ULTIMATE BLOOM

Documentation V. 1.07

Support, Questions, Comments, Bug report
mikael.paroxe@gmail.com

TABLE OF CONTENTS

RELEASE NOTES

VERSION 1.07

Changes & Fixes

- The bokeh flare renderer have been optimized
- The Ultimate Bloom folder is now movable
- The stability filter will now clamp values to avoid out of range errors
- More options have been added to the bokeh lens flares

VERSION 1.06

New Features

- A new temporal stability filter have been added.

VERSION 1.05

New Features

- Relative sampling have been added to resolve the common inconsistent bloom size relative to the screen resolution

Bug Fixes

- Direct downsampling optimization now works with filmic bloom curve

Changes

- The UI have been refined
- Some default values have changed to better ones based on usage experience

VERSION 1.04

This version fixes some shader compilation errors when creating the build in Unity 5.

VERSION 1.03

New Features

- More quality settings have been added to bokeh lens flares
- Some quality bokeh textures have been added as samples for the bokeh lens flare and the dirt texture generator

Bug Fixes

- A compilation error in the build phase have been fixed in the bloom curve class
- The shader folder have been renamed to Resources to make sure that the shaders are included in the build.

Changes

- The bokeh lens scale will now be relative to the buffer size (specified by the quality settings)

VERSION 1.02

New Features

- Two optimization flags have been added (Direct downsampling and Direct upsampling)

Changes

- Ghost and bokeh lens flare have been optimized a lot
- More informations have been added to the performance section
- A build phase compilation error have been removed

VERSION 1.01

New Features

- Anamorphic flares can now be set to vertical

Bug Fixes

- Values are now clamped to avoid out of range half-float when dealing with HDR in Unity5

Changes

- Bokeh lens flare default values have been set to a better looking default
- Added more information to the performances section
- All C# warnings have been removed
- Some files have been moved to create a cleaner file hierarchy

ABOUT ULTIMATE BLOOM

Ultimate Bloom has been developed to bring an unprecedented artistic control over Bloom rendering to Unity. While developing Ultimate Bloom, the objective was to create a production ready and full featured bloom engine which is flexible enough to be compatible with most game graphics pipeline.

At its core, it uses state of the art algorithms to render high quality bloom at a low performance cost. The effect is rendered using a flexible layered bloom algorithm which gives a total artistic control over the rendering process.

FEATURES

- Flexible layered bloom algorithm with in-depth artistic control
- High-end core engine featuring high quality at low cost
- Novel Filmic Bloom Curve giving unprecedented control over bloom rendering
- Modular architecture: it's possible to disable unnecessary features to boost performance
- Bokeh Lens Flare, Anamorphic lens flare, Star lens flare, Ghost lens flare
- Lens dirt mask
- Lens dirt texture generator
- Anti-"downsize artefacts" and Anti-"jiterring" algorithm
- Compatible with HDR and non-HDR pipeline
- Linear and gamma space compatibility

QUICK START

To add Ultimate Bloom to your camera, go to: "**Component/UltimateBloom/Add to selected camera**"

You can configure the size and intensity of the bloom in the bloom settings menu. You should be able to adjust how the intensity is rendered by choosing an intensity method in the intensity settings menu.

THE CORE ALGORITHM

The bloom rendering algorithm used in Ultimate Bloom works by rendering and combining multiple layers. A bloom pass is computed for each layers on progressively smaller images while keeping the same blur radius in pixels. This means that the first layer have the smallest bloom radius and for each subsequent layers, the bloom radius is progressively higher. All layers are then combined and added to the source image.
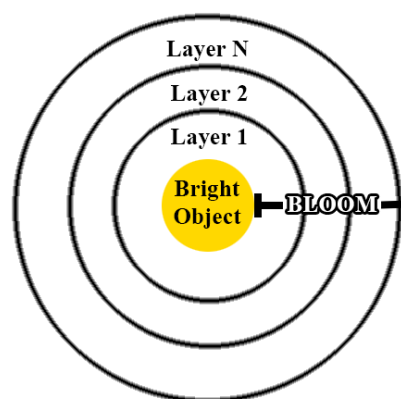
BLOOM SETTINGS

HDR: If HDR is enabled, the system will compute bloom with HDR textures (HalfFloat). Otherwise, it will use LDR textures (RGBA8).

Layers: The number of layers which is also the number of time the input image is downscaled. It will also set the radius of the bloom.

Layer N Enabled: If disabled, bloom won't be computed for the layer N

Layer N Intensity: The intensity of the bloom for the layer N

Layer N Tint: The tint of the bloom for the layer N.
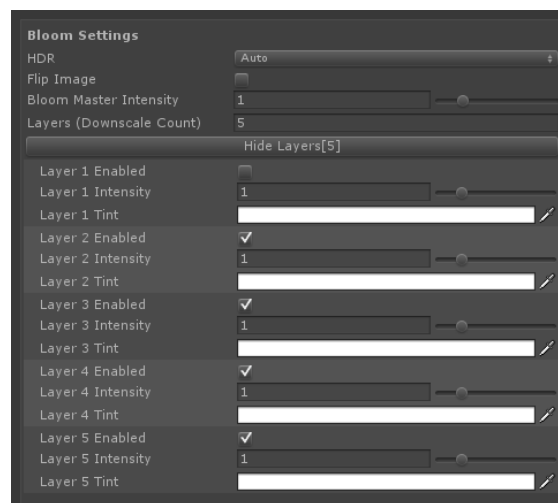
Temporal Stability Filter: With this feature enabled, a temporal aware fetching will be done in the downsampling phase. This will eliminate most of the bloom jittering when moving the camera.

Sampling Mode: You can either choose Fixed sampling or Height relative sampling. If you choose fixed sampling, the blur kernel of the bloom will remain at the same size in pixels for all screen resolutions. This will result in a smaller bloom size in larger resolution since the kernel won't scale up with the screen resolution.

 If you choose Height Relative, the blur kernel size will get bigger as the resolution gets higher. It works by choosing the best blur kernel between 6 different kernel size. You must choose the minimum height of your target screen, and the 6 blur kernel will get associated with their optimal resolution. The minimum height also defines the bloom size, you might want to set the minimum height to a lower value than your target minimum resolution to get a bigger bloom radius.
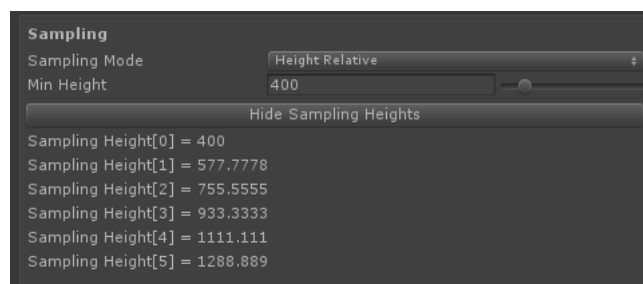
Direct Downsampling Optimization: If this optimization flag is enabled, the downsampling process won't be progressive anymore. This will introduce jittering and it should be enabled only on low end hardwares.

Direct Upsampling Optimization: If this optimization flag is enabled, the upsampling process won't be progressive anymore. The difference in quality is hardly noticeable especially if a medium or a large upsampling kernel is used (it is, however, more noticeable with anamorphic lens flare). This optimization can provide a great boost in performance.

There is more details in the **Performance Optimization** section.
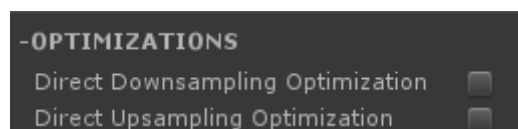
Intensity Management: If you choose Threshold, the bloom will be computed by blurring colors over the intensity threshold. If you choose Filmic Curve, the bloom will be computed by blurring all the images and by multiplying the intensity by the Filmic Curve.

The filmic curve give you more control over the bloom rendering, especially if you use an HDR camera.

Filmic curve settings:

- **Minimum Intensity:** The minimum intensity (threshold) at which the bloom is computed

- **Maximum Intensity:** The maximum intensity treated by the curve. In a HDR pipeline, you can set this value higher than 1 to compute HDR values.
- **Mid-Value Range:** The position of the transition between the bright and the dark intensities. This value set which intensity range is dark and which intensity range is bright. [0= minimum intensity, 1 = maximum intensity, values in between are interpolated]
- **Dark Colors Intensity:** The intensity of the bloom in the dark colors range.
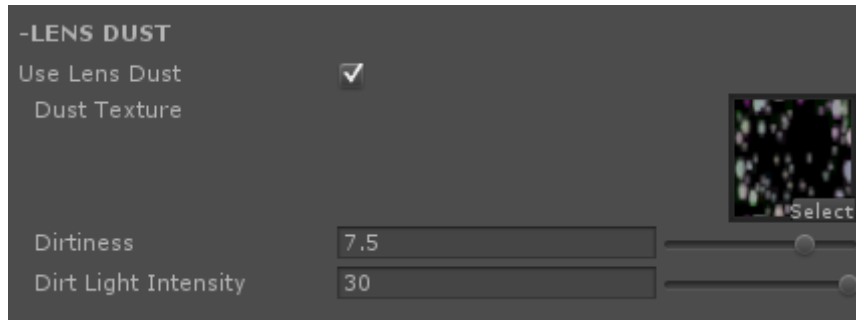- **Bright Colors Intensity:** The intensity of the bloom in the bright colors range.
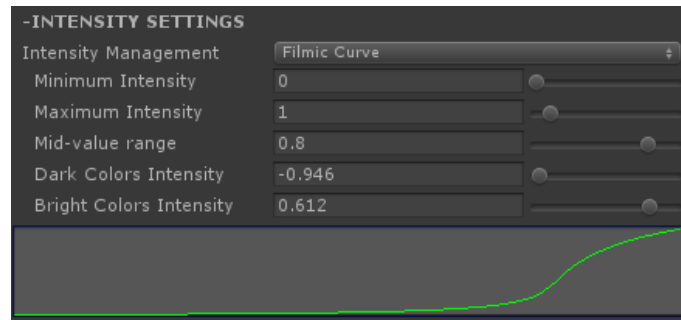
Dirt Texture: The texture mask used as lens dirt

Dirtiness: How much the dust texture appear when affected by bloom

Dirt Light Intensity: How much the lens dust will absorb or transmit light

Flare Rendering: Flare can be blurred if necessary. Flare without bokeh texture will most likely look better when blurred.

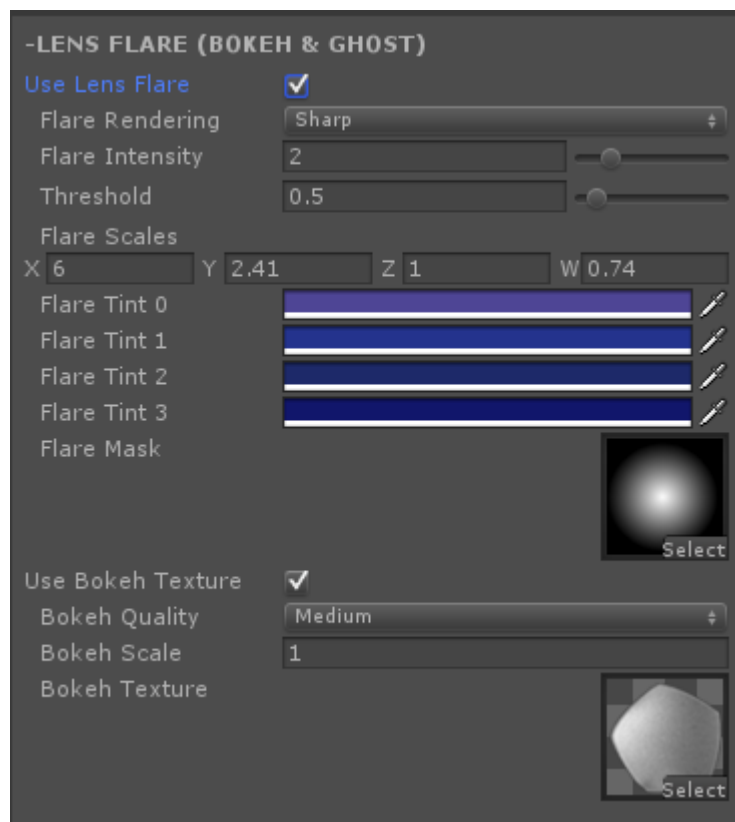**Flare Intensity:** The intensity of the lens flare

Threshold: Flare will be computed on intensities higher than the threshold. The flare threshold is combined with the bloom threshold if the intensity function is set to threshold.

Flare Scales: The scale of each of the 4 instances of the flare.

Flare Mask: The flare mask is used to hide the ugly borders of the flares caused by the limitation of the screen space algorithm.

Bokeh texture settings:

- **Bokeh quality:** An higher quality will increase the buffer size used to render the bokeh texture. Bokeh rendering can be really expensive and the performance depends on how many pixels is over the bokeh flare threshold. You should make sure that the threshold is always high enough to only get the brightest spots of your scene.
- **Bokeh Scale:** The scale of the bokeh texture.
- **Bokeh Texture:** The texture used to shape the lens flare. You can uses the bokeh textures located in the "UltimateBloom/Graphics" folder or your own textures.

Layers: The number of layers which is also the number of time the input image is downscaled.

Intensity: The intensity of the flare

Scale: The scale of the flare. It should not be a lot higher than the number of blur pass, otherwise it will create gaps.
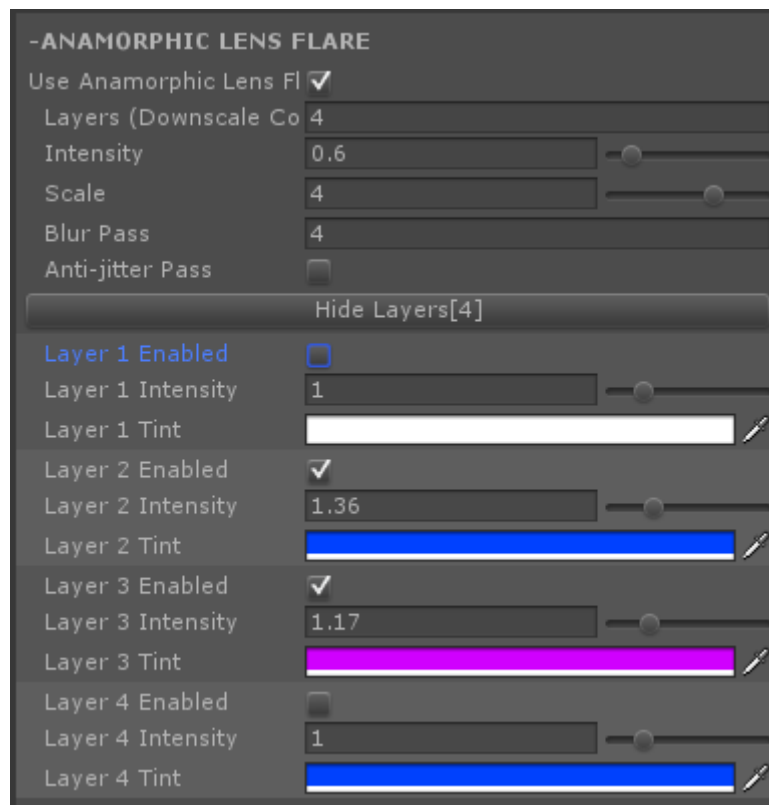
Blur Pass: The number of blur pass per layer. It makes longer stripes.

Anti-Jitter Pass: If enabled, the anti-jitter pass will limit the jittering in the flares. It should be used only if the flare configuration creates jitters in the flare (i.e. by using layers 4+).

Layer N Enabled: If disabled, the flare won't be computed for the layer N

Layer N Intensity: The intensity of the flare for the layer N

Layer N Tint: The tint of flare bloom for the layer N.

GENERAL OPTIMIZATION DISCUSSION

The system is modular which means that if you don't use a layer or a lens flare effect, it won't affect performance. If performance is an issue, disabling functionalities or layers you don't need will help.

The first layer (layer 1) is computed at full resolution, it is by far the most heavy layer to compute (it takes around 4 times more computing time than the second layer). If performance is a problem, you should disable the first layer before anything else (and maybe the second layer for low-end hardwares). The logic is that for every subsequent layers, the computing cost is smaller than the previous layer by approximatively four times, which makes the first layer significantly heavier than every other layers. Additionally, the number of blur pass used on the lens flares can be heavy especially when used on the first layer.

The upsampling kernel size will influence performance too. You may want to set the kernel size to medium or low if performance is an issue.

If you enable HDR on your camera Ultimate Bloom will compute everything in HDR which is more expensive than LDR. You can put Ultimate Bloom after a tonemapping effect with HDR Off, but Ultimate Bloom should look better if computed before tonemapping.

The optimization flags Direct Upsampling and Direct Downsampling only works if some layers are deactivated (ie. layer 1 and layer 2). The core algorithm will progressively downscale and upscale the bloom texture (once for each layers) and by enabling the optimization flag you can force Ultimate Bloom to directly downscale or upscale the bloom texture over the unused layers. This will introduce jittering if direct downsampling is enabled and the bloom will be a little bit less smooth if Direct Upsamling is enabled (it can be unnoticeable).

Using ghost lens flare is fine and should not affect performance a lot. However, using bokeh lens flare (by setting a bokeh texture) can be heavy on the GPU. The performance depends on two things. The first one is the scale and a higher scale will cost more to render. The second one is the number of pixels that are over the flare threshold and you should make sure that only the brightest spots of your scene gets over this threshold(i.e. the sun or a torch in a cave) .

An example of a bloom setup that would run well on a low-end hardware could look like this:
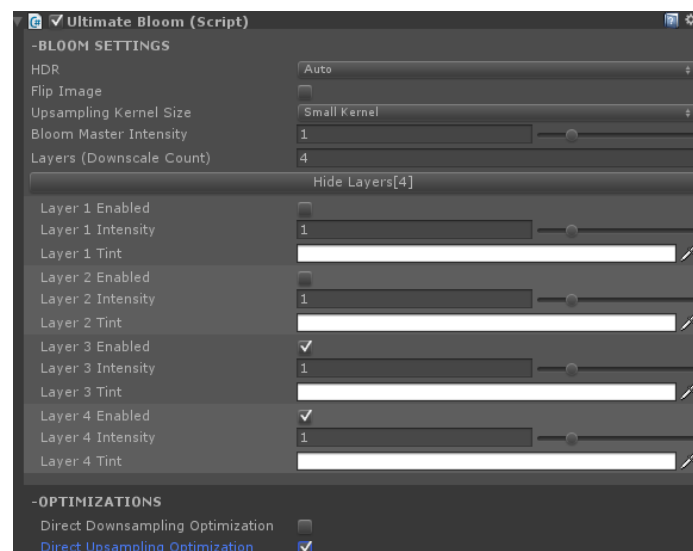
OPTIMIZING ACROSS DIFFERENT TARGET HARDWARES

The best way to optimize bloom rendering across hardware's is to remove the higher frequencies of the bloom. Those higher frequencies are the first layers: Layer 1(the highest frequency), Layer 2(lower frequency than layer 1), Layer 3(lower frequency than layer 2) etc. That way, the bloom will contains less details but the bloom radius will remain the same while getting a much higher performance.
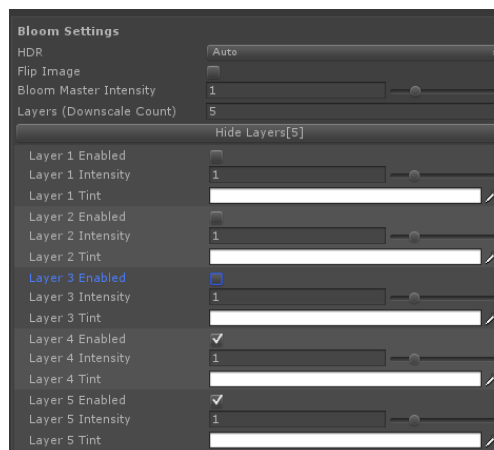
As an example, you could have three different bloom quality settings. The higher quality settings could be:
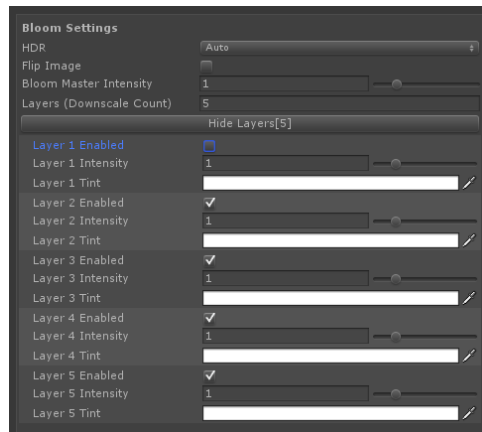
The medium quality bloom could be:
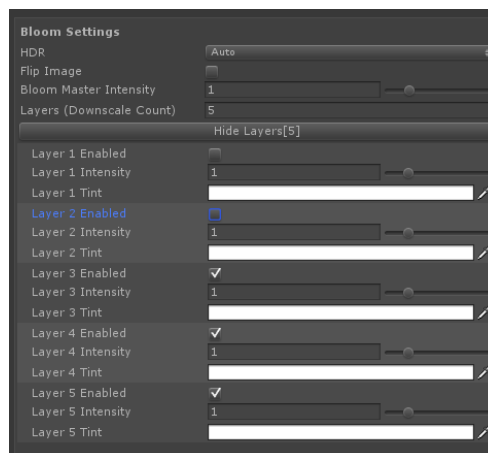


And the low quality bloom could be:

The bokeh texture generator is a simple tool that can be used to quickly create lens dirt textures by using sprites. You can open the window by clicking on "**Window/Bokeh Texture Generator**".

CONFIGURATION

Before anything, you need to create a target texture. You can set the width and the size that you want and click "create and set texture" or you can choose an existing texture. The texture will be named "**Assets/UB_BokehTexture.png**".

The bokeh texture generator for lens dirt works by applying lens effects to randomly generated sprite, for a number of layers. You can choose the number of layer by setting the property "**Layer Count**". For each layers, you can set those properties:

- **Sprite Texture:** The sprite used to populate the layer (the sprite pixels should contains transparency values to make the instances blend well together). You can use the textures located in the "Paroxe/UltimateBloom/ImgForDirtGenerator" folder.
- **Minimum/Maximum Size:** The size range of the sprite in pixels
- **Blur Radius:** The Gaussian blur radius
- **Vignette Power:** This value determine how much sprite should appear exclusively around the edge of the texture.
- **Hue Variation:** The variation of color
- **Intensity:** The intensity of the sprite
- **Density:** The density of the sprite in the image (higher value means more instances)
- **Chromatic Aberration:** The force of the color distortion effect
- **Random Rotation:** If enabled, the sprite will be randomly rotated per instance
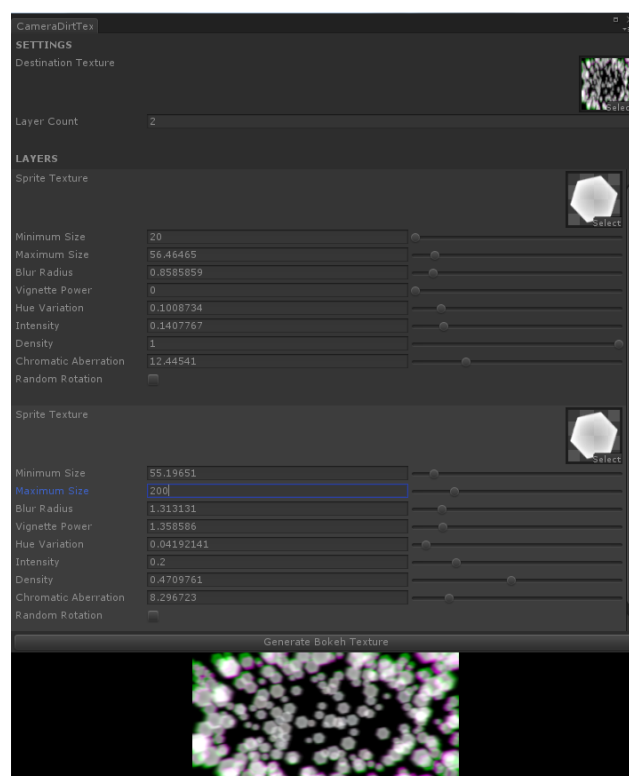
SCRIPTING

If you want to programmatically change parameters, you can look at the UltimateBloomEditor class located in the Paroxe/UltimateBloom/Editor folder. It contains examples on how to modify every parameters of UltimateBloom in C#.