol" is your former return value**, replace it. Finally add properties shown in BloomHDR.shader except \_MainTex.

# Create a new material for this shader, and apply it on those intended objects. You can now set the "Threshold Luminance" on the material and this value will apply on all intended objects universally, or you can separately set the "Threshold Luminance" on different objects. If you want to separately control the strength of bloom on each object, you can set the "Object Bloom Strength" or even write a script to make it breathe (as BloomBreath.cs). Note that this setting is restricted in range between 0 to 1, which means you can only decrease the strength here. The way to intensify the bloom strength will be introduced in the next step.

# Add ColorGradingBloomHDR.cs as a component to your camera. Increase the size of "Hdr Object Material" by how many new materials you created in last step, and drag the materials to the elements below. After that, set "Upmost Luminance G Channel", "Exposure" and "Bloom Color". And you can set the "Global Bloom Strength" above 1 to intensify bloom of ALL objects.

# To use color grading, download "Cinema Theme" from asset store. Find the script component we added on the camera and drag a LUT (a texture) into "Color Grading LUT". After that, just tick the "Enable Color Grading" box.

---

### **No HDR:**

# Navigate through: File -> Build Settings -> switch platform to Android.

# Navigate through: Edit -> Project Settings -> Quality, and ensure that anti-aliasing is disabled.

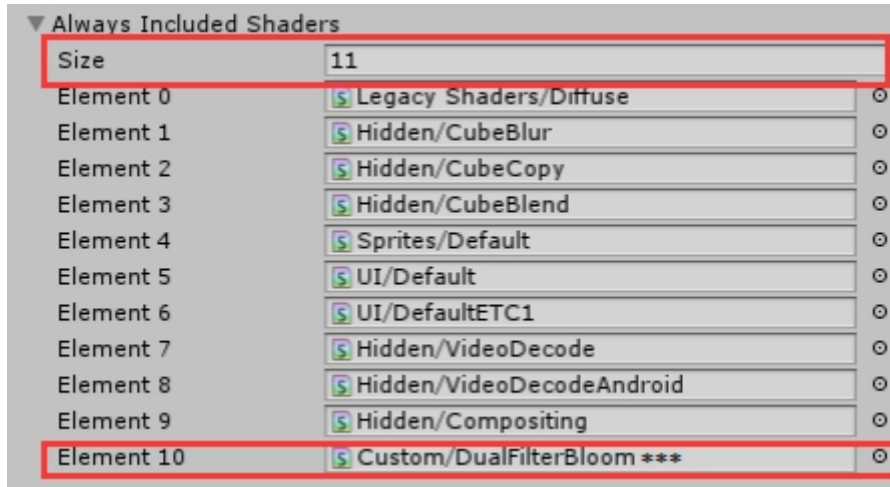
# Copy the following files into your project:

../Assets/Shaders/BloomNoHDR.shader

../Assets/Shaders/DualFilterBloomNoHDR.shader

../Assets/Scripts/ColorGradingBloomNoHDR.cs

# Navigate through: Edit -> Project Settings -> Graphics -> scroll down to Built-in Shader Settings / Always Included Shaders, and increase "Size" by 1, then click the circle on the right and find "DualFilterBloomNoHDR":



# For those objects on which you want to apply these effects, modify their shaders: Create a new shader, and copy the contents of your former shader into it. If you're using unity's built-in shaders, you can go to <https://unity3d.com/get-unity/download/archive> to download them from the drop-down menu:



Note that we can only modify vertex-fragment shaders, and we can't modify surface shaders using this method (because of using MRT).

Then modify the newly created shader according to BloomNoHDR.shader. Find the fragment function, and change its return type to mrtOutput, then add a definition of struct mrtOutput. After that, add the contents from "//MRT" to the end of the fragment function. **Here "col" is your former return value**, replace it. Finally add properties shown in BloomNoHDR.shader except \_MainTex.

# Create a new material for this shader, and apply it on those intended objects. You can now set the "Threshold Luminance" on the material and this value will apply on all intended objects universally, or you can separately set the "Threshold Luminance" on different objects. If you want to separately control the strength of bloom on each object, you can set the "Object Bloom Strength" or even write a script to make it breathe (as BloomBreath.cs). Note that this setting is restricted in range between 0 to 1, which means you can only DECREASE the strength here. The way to intensify the bloom strength will be introduced in the next step.

# Add ColorGradingBloomNoHDR.cs as a component to your camera. After that, set the "Bloom Color". And you can set the "Global Bloom Strength" above 1 to intensify bloom of ALL objects.

# To use color grading, download "Cinema Theme" from asset store. Find the script component we added on the camera and drag a LUT (a texture) into "Color Grading LUT". After that, just tick the "Enable Color Grading" box.

---

### **Translucent:**

# On top of other files, copy this following shader into your project:

../Assets/Shaders/BloomNoHDRTranslucent.shader

# According to this shader, change the blending mode to "One OneMinusSrcAlpha". And in the end of fragment function, add " \* col.a" to specific places as the comment instructed.

### **How to tweak? For artists:**

Now if it can run without problem, you can tweak the following places to make it look better:

# On material:

1. If you set "Threshold Luminance" high, only very bright areas on the object may bloom.

2. For each specific object, you can tweak "Object Bloom Strength" to turn bloom on or off, and you can even attach BloomBreath.cs to an object to make it look like breathing.

# On script:

1. Increase "Bloom Extend" and the bright fringe will bleed further away from the object. When the fringe extends too far, the bloom may look fading out. At this time you can increase "Global Bloom Strength".
2. Adjust "Global Bloom Strength" to make bloom brighter or darker.
3. You can change the color of bloom independent from objects by changing "Bloom Color". A small trick is that, you can make the color-saturation higher for the bloom, and make the object itself whiter and less saturated, in order to make a lamplight-ish effect. Just like the demo project, the bloom is blue and the objects are white.

Author: Marcus Xie