Post Processing

There are three render pipelines that support post-processing within Unity, which are the Built-In Render Pipeline, Universal Render Pipeline (URP) & High-Definition Render Pipeline (HDRP). For the Built-In Render Pipeline, we must download the post-processing version 2 package to utilize the post-processing effects. On the other hand, both the URP & HDRP are already installed in Unity projects, depending on the template we are using. Below are references to the different post-processing effects that are available for each render pipeline.

Effects	URP	HDRP	PPv2
Ambient Occlusion	Yes	Yes	Yes
Anti-aliasing	Yes	Yes	Yes
Auto Exposure	No	Yes	Yes
Bloom	Yes	Yes	Yes
Channel Mixer	Yes	Yes	Yes
Chromatic Aberration	Yes	Yes	Yes
Color Adjustments	Yes	Yes	Yes
Color Curves	Yes	Yes	Yes
Fog	No	Yes	Yes
Depth of Field	Yes	Yes	Yes
Grain	Yes	Yes	Yes
Lens Distortion	Yes	Yes	Yes
Lift, Gamma, Gain	Yes	Yes	Yes
Motion Blur	Yes	Yes	Yes
Panini Projection	Yes	Yes	No
Screen Space Reflection	No	Yes	Yes
Shadows, Midtones, Highlights	Yes	Yes	No
Split Toning	Yes	Yes	No
Tone mapping	Yes	Yes	Yes
Vignette	Yes	Yes	Yes
White Balance	Yes	Yes	Yes

From the information above, I personally feel that the HDRP would be the best render pipeline for what we want to do, since:

1. HDRP is included within Unity, without having to download another package.
2. HDRP has the most options compared to the other render pipelines.

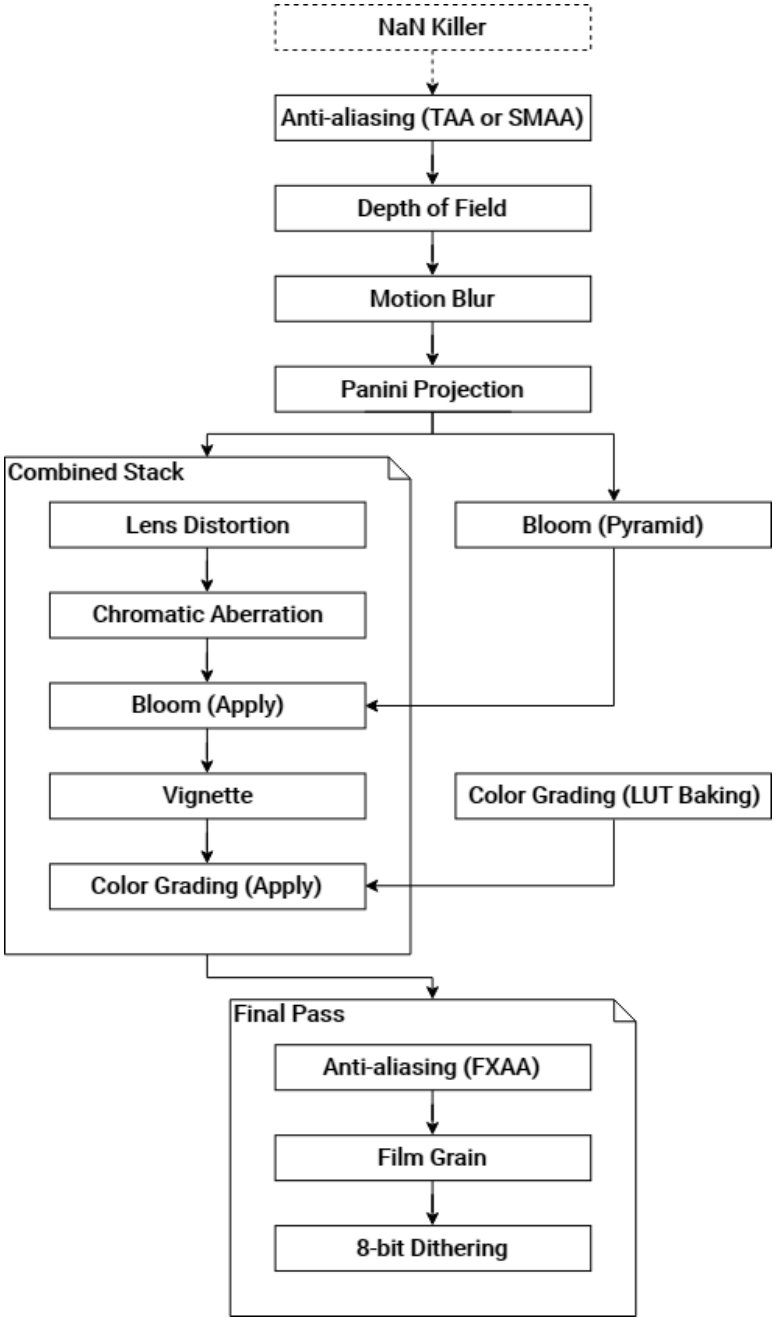
The effects we most likely are going to use are:

- **Color Curves** to adjust scene saturation to differentiate muted and restored states
  - https://docs.unity3d.com/Packages/com.unity.render-pipelines.high-definition@15.0/manual/Post-Processing-Color-Curves.html
- **Fog** for levels like the Hub (which will primarily be in the sky) and/or Bargaining (that can provide an atmospheric, aged aesthetic)
- **Screen Space Reflection** for the Depression level to create wet floor surfaces

There are, of course, more effects that we will use, but these are the ones that stood out to me.

https://docs.unity3d.com/Manual/PostProcessingOverview.html

Below is an image of the HDRP post-processing execution order. This effect grouping is later combined into the same compute shader to help minimize the number of passes.



https://docs.unity3d.com/Packages/com.unity.render-pipelines.high-definition@15.0/manual/Post-Processing-Main.html

Other References:

- Post-Process Layers & Volumes: https://docs.unity3d.com/Packages/com.unity.postprocessing@3.0/manual/Quick-start.html
- Screen Space Reflection: https://youtu.be/a0OQvWAPeuo?t=297
- PPv2 Process Analysis: https://medium.com/codex/how-to-use-post-processing-in-unity-part-i-3744c650d945

Cell Shading

Cel/Toon Shading, based on information from the Gane Developer website, "is a rending technique used to simulate the lighting in cartoons and comics". The way cel shading differentiates from traditional shading is by taking the natural gradient from a lit object and cutting it using either "no interpolations or interpolated values". In other words, it just makes a cool effect. The article later goes into the different rendering processes like Forward Rendering, which calculates every pixel within the scene to see which areas will be lit and which ones will not (very time-consuming and complex). The other process is a light pre-pass, which is more efficient as it takes every light within the scene and stores them. However, the downfall of this process is needing more complex hardware and its inability to work with custom lighting models. On the bright side, with Unity 5 we can now use deferred rendering, which stores all the scene geometry along lighting to calculate the final shading. This article focuses more on the user creating their own toon shader with camera image effects. However, we will approach this topic differently with a possible cel shader package.

https://www.gamedeveloper.com/programming/next-gen-cel-shading-in-unity-5

Other References:

- Flexible Cel Shader **FREE** Asset Package: https://assetstore.unity.com/packages/vfx/shaders/flexible-cel-shader-built-in-pipeline-112979
- Toon Shader Scripting: https://roystan.net/articles/toon-shader/#:~:text=Toon%20shading%20%28often%20called%20cel%20shading%29%20is%20a,how%20to%20write%20a%20toon%20shader%20in%20Unity.