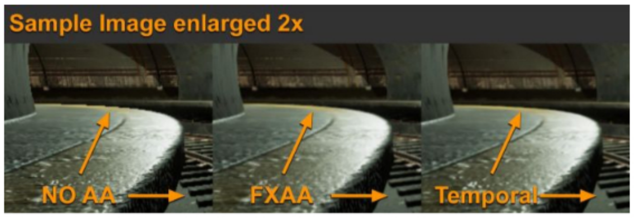
Understanding Post Process Features

Checked with version: 2017.3 - Difficulty: Intermediate

As you might expect given the name, Post Processes are rendering effects that are based on an existing rendered Scene. Effects in Post Process are usually Scene view dependant or layered on top of the rendered Scene before generating the final render. The clear advantage of this feature is the instant visual feedback and dramatic improvement to the Scene without the need to alter existing content. There are lots of features in Post Process that are not considered a baseline requirement for creating believable Scenes. However its capability to enhance a Scene further is certainly worth the time it takes to understand the system. The goal here is to give you the information needed to decide which Post Process effects are right for your situation and to avoid the pitfalls that can come with these advanced features. More info can be found here: https://docs.unity3d.com/Manual/PostProcessingOverview.html

- **Anti aliasing.** When rasterizing a 3D polygon into a 2D screen with limited resolution, the final pixels will show a stair-stepping effect (aliasing). There are different solutions for Anti-Aliasing techniques in real-time 3D (Supersampling, MSAA,FXAA, SMAA, Temporal, and any combination of those or newer methods). The most popular techniques currently are FXAA and Temporal Anti-Aliasing due to their effectiveness and relatively high performance.



The sample above showcases that FXAA does a good job of fixing some of the glaring aliasing effects. Temporal, however, takes it a step further and can be seen to perform a much better job in the tram rails.

FXAA is a pure post process anti-aliasing. In simple terms, the rasterized Scene is captured, edges are analyzed and an algorithm run on top of the existing image to smooth it out. It is straightforward, doesn't have any complex dependencies, and is fast.

Temporal Anti-Aliasing is a lot more complex. Temporal uses jittering and the previous frame as additional data to be blended in. It uses motion vectors to predict which pixels need to be rejected or accepted to render the final frame. The idea is to increase the effective resolution of a frame with more data without the need to render the Scene larger than its final resolution (Supersampling). The benefit is clearly a much smoother anti-aliasing, similar to the quality given by SuperSampling, without the major performance impact.

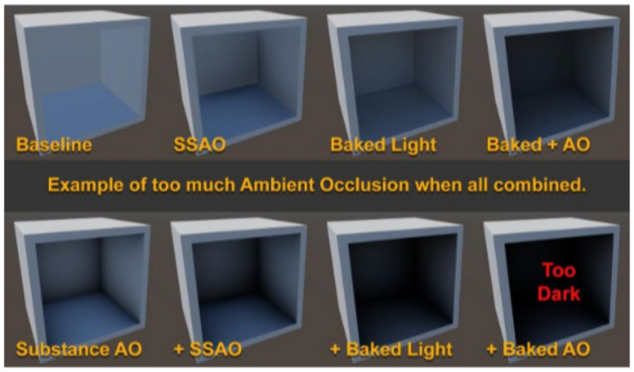
Like everything in real time rendering, there's always trade off. Temporal AA requires motion vectors to function and has a larger performance cost when compared to FXAA. Temporal's complex nature of predicting the final image can cause some artifacts for fast moving objects and texture blurines which might not be suitable for some applications.

- **Ambient Occlusion.** Ambient occlusion post process is an approximation of ambient occlusion based on screen space data, mainly depth, hence it is usually called Screen Space Ambient Occlusion (SSAO). As explained on the lighting setup section, SSAO can gives better fidelity when shading ambient lighting, especially for dynamic objects that typically don't have any occlusion interaction between static Scene and dynamic Scene.

Screen Space Ambient Occlusion - Debug View

While in general SSAO helps a Scene's ambient shading, it can cause too much occlusion. Having per object baked Ambient Occlusion from offline Digital Content Creation Software with additional ambient occlusion from light baking puts SSAO as the third layer of ambient occlusion.

Make sure to keep the final output in mind when setting up SSAO and try to find a balance with the other Ambient Occlusion solutions.



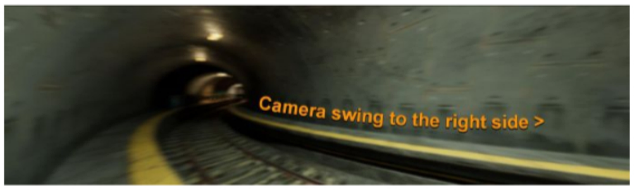
Creating Believable Visuals

Sample showcasing the issue of adding too much AO causing open area to be very dark.

- **Screen Space Reflection.** Similar to SSAO, Screen Space Reflection use the current Scene view to approximate reflections via ray tracing. To get a believable results, it is almost always a good idea to enable this feature. It adds a highly accurate reflection that compliments the normal cubemap captured reflection. However, enabling this feature does restrict rendering to deferred rendering only and has a performance impact. Another downside of SSR is "screen space", anything not on the screen will not generate reflection hits and can cause a missing reflection such as the sweeping effect at the edges of the screen.
- **Depth of Field.** What most DCC understood from Depth of field effect is actually the lack of DoF. Blurry foreground and background images focused only at small point in space is what usually comes to mind when talking about DoF. While this effect can give the cinematic feel of a large sensor camera, it can also be used to change the scale perception of a Scene like how a tilt-shift camera lens gives a miniature effect.

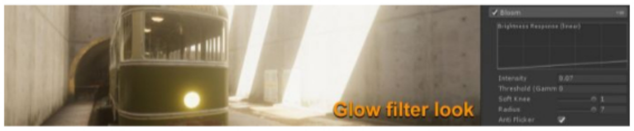
The above example is a real-life photograph looking like a miniature by faking DOF.

- **Motion blur** Some argue that motion blur induces motion sickness or causes a distraction while others swears by it. Should you enable it? The answer depends on the desired effect and application. A subtle motion blur goes a long way in blending the transition of one frame to another. This is especially true, if there is a massive difference in Scene translation typically found in first person or 3rd person cameras. For example, a wide angle view where the player can swing their camera really fast from left to right will look jittery and gives a stop motion look without motion blur even if it is rendering at 60FPS.

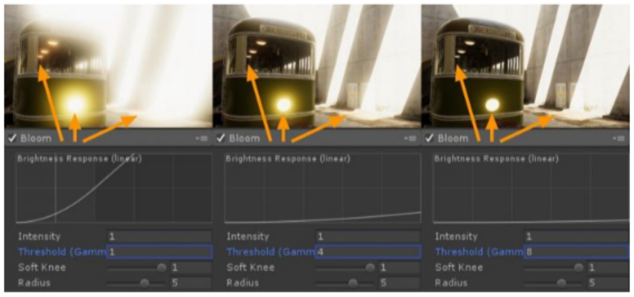


The sample above is running motion blur at a Shutter angle of 180 degrees. (Full 360 shutter angle will give you a full frame duration trail, while anything less means less trail). With that in mind, if you are aiming for a stop motion look, then by all means disable motion blur.

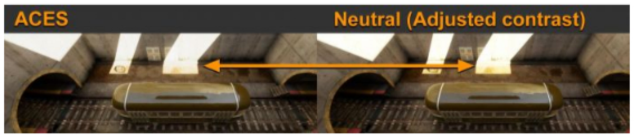
- **Bloom and emissive** Bloom in real-life is an artifact of a lens where light beams aren't focused properly, usually found on lower quality camera lenses or some special effects glow camera filter.



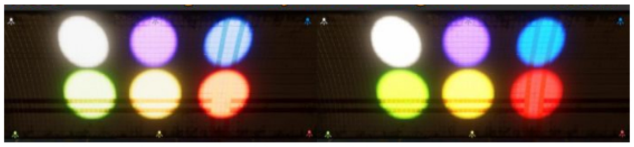
```
Intensity          0.07
Threshold (Gamm    0
Soft Knee          1
Radius             7
Anti Flicker       ✓
```
Two main things that are usually expected from using a bloom are a hazy soft image (as seen above, 0 threshold setup) or used to differentiate elements of high intensity or bright light (image below).


✓ Bloom
Brightness Response (linear)
✓ Bloom
Brightness Response (linear)
✓ Bloom
Brightness Response (linear)
```
Intensity          1      Intensity          1      Intensity          1
Threshold (Gamm    1      Threshold (Gamm    4      Threshold (Gamm    8
Soft Knee          1      Soft Knee          1      Soft Knee          1
Radius             5      Radius             5      Radius             5
```
Overusing this features might backfire as seen at the sample where there are lots of high intensity pixels while the threshold of the intensity started to bloom very early (Threshold Gamma 1.0 means anything above value of 1.0 in current exposure will bloom intensely) Selecting the value of the threshold depends on the specific values of your emissive surfaces, the lighting setup for the Scene, and whether you have enabled eye adaptation.

- **ToneMapper type** A tonemapper is a way of processing a linear HDR buffer of input data and rendering it back out to the designated colour space for final output. This is similar to how a camera works. In Unity Post processing there are two types of tonemapper, Neutral and ACES (Academy colour Encoding System). More info for ACES can be found in Wiki. At first glance the difference between the two is in its default contrast of the tonemapper. However that isn't the main difference between the two as you can adjust neutral to be similar in contrast to ACES (seen below that the two image is almost identical).



Above Neutral settings: Black In 0.02, White In 10, Black Out -0.04, White Out 10, White Level 5.3 and White Clip 10. The main difference needed to be taken into account is how the two tone mappers handle high intensity colour values such as coloured light or emissive effects (e.g. explosion effects or fire).



The above image showcase how ACES tonemapper normalizes high intensity colour differently than the Neutral tonemapper.

- **Chromatic Aberration, Grain and Vignette** These are a post process effect to simulate artifacts from real life camera systems. To prevent these being abused, it is a good idea to understand how each of these occurred in real cameras.
  - Chromatic Aberration or CA effect is a dispersion of colour that appears on an image if the lens of a camera fails to focus all colour to the same convergent point. This is usually found in a poorly calibrated lens or lower quality lens. While this can sometimes add a sense of realism to a digital Scene, this can also mean your virtual camera is directed to convey a low quality lens.
  - Grain seen in final image of a real photograph or cinema are usually the telltale sign of an insufficient quantity of useful light entering the sensor, such as dark Scene or a high iso camera sensor/film translating to noise. This effect can be used to simulate camera limitations added to a pure clean 3D rendered Scene to feel more believable. However having too much noise in a Scene can distract viewer with a false sense of motion and effect the contrast of the final rendered image.
  - Vignette effect, similar to the CA effect, is an artifact where a lens could not give consistent light coverage from the center to the edge of the sensor/film of a camera. This can be used to give some sense of focus for a central point of a Scene, but can also be abused and make a Scene look like it was processed by an amature post editor.

The key takeaway from these post process fundamentals are for a content creator to effectively use the features with a sense of purpose rather than a "make it advance rendering" checkboxes as each additional effects added to the Scene cost some performance.

Post Process missing in this explanation are Eye Adaptation, colour Grading (other than tonemapper) and User Lut (Lookup Table). These post process effects warrants a deeper explanation. General information of these effects can be found here:
https://docs.unity3d.com/Manual/PostProcessing-EyeAdaptation.html

https://docs.unity3d.com/Manual/PostProcessing-ColorGrading.html

https://docs.unity3d.com/Manual/PostProcessing-UserLut.html