

SC Post Effects

by Staggart Creations

1. SC Post Effects	3
2. To be added	3
3. Getting started	3
3.1. Installation	4
3.2. Setup	5
3.3. Profiles	6
3.4. Volumes	7
4. Compatibility	8
5. Known Issues	8
6. Refunds	10
7. Effects	10
7.1. Black Bars	10
7.2. Color Washing	11
7.3. Color Split	12
7.4. Color Depth	13
7.5. Danger	13
7.6. Double Vision	14
7.7. Dithering	15
7.8. Edge Detection	16
7.9. Fisheye	17
7.10. Fog Gradient	18
7.11. Kaleidoscope	18
7.12. Invert	19
7.13. Kuwahara	20
7.14. Pixelize	20
7.15. Refraction	21
7.16. Ripples	22
7.17. Scanlines	23
7.18. Screen Gradient	24
7.19. Sharpen	24
7.20. Tube Distortion	25

1. SC Post Effects

Thank you for purchasing SC Post Effects!

This package offers 25 non-photo realistic (NPR) post effects to enhance your art style or feedback. Made to reap the full benefits of the Post Processing Stack v2.

This package is in its beta stage, some features might not work correctly, or are in an early stage. The final release price will be set at \$30

Feedback and suggestions can be made in the forum thread:

<https://forum.unity.com/threads/513191>

This manual is intended to clarify the use of the package.

2. To be added

Currently, getting all the effects to a working standard is priority.

Other effects may be added in the future, suggestions are welcome!

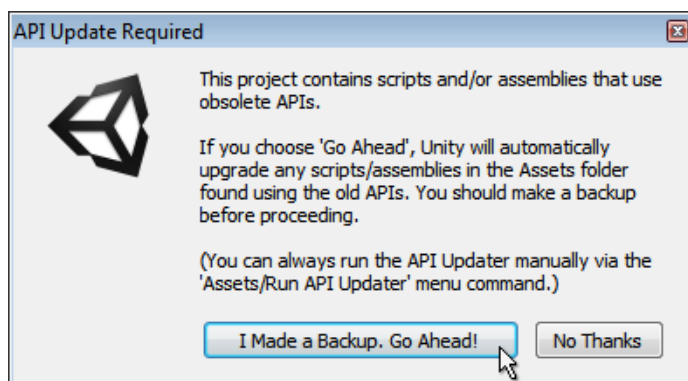
3. Getting started

Import the Unity package through the Asset Store download manager (Found under **Window->Asset Store**).

All the effects are contained in separate folders, so you easily leave out any effects you don't want.

Be sure to include the "Core" folder, it is not required for the effects to work, but it contains installation and help functionality.

You may receive a warning message about obsolete APIs, it is completely safe to run the API updater.



You will be presented with the installation screen, should anything not check out, instructions will be displayed



In Unity 2018.1 you may see a number of errors that read:

failed to open source file: 'PostProcessing/Shaders/StdLib.hlsl' at line 6 (on d3d11)

This is because the Post Processing package hasn't been installed yet. Unfortunately, unlike in scripts, there is no way to check for the installation in shaders, in order to avoid this.

After correct installation, these errors will be cleared!

3.1. Installation

This package is required to be used with the Post Processing Stack v2, which is currently in beta. **It is not backwards compatible with v1.**

Unity 2017.3 and older

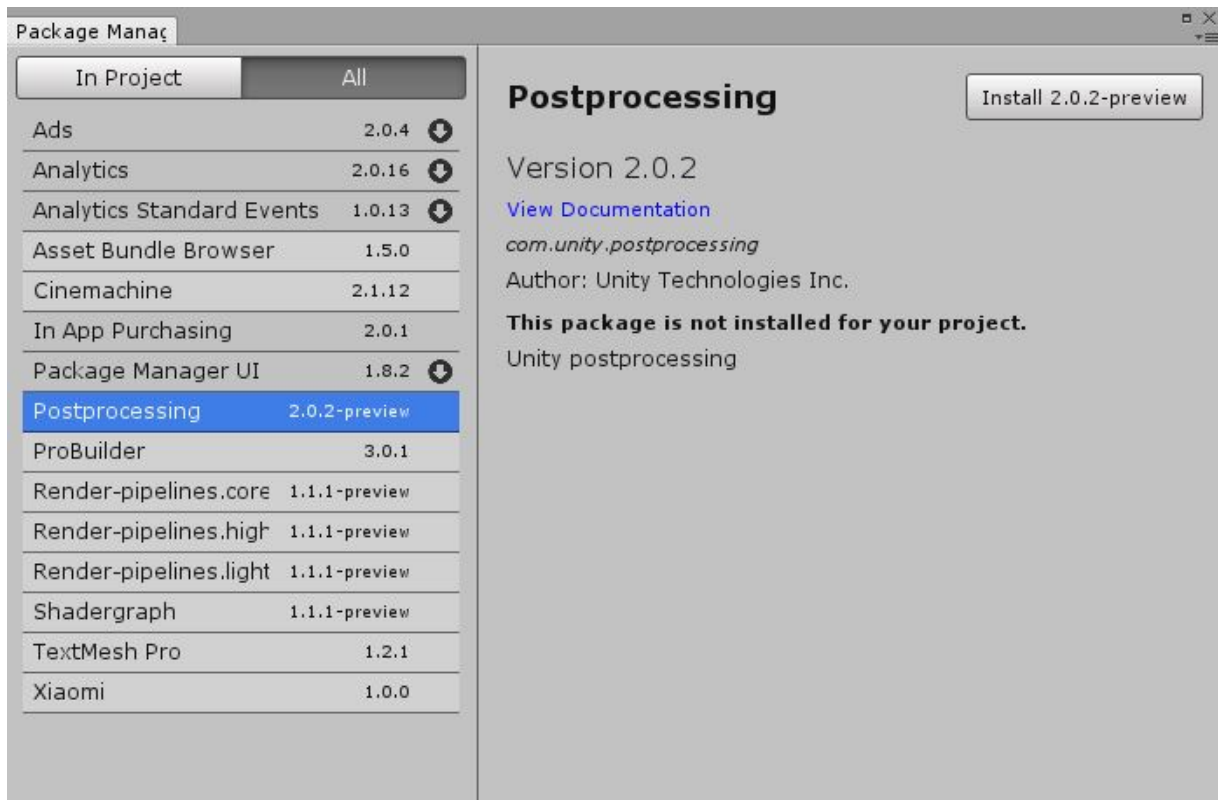
The PPS has been downloaded from GitHub and packaged:

[DOWNLOAD UNITY PACKAGE](#)

This package is always updated with the latest compatible version

Unity 2018.1

Go to Window -> Package Manager



Install the *PostProcessing* package and you're set!

3.2. Setup

If this window is not already open, you can find it under **Help -> SC Post Effects**

It allows you to automatically set up a layer, and add the Post Processing Stack to your scene.



Manual set up:

Project setup

- Go to Edit->Project Settings->Tags and Layers.
- Add a new layer, dedicated to post processing.

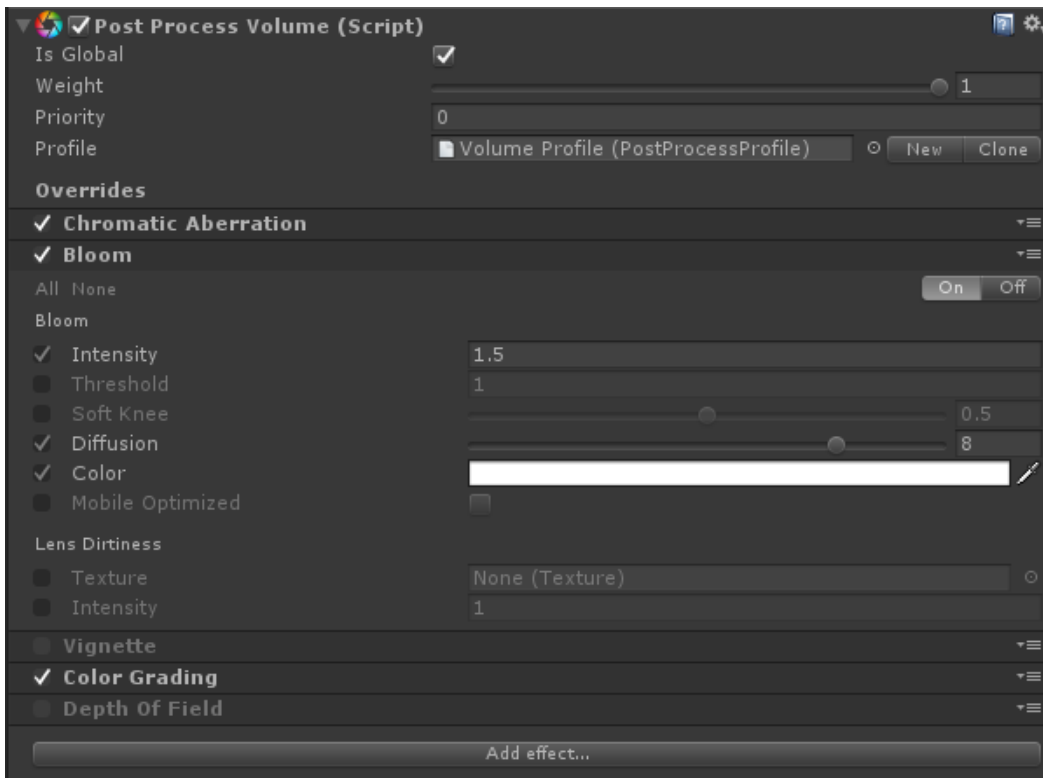
Camera setup

- Select your main camera, and add a “Post Process Layer” script component.
- On the component, in the “Layer” dropdown, select the layer you created a moment ago.

Post Processing Volume Setup

- Go to **GameObject -> 3D Object -> Post-process Volume** to create a new volume.
- Assign the layer to this object
- Tick the “Global” box
- Choose a profile in the profile field

3.3. Profiles



Each field has an override checkbox on its left, you'll need to toggle the settings you want to override for this volume before you can edit them. You can quickly toggle them all on or off by using the small All and None shortcuts at the top left.

The top-left On/Off toggle is used to override the active state of the effect itself.

Finally, you can right-click and effect title to show a quick-action menu.

From there you can start adding local volumes with various priorities and blend distances to your scene and see all of them blend automatically.

There are a few example profiles in the *SC Post Effects/_Examples/Profiles* folder you can try out.

A demo scene, where all the effects can be viewed individually will be available closer to the final release.

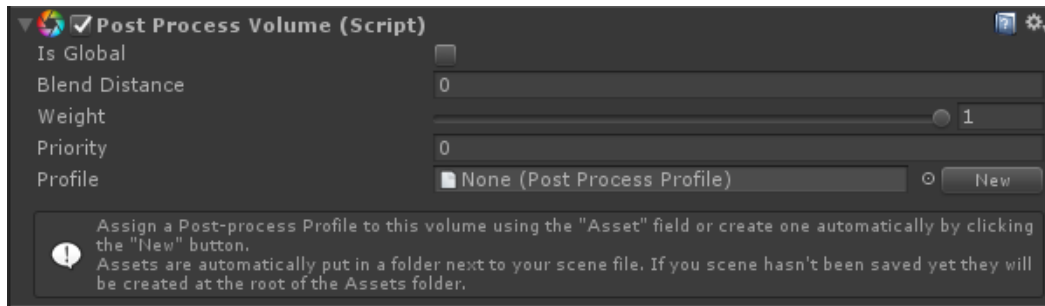
3.4. Volumes

(Excerpts from GitHub documentation)

The way post-processing works in this framework is by using local & global volumes. It allows you to give each volume a priority and a set of effect overrides to automatically blend post-processing settings in your scene. For instance, you could have a light vignette effect set-up globally but when the player enters a cave you would only override the intensity setting of the vignette to make it stronger while keeping the rest of the settings intact.

The Post Process Volume component can be added to any game object, the camera itself included. But it's generally a good idea to create a dedicated object for each volume. Let's start by creating a global Post Process Volume. Create an empty game object and add the component to it. Don't forget to add it to a layer that's being

used by the mask set in the Post Process Layercomponent you added to your camera!



By default it's completely empty. From there we can do two types of volumes:

- **Global:** a global volume doesn't have any boundary and will be applied to the whole scene. You can of course have several of these in your scene.
- **Local:** a local volume needs a collider or trigger component attached to it to define its boundaries. Any type of 3D collider will work, from cubes to complex convex meshes but we recommend you use simple colliders as much as possible, as meshes can be quite expensive to traverse. Local volumes can also have a Blend Distance that represents the outer distance from the volume surface where blending will start.

We want a global volume, so let's enable Is Global.

Weight can be used to reduce the global contribution of the volume and all its overrides, with 0 being no contribution at all and 1 full contribution.

The Priority field defines the volume order in the stack. The higher this number is, the higher priority a volume has.

4. Compatibility

VR

Compatibility has been untested. Theoretically, the shaders are set up to work with Single-Pass Stereo Rendering, but at this time I cannot provide certainty.

Mobile

Mobile devices are bandwidth limited, so post processing must always be used sparingly. Multi-pass effects such as Blur and Lens Flares should be avoided.

This platform has not yet been tested for artifacts or performance, so it is currently not considered compatible.

5. Known Issues

Consoles errors/warnings

Some shaders throw the following error:

Assertion failed: Failed to create DX11 vertex declaration; something wrong with vertex shader input data? (hr=80070057)

It is still unclear what causes this, but the error only shows when opening Unity, or importing the assets. Other than that, it's harmless. I intend to resolve this in the future.

Failed to open source file: '../Shaders/StdLib.hlsl' at line XX

If you've updated for 2017.3 to 2018.1, you'll have to re-download the package from the Asset Store. A separate version has been submitted for 2018.1+ which works with the file paths of the Package Manager's post processing plugin.

Should this happen in 2017.3 or older, re-import the offending shader. This may happen if you've removed the Post Processing Stack files, or moved them.

The type or namespace name 'PostProcessing' does not exist in the namespace 'UnityEngine.Rendering'. Are you missing an assembly reference?

All the script check for the UNITY_POST_PROCESSING_STACK_V2 define symbol (found in Player Settings). If you delete the Post Processing Stack files (or package in 2018.1) from your project, this directive is not automatically removed, hence these errors will be thrown.

If you want to remove the Post Processing Stack, remove the SC Post Effects folder along with it.

OpenGL

Multi-pass effects such as Blur, Ambient Occlusion 2D and Lens Flares currently do not work correctly.

The Kaleidoscope effect doesn't work on OpenGL at all, the reason is currently unknown.

Limitations

The Fog Gradient and Screen Gradient effects only work with a gradient texture, which is to be authored externally in something like Photoshop (PSD files can be used as textures!).

This is due to the fact that the "PostProcessEffectEditor" class currently does not support intermediate calls. So calling a function when one of the parameters has changed, like in regular editor scripts, is not possible. Everything is to be done in the rendering loop.

It is possible to create a texture from a Unity gradient, this would have to be done constantly at runtime, which is far from efficient, and breaks compatibility with volume blending.

Editor

Some effects that rely on the depth buffer (such as Edge Detection and Fog Gradient) may look slightly different in Scene View, since the far clipping plane varies, depending on your scene.

Fog Gradient

The Fog Gradient effect may show artifacts where the skybox and objects meet.



This is to be resolved in a future update.

6. Refunds

Please refer to [our refund policy](#)

7. Effects

It's important to remember the rendering order can greatly affect the behavior of the effects. You can adjust the order on the "Post Process Layer" component on your camera.

A few examples:

- If the Ripples effect is put before the Edge Detection effect, the edges will not be affected by the Ripples.
- If Tube Distortion is put after Scanlines, the scanlines will also be distorted.

7.1. Black Bars

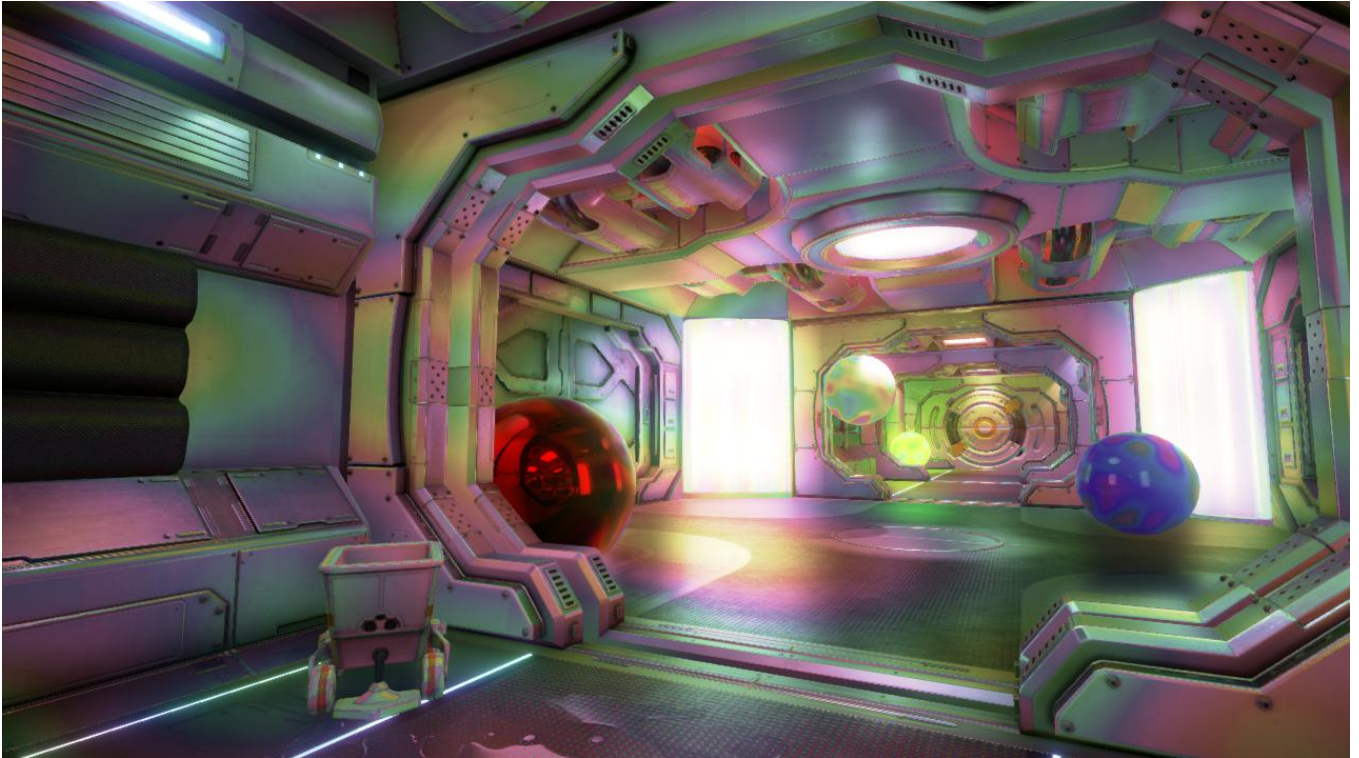
Allows you to add a black bars to your screen, either horizontal (21:9) or vertical (4:3).



7.2. Color Washing

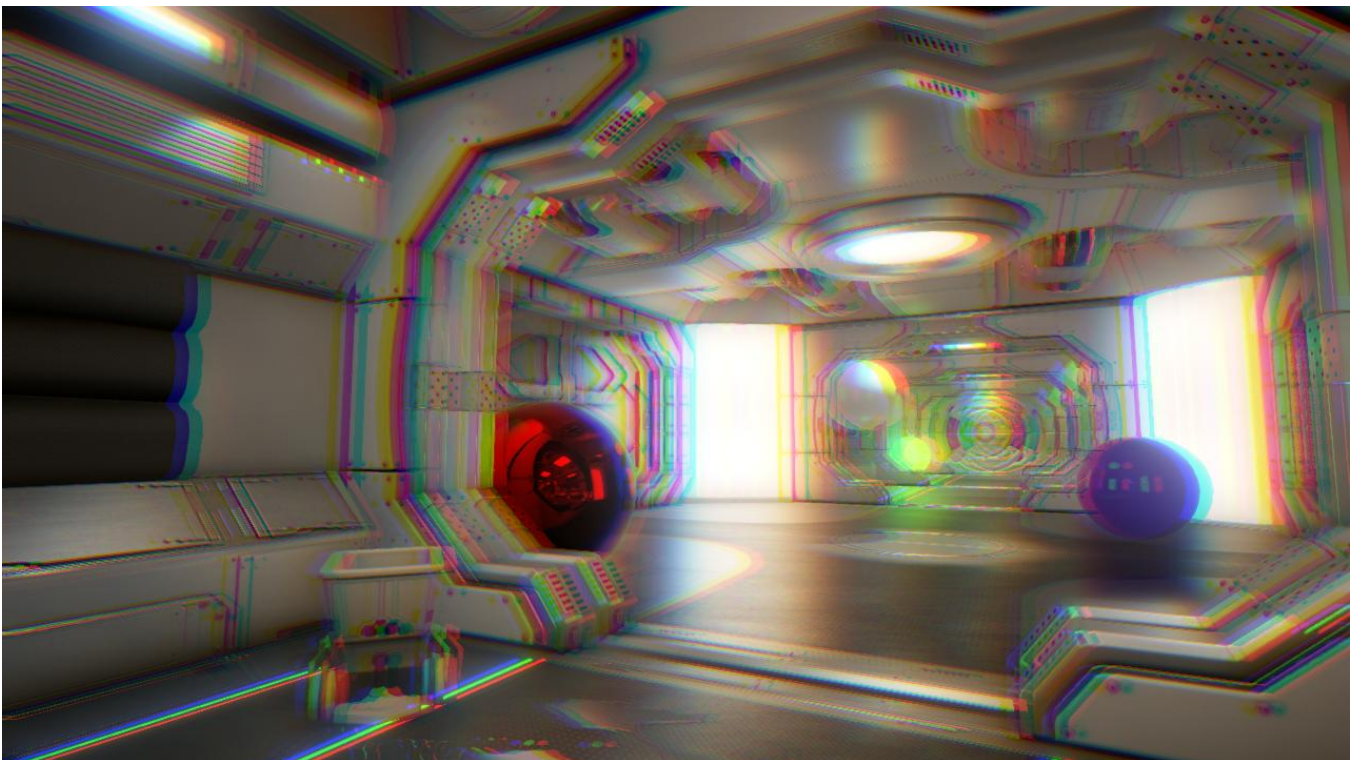
A 3D hue shifting effect.

The “Geo Influence” parameter determines how strongly the effect adheres to the scene geometry. High values may induce banding artifacts.



7.3. Color Split

Offsets the RGB channel, making the colors appear to separate from each other.



7.4. Color Depth

Simulates low color bit depth, as seen in older games when hardware was limited.



7.5. Danger

Can be used to create a effect that simulates pain or poisoning. Especially useful when used with volumes. You could set up a “frost” effect on your Global volume, and create local volumes around say... barrels of fire, that override the “Size” parameter.

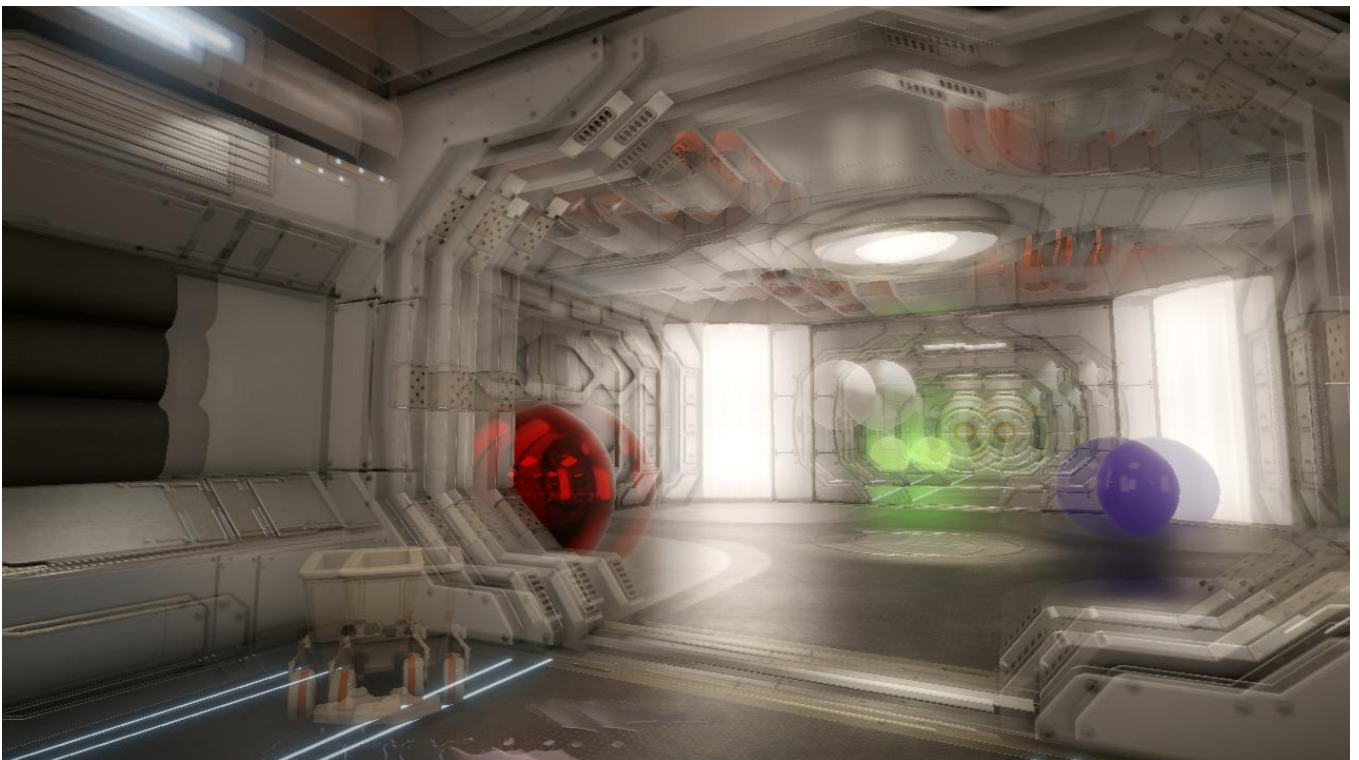
A grayscale texture can be used to improve this effect. A few examples are included in the *SC Post Effects/_Examples/Overlays* folder.

In order for the “Size” parameter to work as intended, your texture must have white values between 0 and 1. The parameter interpolates between these values, making the texture appear to grow towards the center of the screen.



7.6. Double Vision

Makes the camera go cross eyed.



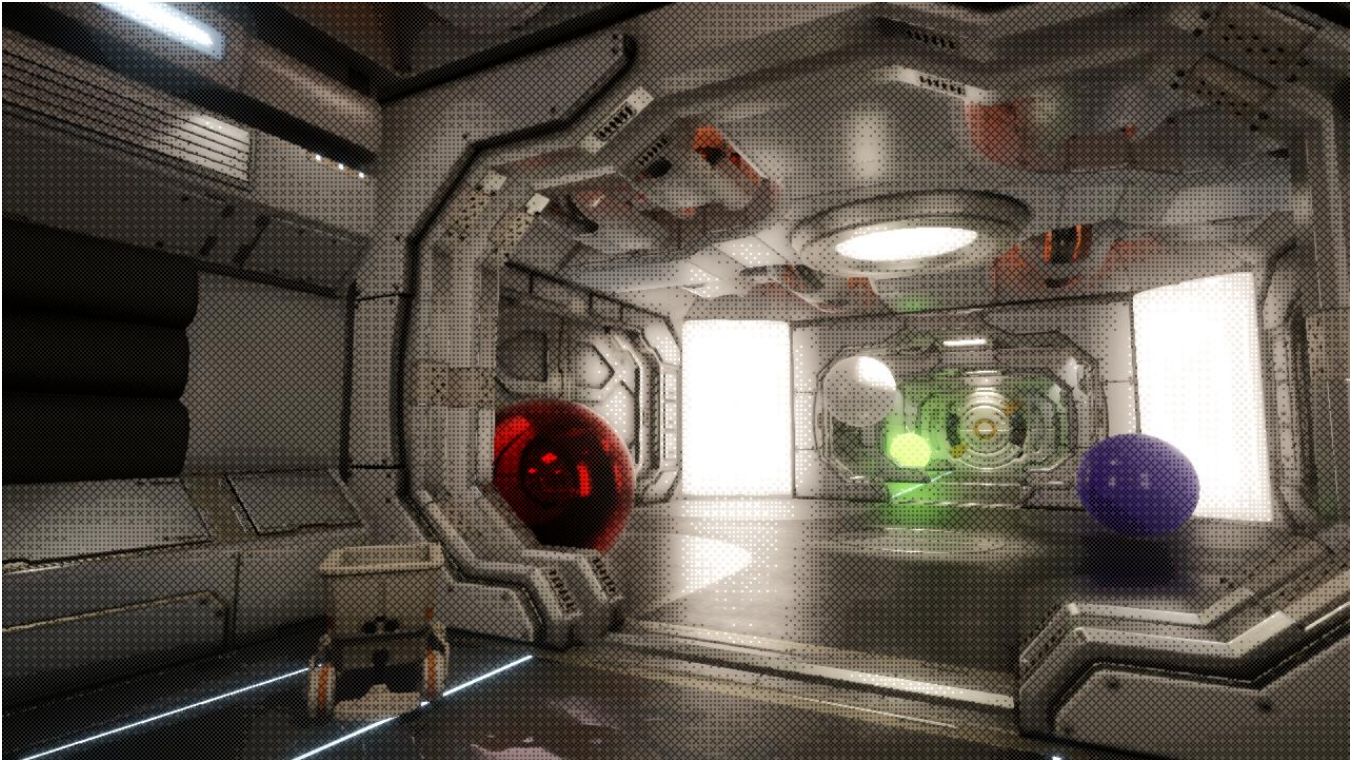
7.7. Dithering

Reduces the image fidelity by omitting certain pixels through a Bayer 4×4 dithering matrix.

Brighter parts of the screen will show less dense dithering, whereas dark areas may turn entirely black.

This effect is currently calculated entirely on the GPU. If this proves to be too inefficient, an option to use a lookup

texture will be added.



7.8. Edge Detection

Draws a line where an edge is detected. This is a revamp of the legacy edge detection image effect. The option to set an edge color and opacity has been added.

Several method can be used, the result of which entirely depends on what is drawn on the screen.

Triangle Depth Normals

Checks the difference between a pixel's normal value and their distance from the camera.

Robert's Cross Normals

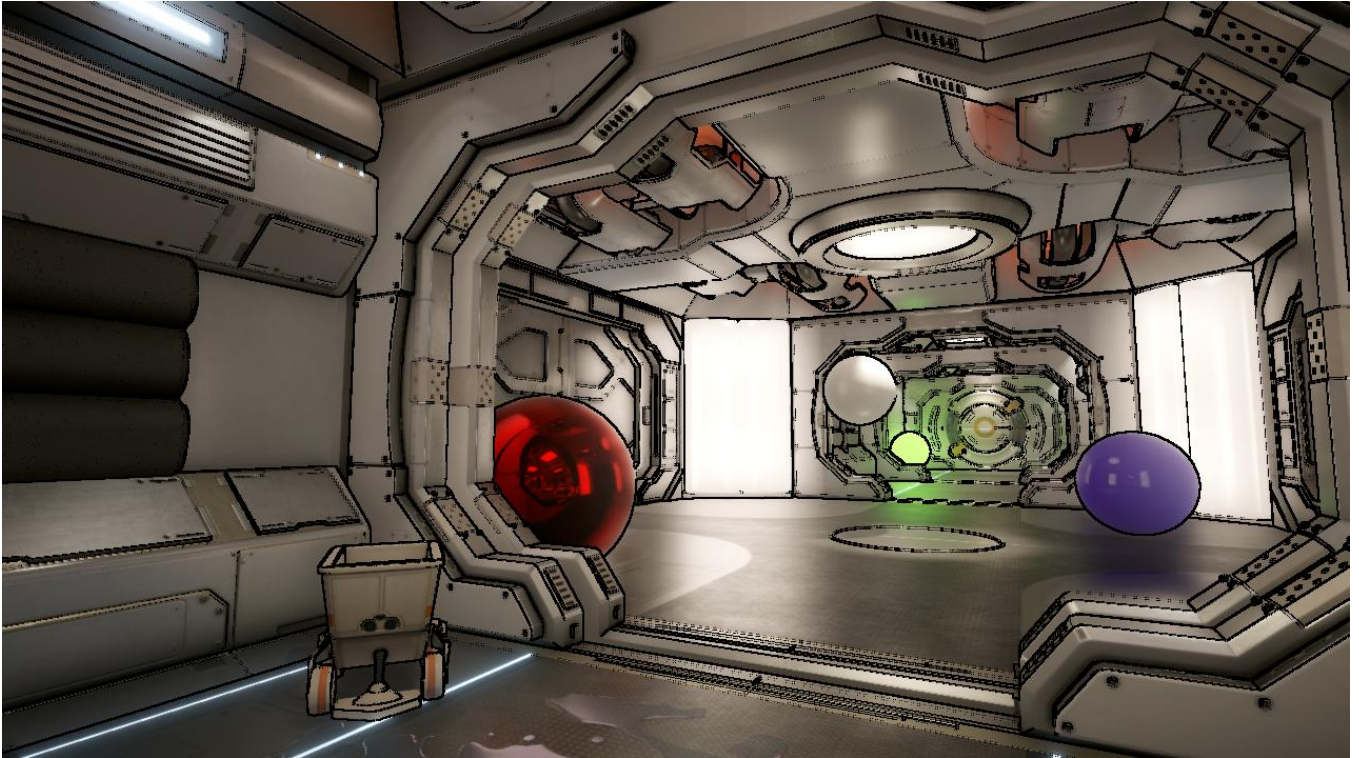
Works the same as the method described above, but uses an addition sample for higher precision.

Sobel Depth (Thin)

Draws an edge where neighboring pixels greatly differ in their depth value. Same method as used in "Borderlands"

Triangle Luminance

Uses the image's luminosity and creates an edge where the values differ. This works great for aesthetics that make use of strong and sharp color difference.



7.9. Fisheye

Simulates a high field of view, often referred to as a “fisheye lens”.



7.10. Fog Gradient

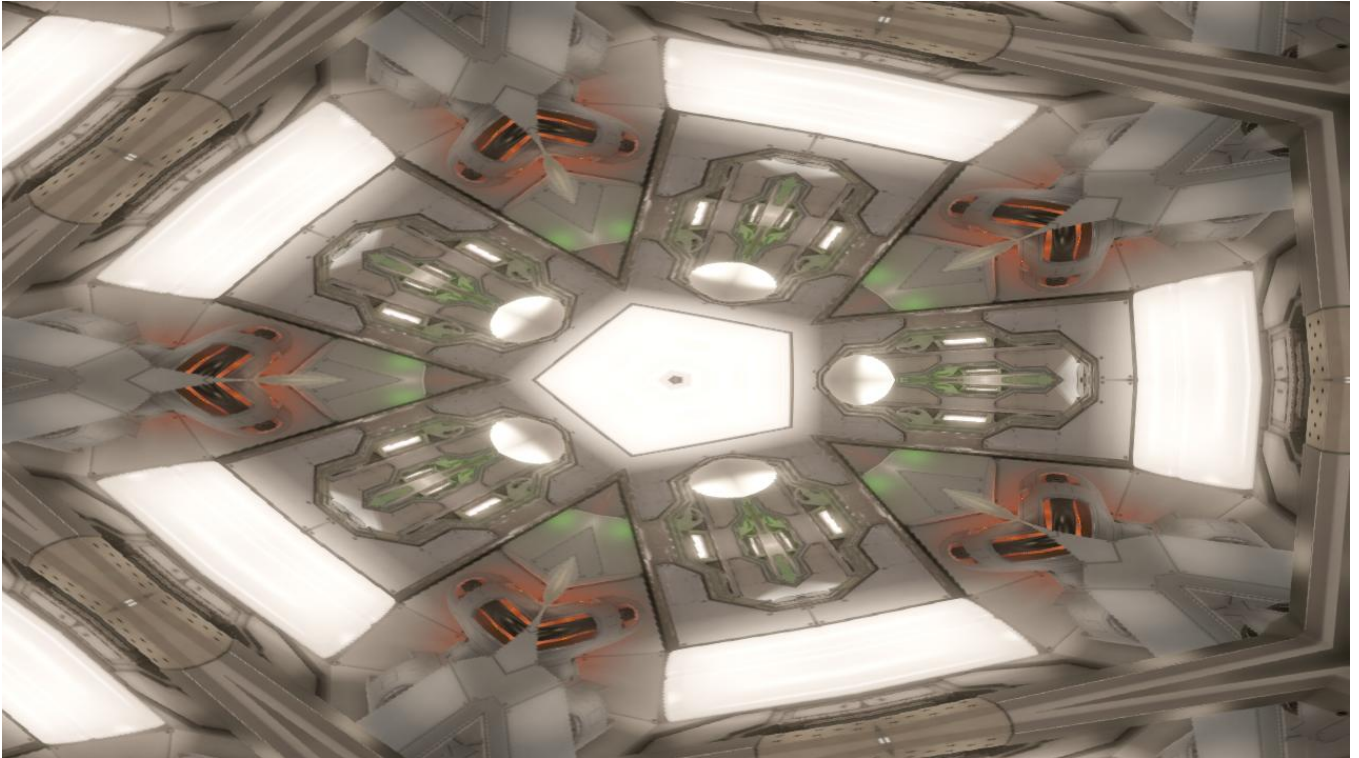
Allows you to use a gradient for the fog, so it's possible to have different colors as the fog goes into the distance.

Note: The default Unity fog is still separate, so its configuration has no effect on this.



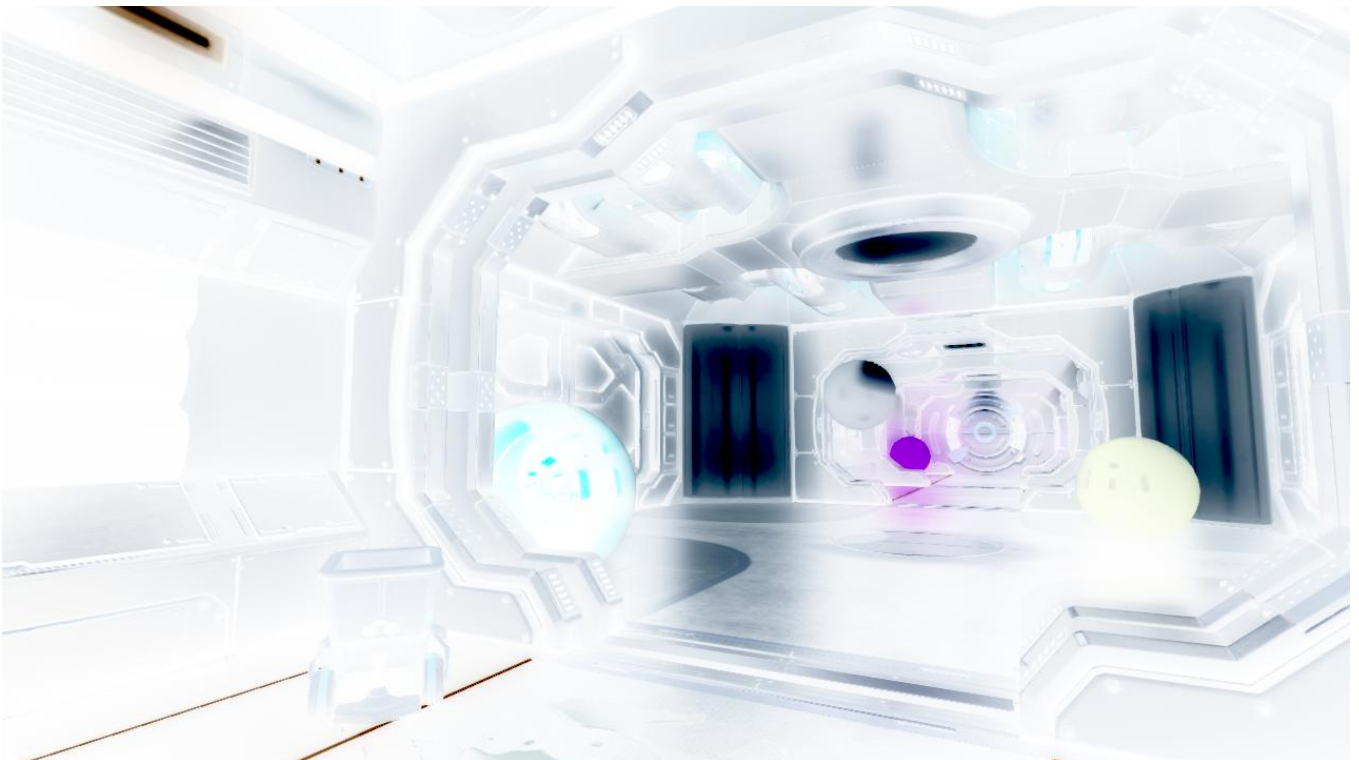
7.11. Kaleidoscope

Tiles the screen in “pie pieces”. Has great uses for non-game purposes, like dynamic art works.



7.12. Invert

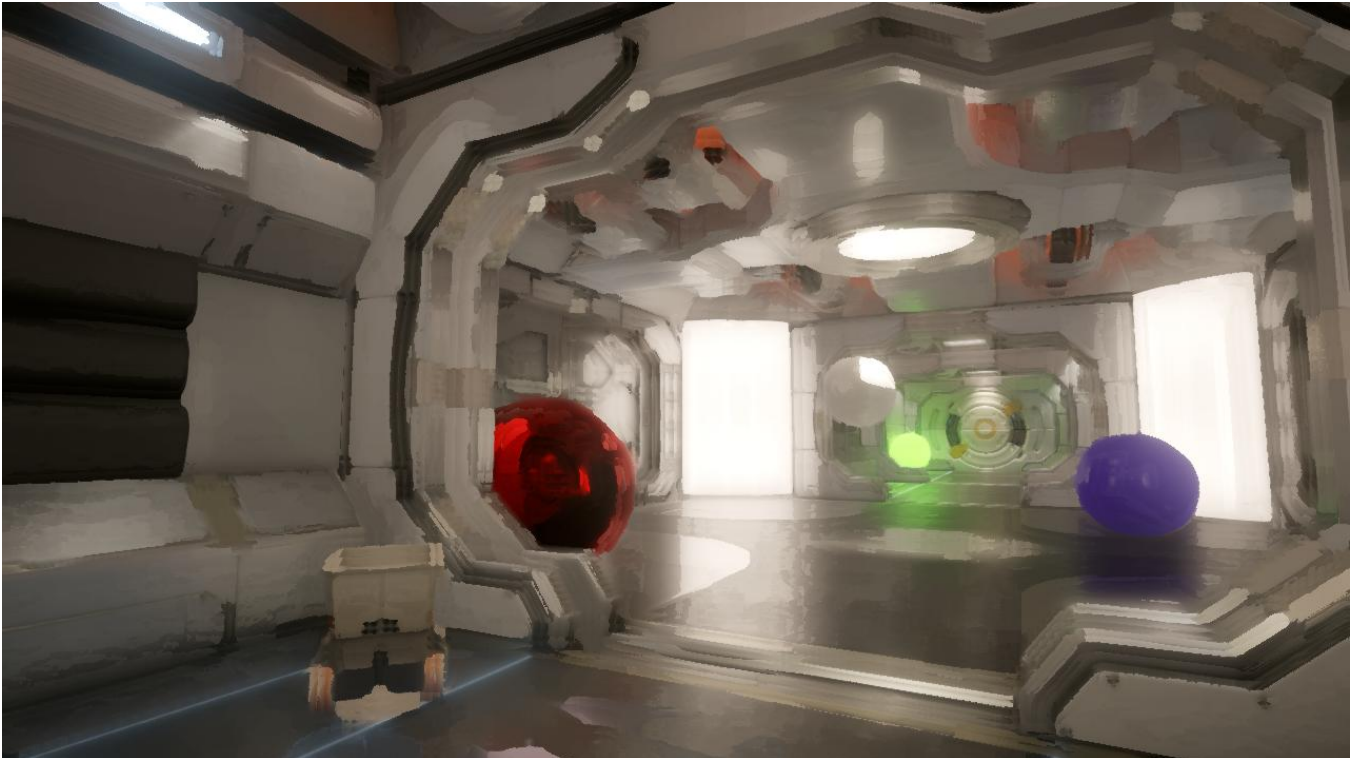
Inverts the screen colors.



7.13. Kuwahara

Transform the image into an oil painting. Has the option to fade out the effect over distance.

This effect is quite heavy, which should be kept in mind when dealing with a performance sensitive project.



7.14. Pixelize

Simulates a low resolution screen to easily transform your image into pixel art.

Note: Does not actually lower the rendering resolution, the effect is only visual.



7.15. Refraction

When supplied with a refraction map this can be used to create cracked glass, frost, scratches, water drops etc.

Several example maps are included:

- Clear ice
- Cracked glass
- Frost
- Scratches
- Shards
- Water drops

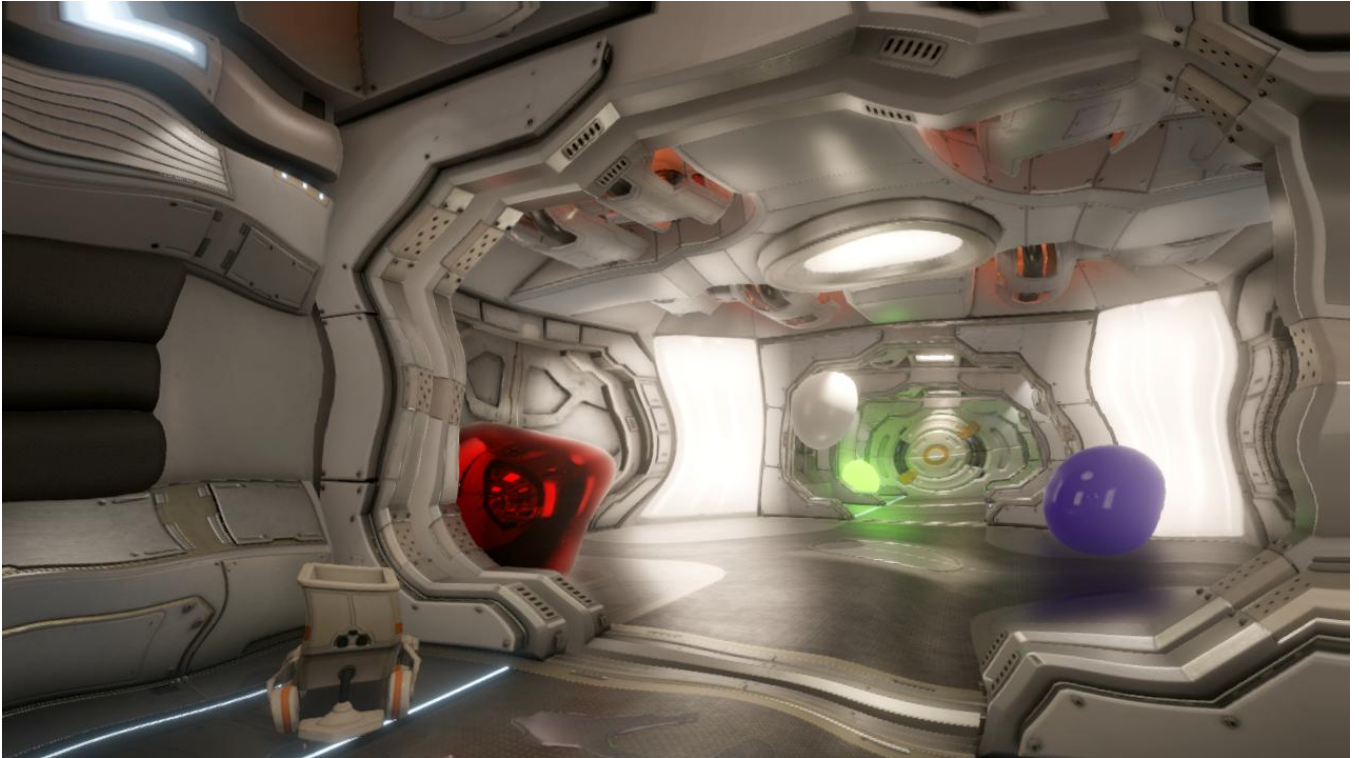
If you'd like to use your own refraction map, you can convert a normal map to a DUDV map, which is similar to a normal map, but doesn't have a blue channel. More information can be found here: http://wiki.polycount.com/wiki/DuDv_map



7.16. Ripples

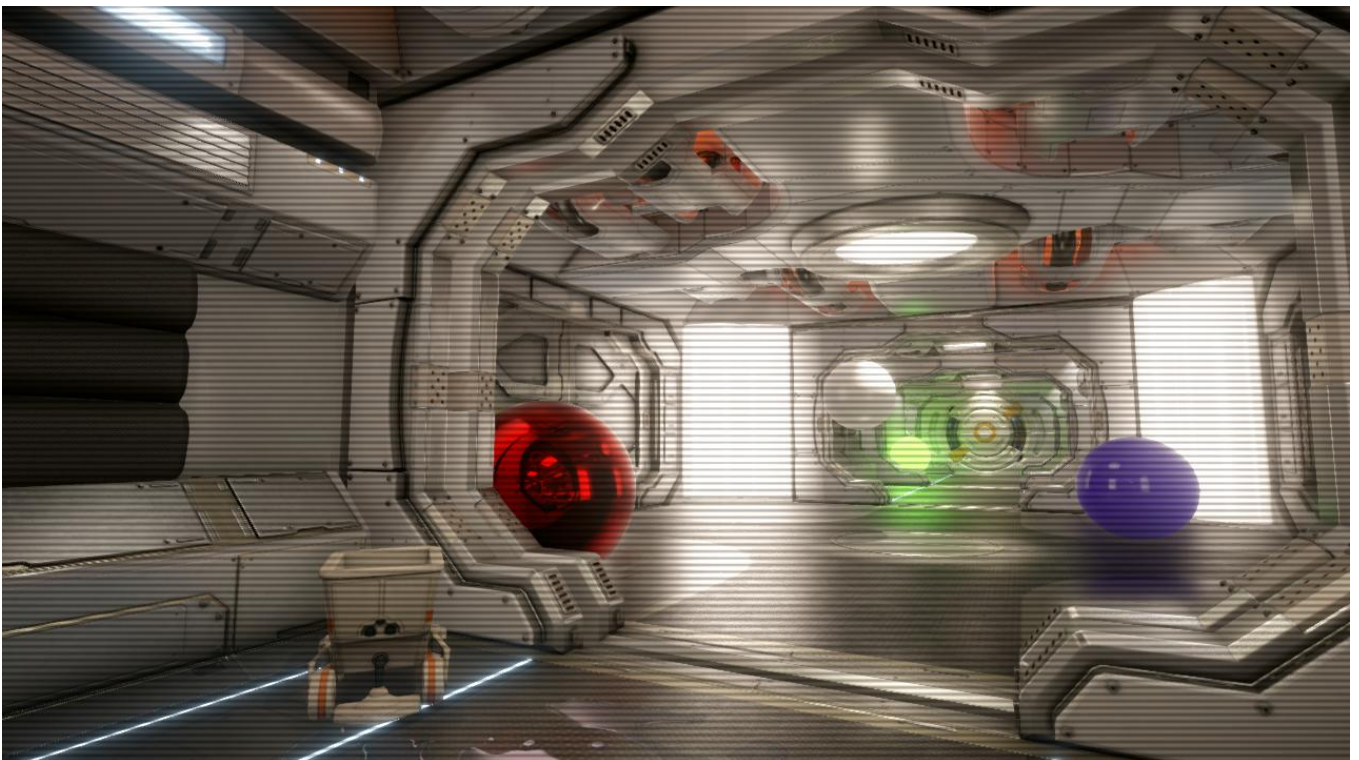
Makes the screen appear to wave. Has both a wave (emitting from the center of the screen) and a omni-directional mode.

Values like strength, speed and distance can be adjusted.



7.17. Scanlines

Draws horizontal lines over the image, to simulate a CRT screen.



7.18. Screen Gradient

Draws a gradient texture over the screen.

This is to be extended with rotation options and blending modes.



7.19. Sharpen

Makes the image appear crisper.



7.20. Tube Distortion

Simulates the curvature of a CRT screen.

Has three modes:

- Bulged
- Pinched
- Beveled

