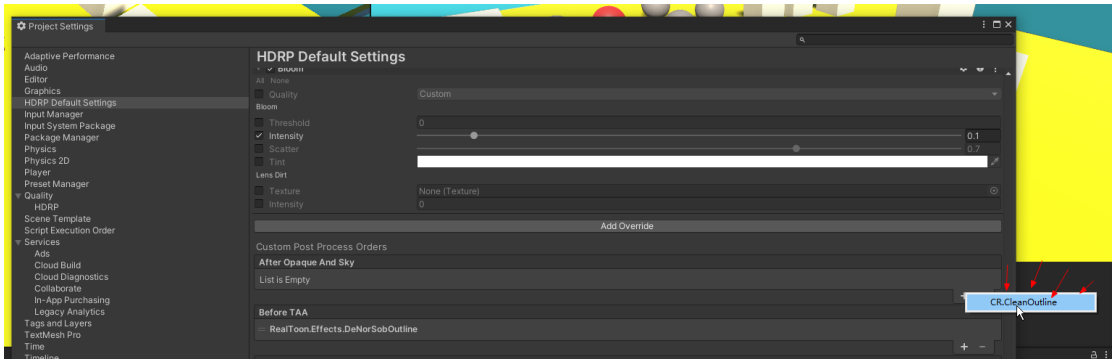
Clean Outline HDRP

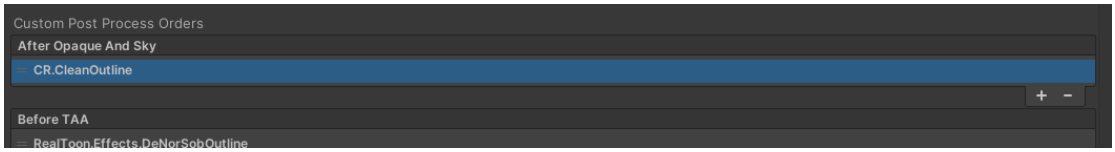
Clean Outline HDRP is a post-processing outline solution for Unity HDRP pipeline. The outline shader will sample the depth and normal or nearby pixels to create outlines.
The general purpose of the shader is very simple, just to focus on creating outline without adding other effects to it. And it already considers some case like the distance of pixel and the threshold of the calculations.

Requirement:
Unity 2020.3x+ HDRP

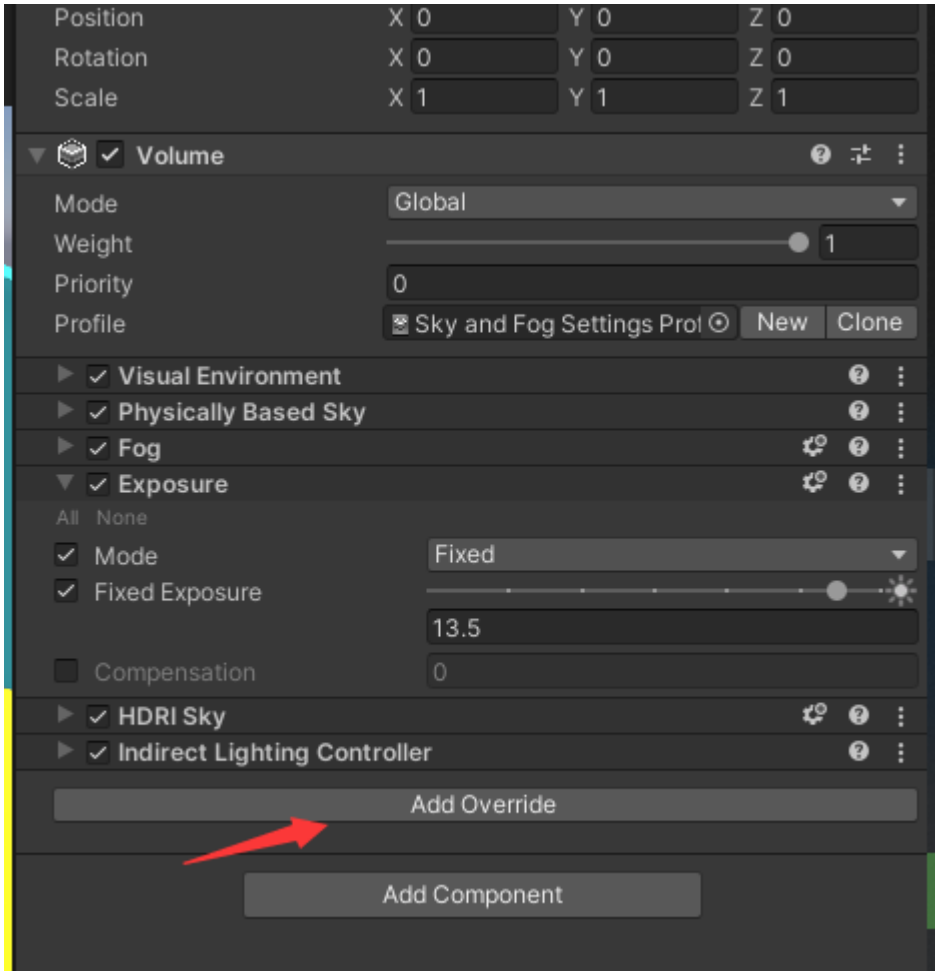
Quick Start:
Import the package of Clean Outline HDRP.



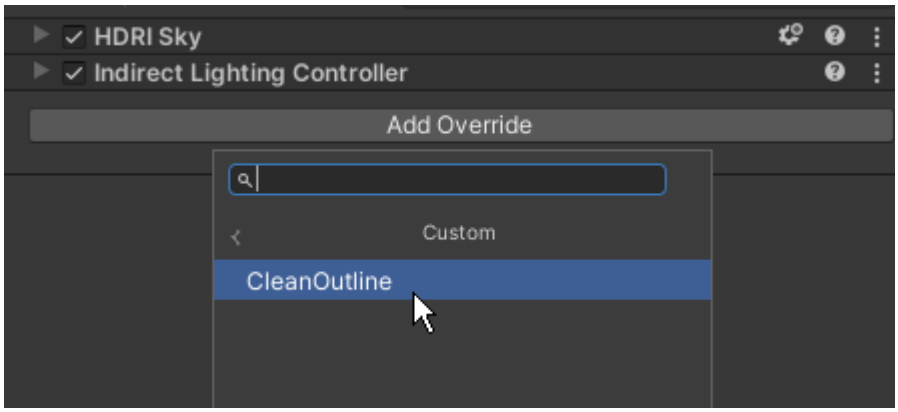
Open Project Setting panel on the menu “Edit->Project Settings”.
Select HDRP Default Setting tab, scroll to bottom and click + button under After Opaque And Sky



Select CR.CleanOutline to add.

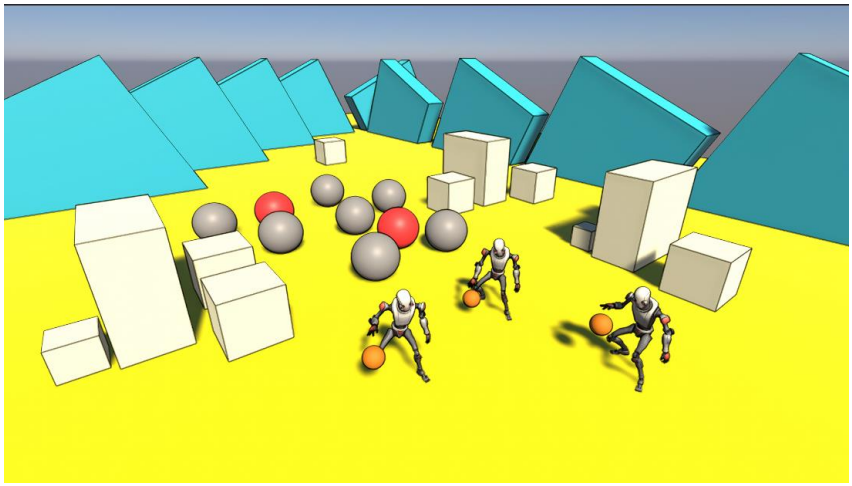


Now in the scene hierarchies, find the global Volume of the scene, click “Add Override”.

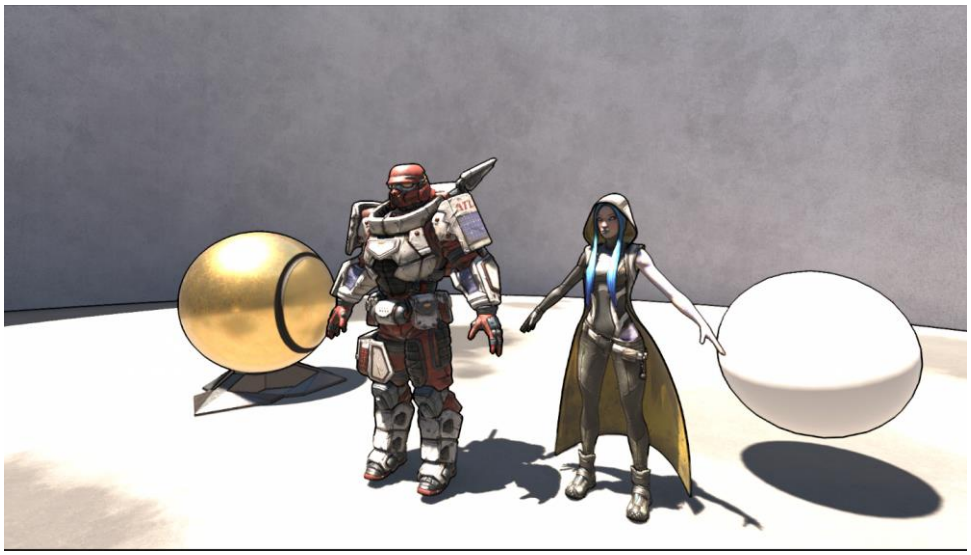


Then select Post-Processing->Custom->CleanOutline to add outline.

Set its Intensity as 1.



Then the scene should have a post outline effect;

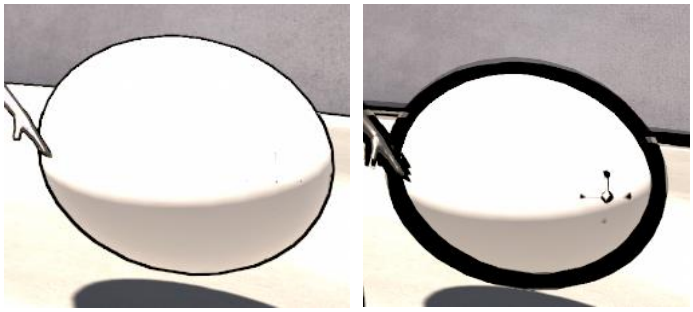


Result with AAA game characters(not included in the package).

Outline Parameters:

First off to tune the parameters, you should have a general concept of thickness and strength. When outline shader shades a pixel on the screen, it will sample the pixel data and some other pixels around it.

So thickness means how much distance it will reach other pixels from the original pixel.

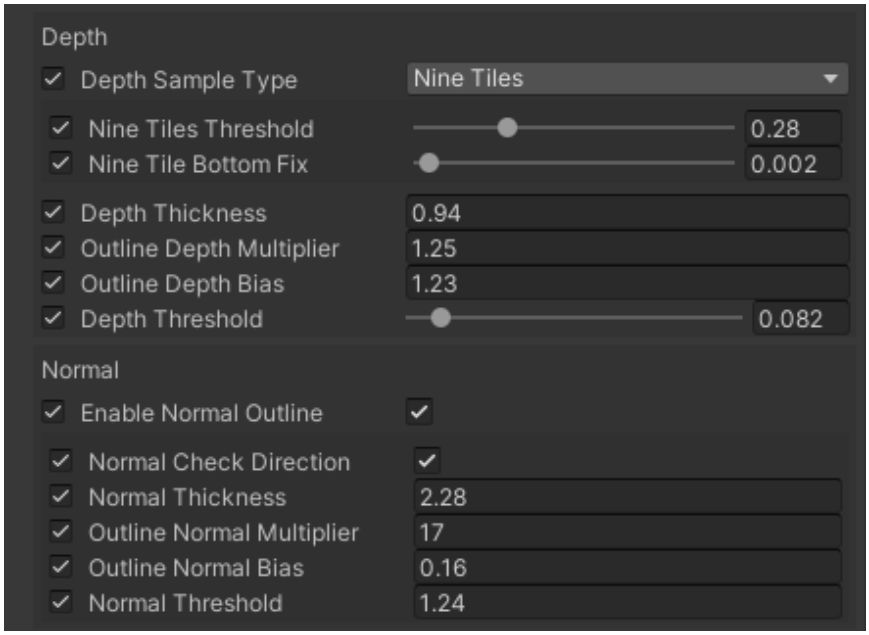


With larger thickness, outline shader will samples pixels more far away like the picture on the

right. So it doesn't mean that larger thickness is alwasy better, for the best value you should find a balance.
And strength means how much affect the outline result to apply to the final result.

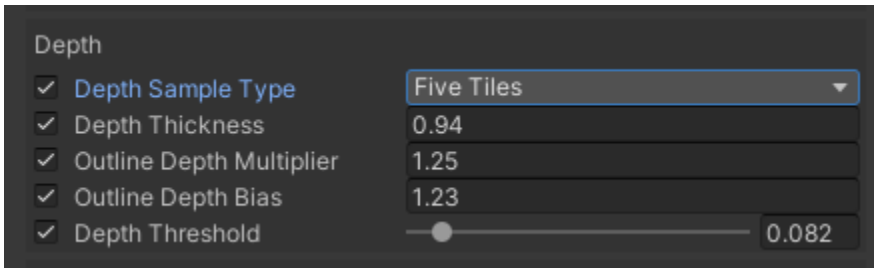
The compare of bigger and smaller strengths in the pictures. The outline becomes lighter with smaller strength.
Intensity controls a global strength of outlines, there are also other parameter that would affect strength separately auto calculated in the shader to the final result.

Intensity : Strength of the outlines.
Outline Color : The color of outlines.
Outline Thickness : The global thickness of outlines, affecting both depth and normal outlines.

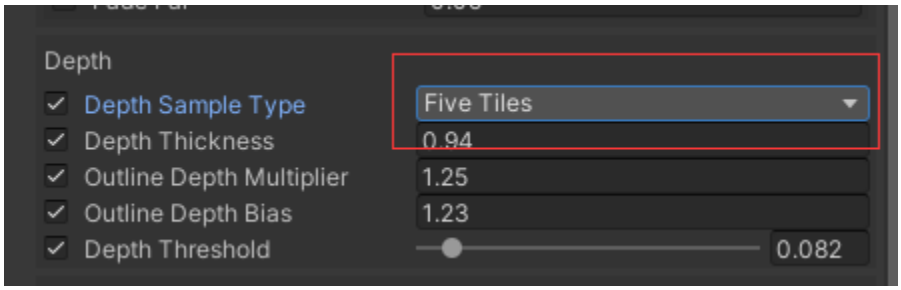


The outline effect include a depth calculation and a normal calculation.
Depth will always be calculated to create outline.
Normal can be optional because it only work good in lowpoly scenes, and for PBR objects with complicated normal maps, the normal outline will look dirty and messy.

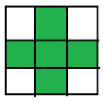
Depth



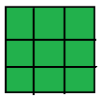
Depth Thickness : Thickness of the depth outline
Depth Multiplier/Bias : Affect the strength of depth outline.
Depth Threshold : After samples of depth difference, there will be a check if the result is under the threshold, and if it is it will be ignored. Depth will only consider the result over the threshold.



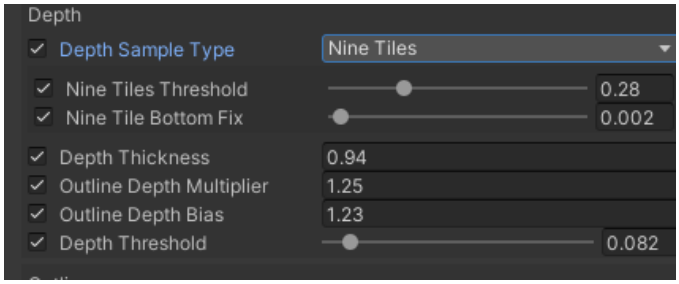
Depth calculation has 2 ways of sampling.



Five Tiles means it will sample the middle of the pixel with 4 other pixels by the side.



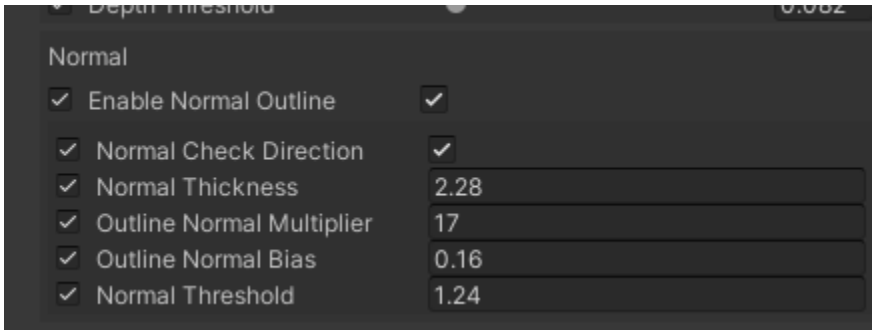
Nine Tiles will sample the middle and 8 other pixels around, and will be more costly.



And in Nine Tiles way, there will be another threshold for checking, and a fix value to fix a weird effect which can be left as it is (0.002).

Best practice is to try to tune the values to see how they take effect, or read the outline shader to see how they work.

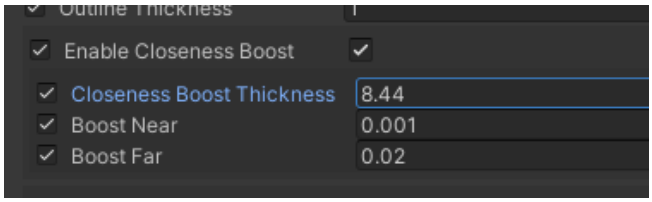
Normal



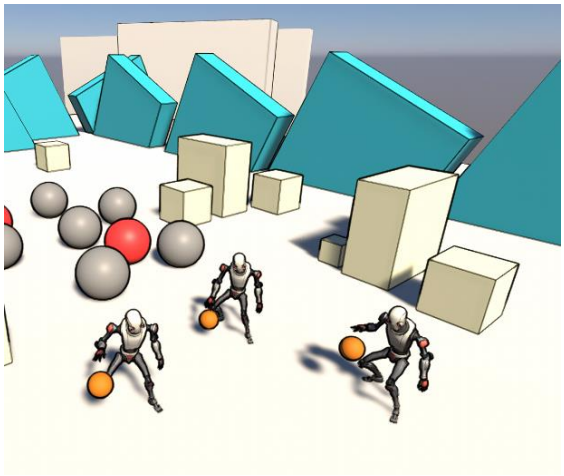
Enable Normal Outline : You can use normal outline as an option, it can work good with lowpoly scenes. But for scenes with PBR normal maps, normal outline might give weird result so you might need to turn if off.
Normal Check Direction: If not toggled, and normal values will just be compared as raw vector values. And if toggled, normal will be seen as direction vectors. Different samples of normal vectors will be calculated into an angle between them to calculate further result.
Normal Thickness : Thickness of normal outline.
Normal Multiplier/Bias : Affect the strength of the normal.
Normal Threshold : A threshold to ignore the result value lower than it for normal outline.

Normal outline will always sample 5 tiles of pixels.

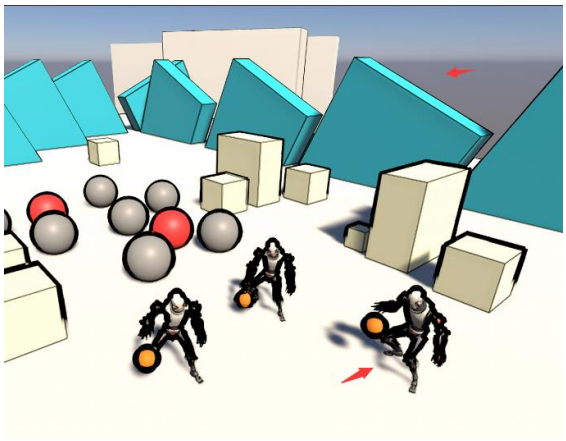
Some modifiers



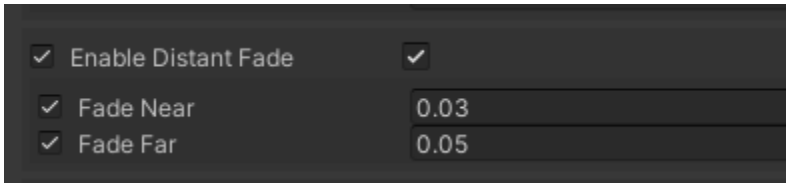
Closeness boost will add more thickness if the pixel is close enough. So in cases like that you want further distance has small thickness, and close distance has larger.
But please note that the thickness means how far distance it will sample near by pixels, and it cannot sample less than 1 pixels. So boost thickness might not often work good, you might turn it off if you can find a good result.



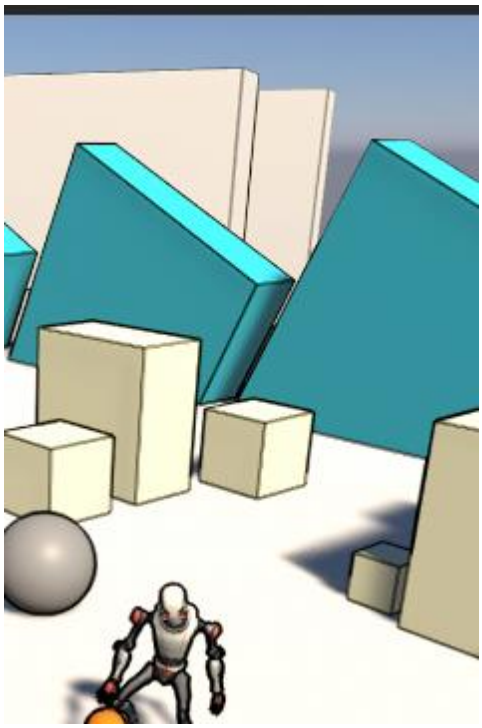
No Closeness Boost for thickness.



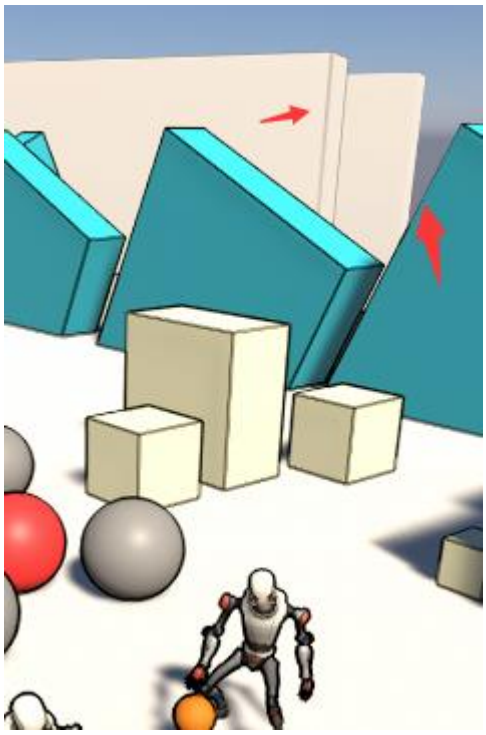
Turn on Closeness Boost, the thickness of close objects appear thicker.



Distance fade will consider that if the pixel's distance (depth) is too far, it's strength will be seen as 0. So that distanced objects will have no or less outline effect.

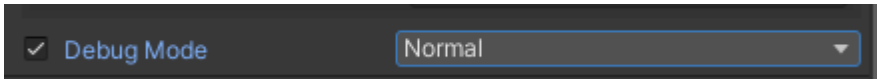


No Distance Fade outline.

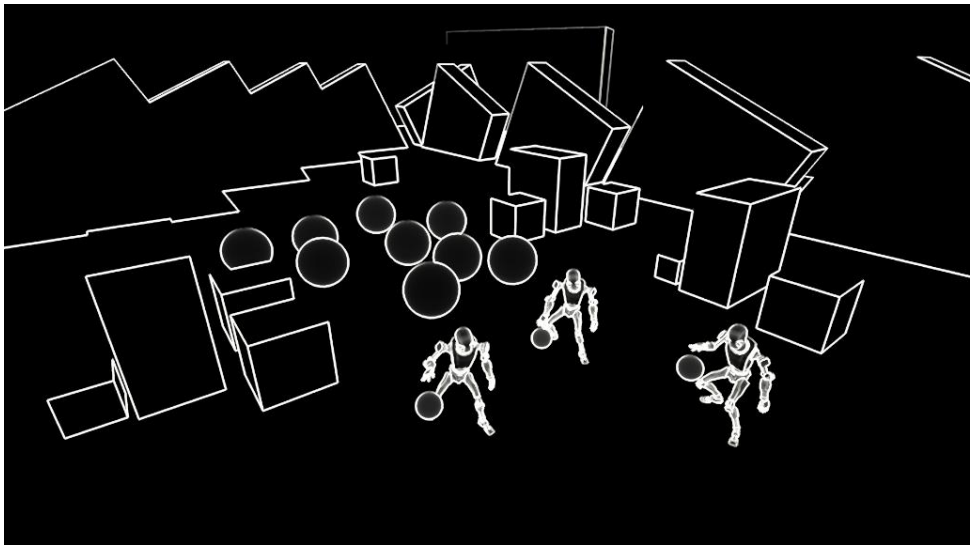


Turn on Distance Fade, the outline strength almost fade to 0 at a distance.

Debug Mode



Turn on debug mode to see how depth and normal outline work standalone in the game view.



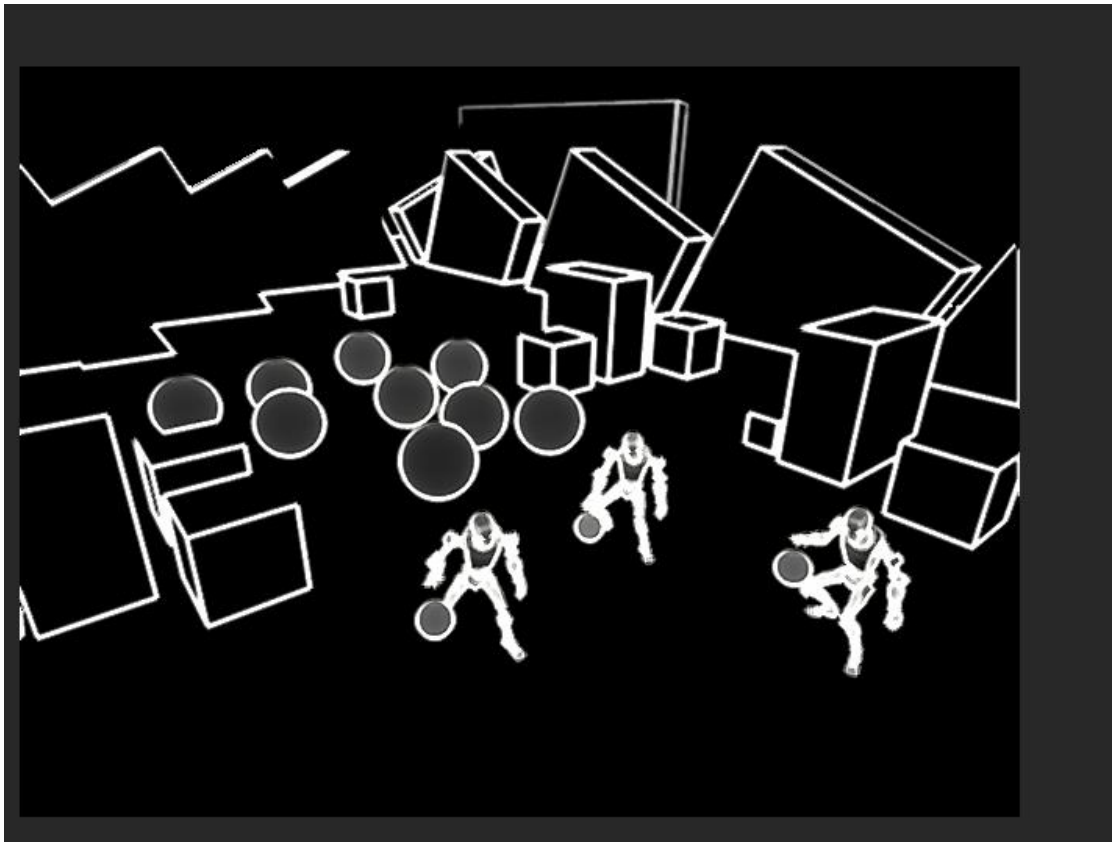
Known Issues:

Depth and normal outline requires object that will write to screenspace textures related with depth and normal when rendered. Which means that transparent object like glass, smoke will have no outline effect.

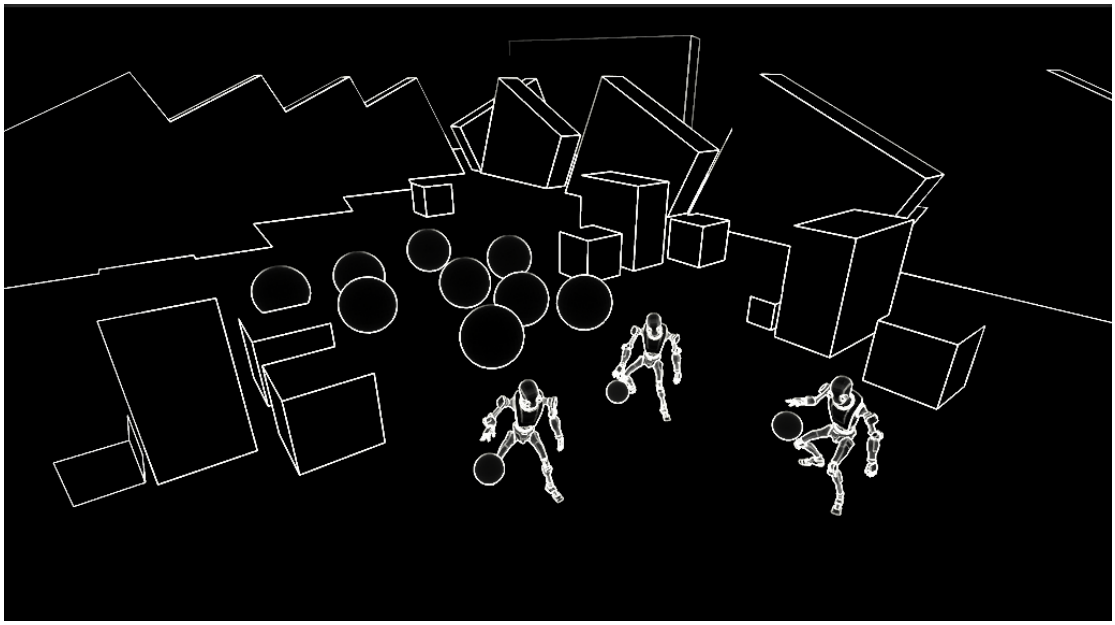
Fullscreen outline samples data from screenspace textures by each pixel and the pixels nearby it, and this would be limited by the resolution of the screen. It might give some saw teeth effect, which could be reduced by anti-aliasing and larger resolution, bigger thickness.

You might also find it that with smaller resolution, the outline will look very thick which might be inevitable as distance to sample around each pixel can't be lower than 1. While with larger resolution outline will look too thin.

So you might need to manually change the thickness by the resolution of the game.

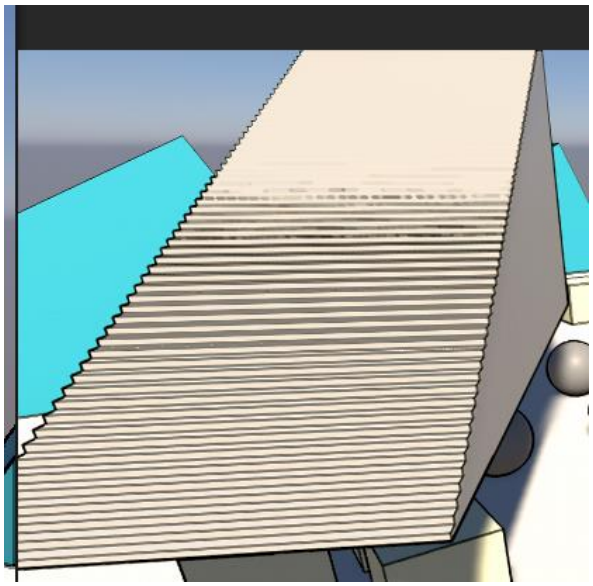


Outline in 640x480 resolution.

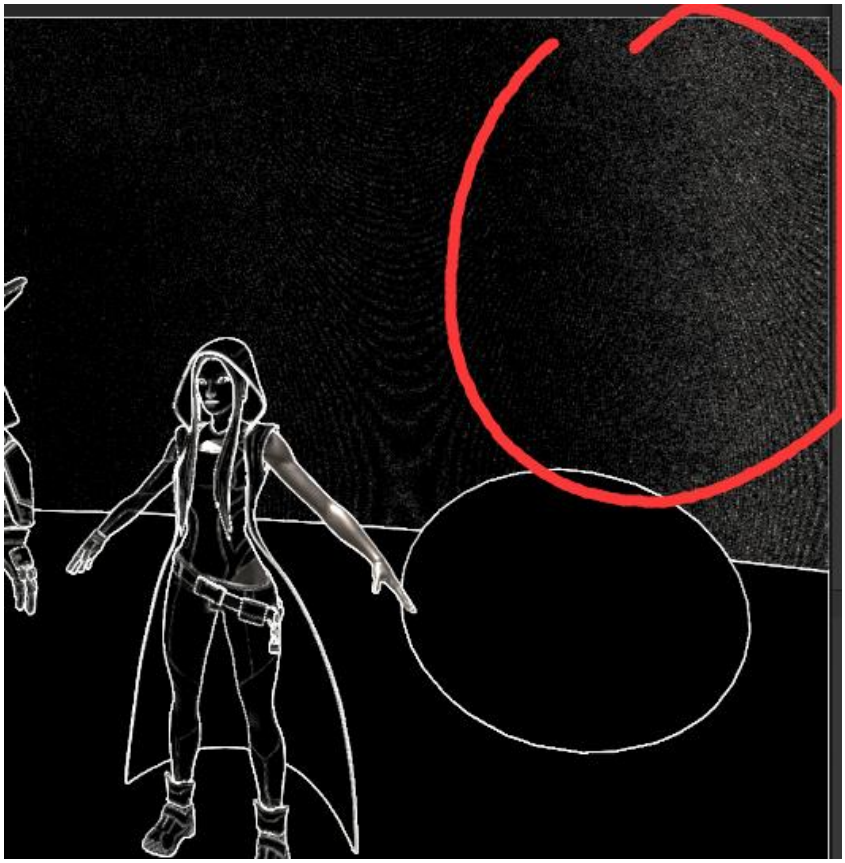


Same outline in 4k resolution.

Sometimes, clustered sudden changes of surfaces and distant objects might give a dense result of outlines.



For example the stairs with too many steps. So you might need to use less steps stair, or use LOD to reduce objects complicated surfaces at range if you want to use fullscreen outline.



Normal map of object's materials might give more normal outline effect on the surface of objects. So if you don't want this fuzzy effect just disable normal outline especially with PBR objects or tune outline normal parameters until they look okay.

Credits:

https://www.vertexfragment.com/ramblings/unity-postprocessing-sobel-outline/

https://github.com/Daodan317081/reshade-shaders

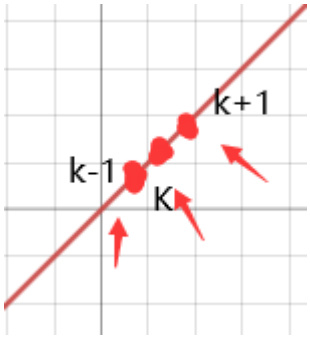
The shader got some ideas from Steven Sell and Alexander Federwisch with links above and thanks to them..

Please not that if you want to take a look at Steven Sell's sobel outline guide. He uses absolute value of difference to do the sample.

```
float4 SobelSample(Texture2D t, SamplerState s, float2 uv, float3 offset)
{
    float4 pixelCenter = t.Sample(s, uv);
    float4 pixelLeft   = t.Sample(s, uv - offset.xz);
    float4 pixelRight  = t.Sample(s, uv + offset.xz);
    float4 pixelUp     = t.Sample(s, uv + offset.zy);
    float4 pixelDown   = t.Sample(s, uv - offset.zy);

    return abs(pixelLeft  - pixelCenter) +
           abs(pixelRight - pixelCenter) +
           abs(pixelUp    - pixelCenter) +
           abs(pixelDown  - pixelCenter);
}
```

Which might cause problems as abs means that the a curve data of the pixels will be giving result out of nowhere from the middle.



For example in a curve ,you want to get the sampled delta value of each point.

D(k-) = f(k) – f(k-1), is the sampled value at point k to the left, let's say it's 0.1 in the graph.

D(k+) = f(k) – f(k + 1), is the sampled value at point k to the right, then it's -0.1 because the curve is linear. So D(k-) and D(k+) will cancel each out.

Then D(k) = D(k-) + D(k+) = 0.1 + (-0.1) = 0

Which is like a linear surface of depth should have no outline effect, only non-linear sudden changes of depth will cause outline.

But if you add "abs" to D(k-) and D(k+), the delta will be

D(k) = abs(D(k-)) + abs(D(k+)) = abs(0.1) + abs(-0.1) = 0.2

There's no point to add outline effect if the depth just changes linearly.

And the depth outline will be dirty when you look from a close angle by the surface in Steven's guide like in the picture above. Which I fixed in Clean Outline shader.

There will be no dirty effect for depth samples.

If you have any problems, you could send me email
ryanflees@hotmail.com
ryanbai2008@163.com