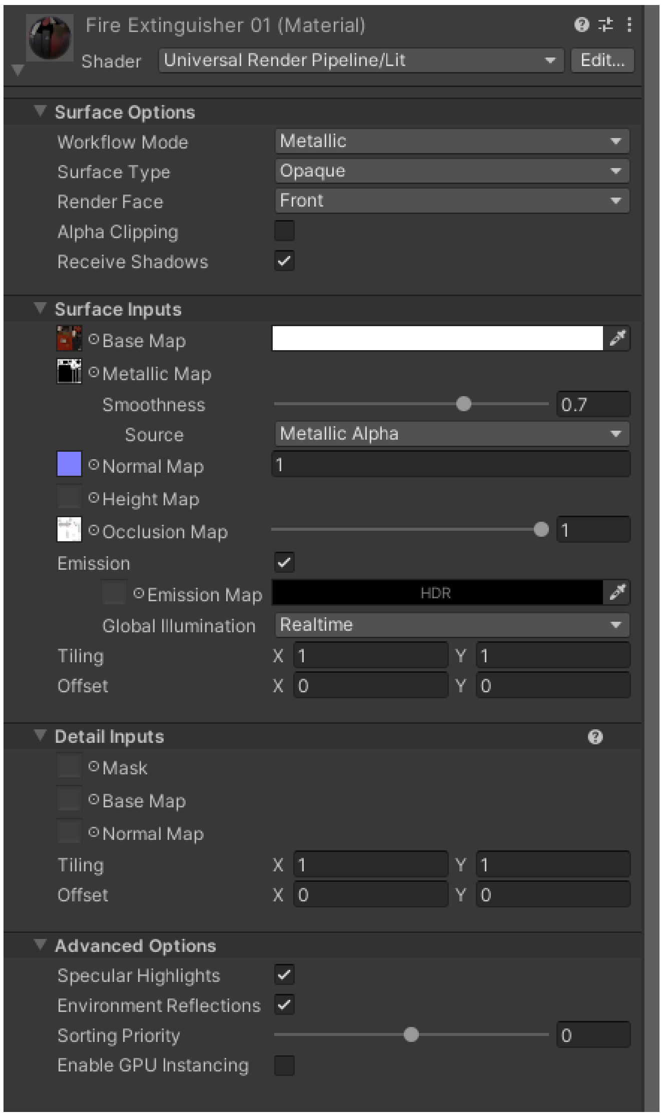
CA1 – Rendering Foundations

Brona Keevers-Roux

Github: https://github.com/hillyleopard133/4th-year-game-dev/tree/main

Video: https://youtu.be/6Xp2JY867Sw

PBR textures



Fire Extinguisher 01 (Material)
Shader: Universal Render Pipeline/Lit

Surface Options

Workflow Mode	Metallic
Surface Type	Opaque
Render Face	Front
Alpha Clipping	☐
Receive Shadows	✓

Surface Inputs

Base Map	
Metallic Map	
Smoothness	0.7
Source	Metallic Alpha
Normal Map	1
Height Map	
Occlusion Map	1
Emission	✓
Emission Map	HDR
Global Illumination	Realtime
Tiling	X 1 Y 1
Offset	X 0 Y 0

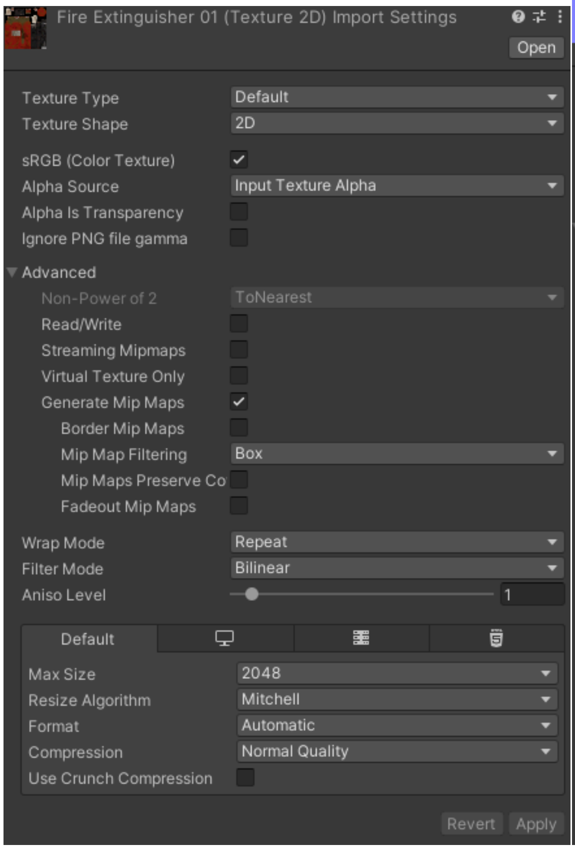
Detail Inputs

Mask	
Base Map	
Normal Map	
Tiling	X 1 Y 1
Offset	X 0 Y 0

Advanced Options

Specular Highlights	✓
Environment Reflections	✓
Sorting Priority	0
Enable GPU Instancing	☐

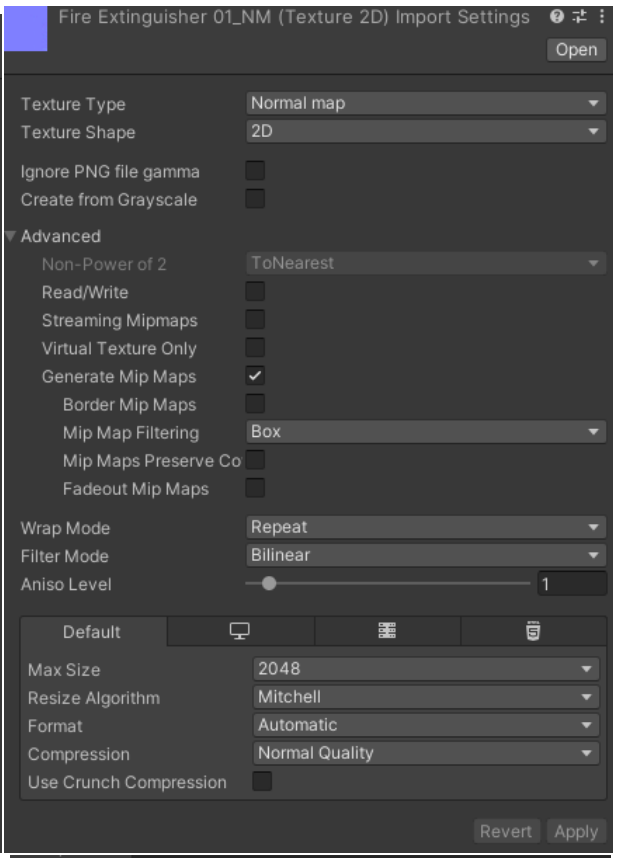
In physically based rendering, Base Colour textures use sRGB because they represent visible colour and need to be converted properly for accurate lighting. This ensures the material looks correct under different light conditions. Data textures such as Normal, Metallic, Roughness, and Ambient Occlusion are different because they store surface information rather than colour. These must have sRGB disabled so Unity reads their values exactly as they are, without changing them. Normal maps also use the Normal Map texture type so Unity interprets them correctly as surface direction data. This ensures the lighting behaves correctly and the material appears physically accurate.


Fire Extinguisher 01 (Texture 2D) Import Settings

Open

Texture Type	Default
Texture Shape	2D
sRGB (Color Texture)	✓
Alpha Source	Input Texture Alpha
Alpha Is Transparency	☐
Ignore PNG file gamma	☐
Advanced	
Non-Power of 2	ToNearest
Read/Write	☐
Streaming Mipmaps	☐
Virtual Texture Only	☐
Generate Mip Maps	✓
Border Mip Maps	☐
Mip Map Filtering	Box
Mip Maps Preserve Co	☐
Fadeout Mip Maps	☐
Wrap Mode	Repeat
Filter Mode	Bilinear
Aniso Level	1
Default	
Max Size	2048
Resize Algorithm	Mitchell
Format	Automatic
Compression	Normal Quality
Use Crunch Compression	☐

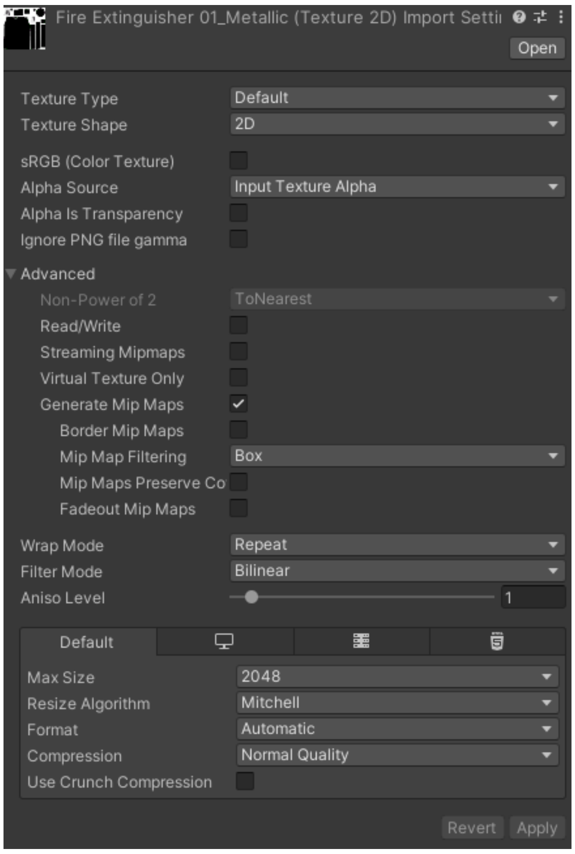
Revert Apply


Fire Extinguisher 01_NM (Texture 2D) Import Settings

Open

Texture Type	Normal map
Texture Shape	2D
Ignore PNG file gamma	☐
Create from Grayscale	☐
Advanced	
Non-Power of 2	ToNearest
Read/Write	☐
Streaming Mipmaps	☐
Virtual Texture Only	☐
Generate Mip Maps	✓
Border Mip Maps	☐
Mip Map Filtering	Box
Mip Maps Preserve Co	☐
Fadeout Mip Maps	☐
Wrap Mode	Repeat
Filter Mode	Bilinear
Aniso Level	1
Default	
Max Size	2048
Resize Algorithm	Mitchell
Format	Automatic
Compression	Normal Quality
Use Crunch Compression	☐

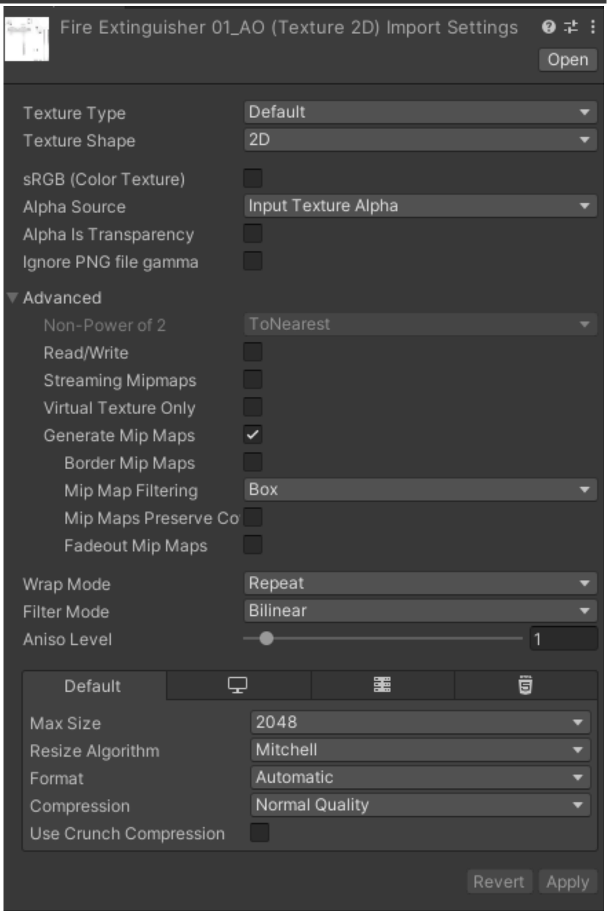
Revert Apply


Fire Extinguisher 01_Metallic (Texture 2D) Import Setti

Open

Texture Type	Default
Texture Shape	2D
sRGB (Color Texture)	☐
Alpha Source	Input Texture Alpha
Alpha Is Transparency	☐
Ignore PNG file gamma	☐
Advanced	
Non-Power of 2	ToNearest
Read/Write	☐
Streaming Mipmaps	☐
Virtual Texture Only	☐
Generate Mip Maps	✓
Border Mip Maps	☐
Mip Map Filtering	Box
Mip Maps Preserve Co	☐
Fadeout Mip Maps	☐
Wrap Mode	Repeat
Filter Mode	Bilinear
Aniso Level	1
Default	
Max Size	2048
Resize Algorithm	Mitchell
Format	Automatic
Compression	Normal Quality
Use Crunch Compression	☐

Revert Apply


Fire Extinguisher 01_AO (Texture 2D) Import Settings

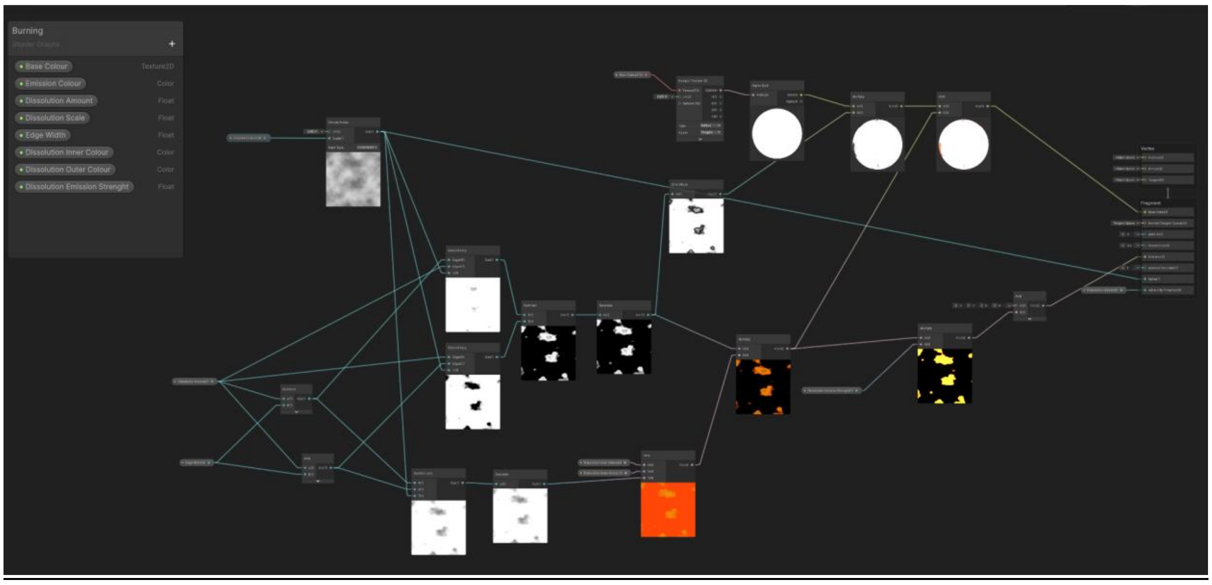
Open

Texture Type	Default
Texture Shape	2D
sRGB (Color Texture)	☐
Alpha Source	Input Texture Alpha
Alpha Is Transparency	☐
Ignore PNG file gamma	☐
Advanced	
Non-Power of 2	ToNearest
Read/Write	☐
Streaming Mipmaps	☐
Virtual Texture Only	☐
Generate Mip Maps	✓
Border Mip Maps	☐
Mip Map Filtering	Box
Mip Maps Preserve Co	☐
Fadeout Mip Maps	☐
Wrap Mode	Repeat
Filter Mode	Bilinear
Aniso Level	1
Default	
Max Size	2048
Resize Algorithm	Mitchell
Format	Automatic
Compression	Normal Quality
Use Crunch Compression	☐

Revert Apply

<u>Shader Graph</u>

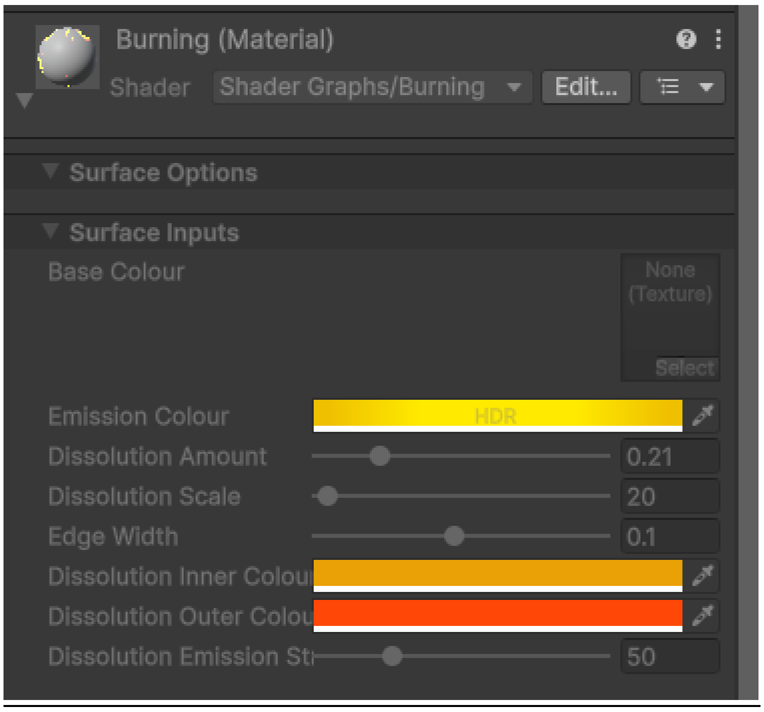
The burn band is made by using two Smoothstep nodes with slightly different thresholds based on the dissolve amount and edge width. Subtracting one from the other creates a thin band where the dissolve is happening. This band is used as a mask for the emission, which makes the glowing burn edge. The Dissolve Amount controls how far the dissolve has progressed, and the Edge Width controls how thick the burn band is. The Noise Scale changes the size and shape of the dissolve pattern, and the Emission Colour and Intensity control how the burn edge looks and how bright it is.



Burning

Shader Graphs

Base Colour	Texture2D
Emission Colour	Color
Dissolution Amount	Float
Dissolution Scale	Float
Edge Width	Float
Dissolution Inner Colour	Color
Dissolution Outer Colour	Color
Dissolution Emission Strenght	Float



Burning (Material)

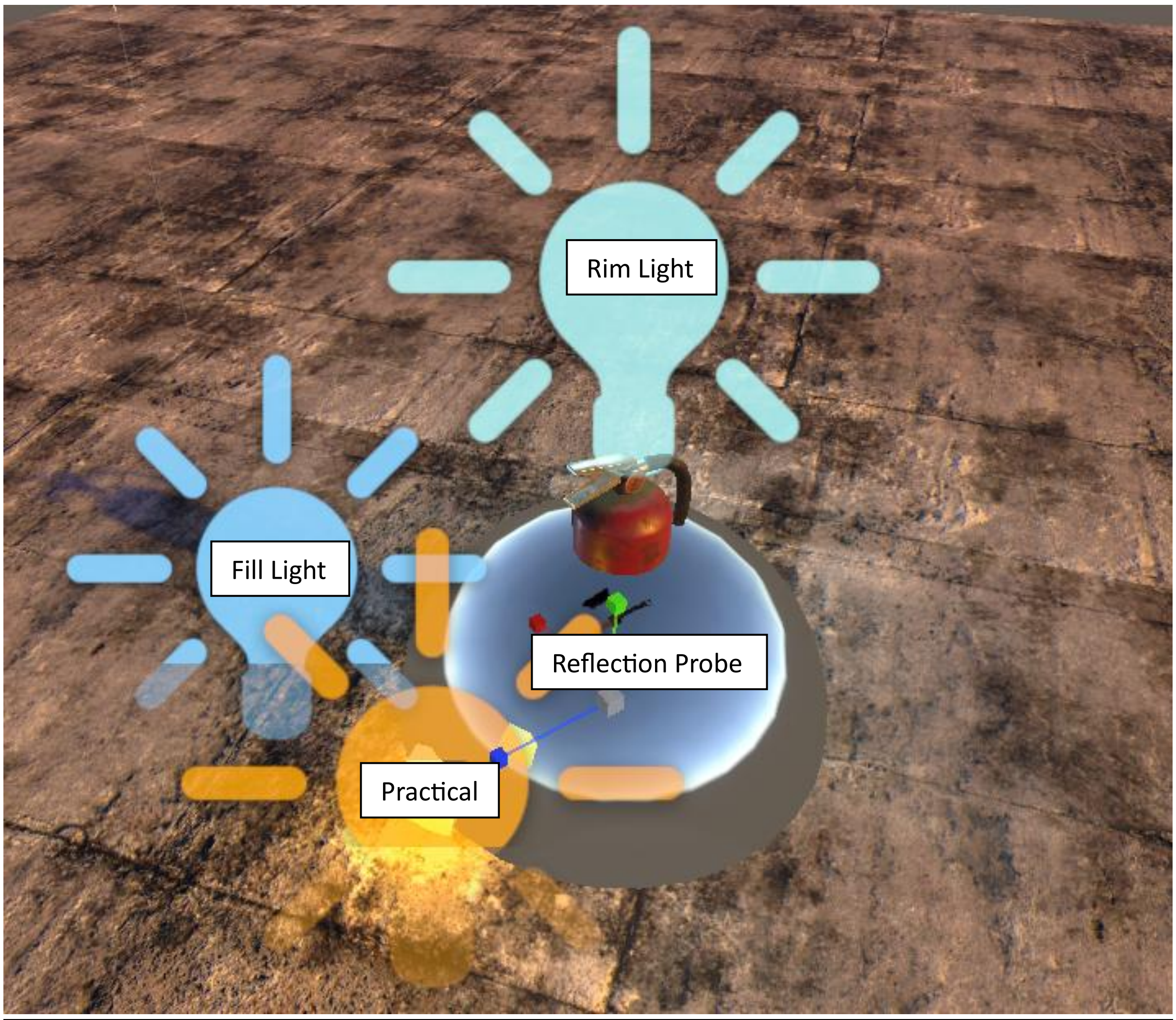
Shader Shader Graphs/Burning Edit...

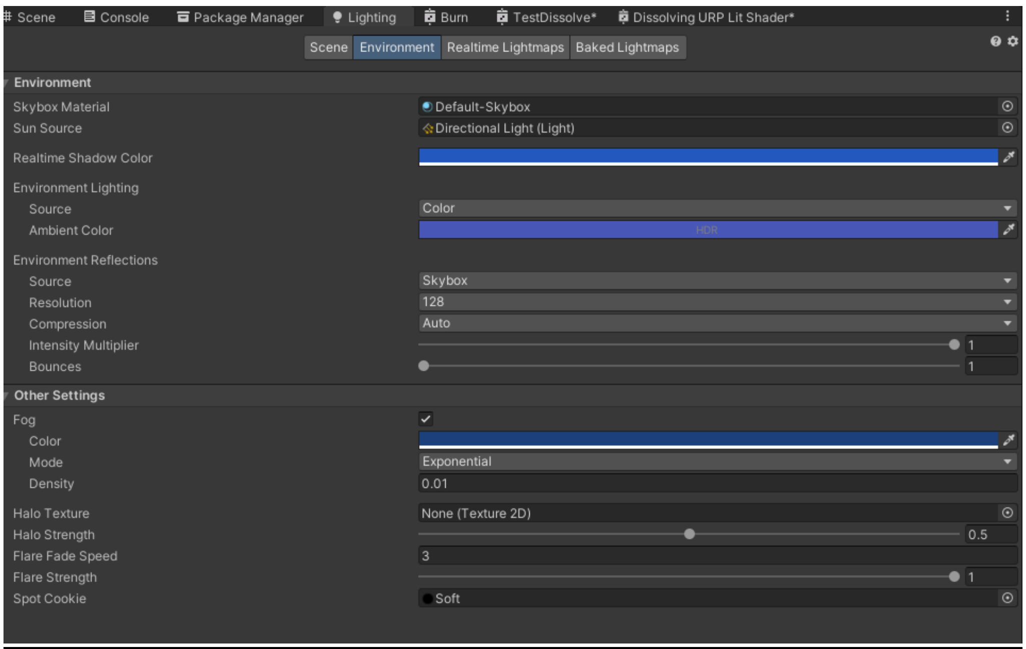
Surface Options

Surface Inputs

Base Colour	None (Texture) Select
Emission Colour	HDR
Dissolution Amount	0.21
Dissolution Scale	20
Edge Width	0.1
Dissolution Inner Colou	🟧
Dissolution Outer Colou	🟥
Dissolution Emission St	50

Scene Lighting





Environment

Skybox Material	Default-Skybox
Sun Source	Directional Light (Light)
Realtime Shadow Color	
Environment Lighting	
Source	Color
Ambient Color	HDR
Environment Reflections	
Source	Skybox
Resolution	128
Compression	Auto
Intensity Multiplier	1
Bounces	1

Other Settings

Fog	✓
Color	
Mode	Exponential
Density	0.01
Halo Texture	None (Texture 2D)
Halo Strength	0.5
Flare Fade Speed	3
Flare Strength	1
Spot Cookie	Soft

<u>Directional Light</u>

The directional light represents the low sun at dusk and acts as the primary light source. It uses a warm colour and low angle to simulate sunset lighting and create long shadows. This establishes the overall time of day and provides the main directional shading for the scene.

<u>Fill Light - Shadow Side</u>

The fill light provides soft illumination on the shadowed side of the hero asset to improve visibility and prevent the object from becoming too dark. A cool colour and low intensity were used to maintain the dusk atmosphere while improving readability. The light is limited to the hero asset using layer culling to avoid flattening the environment lighting.

<u>Rim Light</u>

The rim light is placed behind the hero asset to create a highlight along its edges, improving silhouette separation from the background. This helps the asset stand out clearly, especially in low-light dusk conditions. A slightly cool colour was used to match the ambient sky lighting.

<u>Practical Light - cube</u>

An emissive cube was used to represent a practical light source within the scene. The cube uses an emissive material to simulate a visible glowing object, and a point light was placed at the same position to provide actual illumination. This creates a motivated light source that improves realism and helps illuminate the hero asset and surrounding area. A warm colour was chosen to contrast with the cool ambient dusk lighting and reinforce the scene's atmosphere.

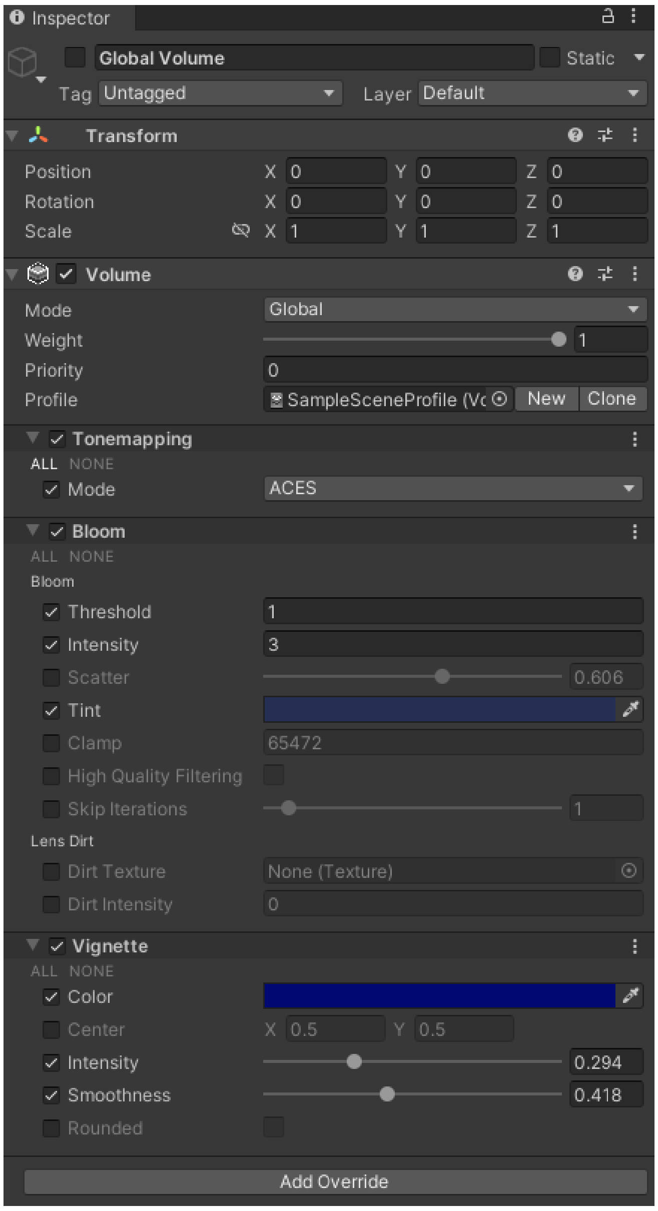
<u>Reflection Probe</u>

The reflection probe captures the surrounding environment and provides accurate reflection data for physically based materials. This improves material realism, particularly for reflective surfaces. A baked reflection probe was used to reduce runtime performance cost.

<u>Environment / Ambient Lighting</u>

The environment lighting was adjusted to create an open dusk ("blue hour") atmosphere. The ambient light colour and realtime shadow colour were set to a soft blue to simulate the cool fill of indirect light at dusk. A subtle blue fog was enabled to enhance depth and mood in the scene. These settings complement the warm directional sunlight and helper lights, maintaining readability while reinforcing the overall dusk lighting style.

Post Processing



Inspector

Global Volume — Static

Tag Untagged — Layer Default

Transform

Position	X 0	Y 0	Z 0
Rotation	X 0	Y 0	Z 0
Scale	X 1	Y 1	Z 1

✓ Volume

Mode	Global
Weight	1
Priority	0
Profile	SampleSceneProfile (Vc — New — Clone

✓ Tonemapping

ALL NONE

✓ Mode	ACES

✓ Bloom

ALL NONE

Bloom

✓ Threshold	1
✓ Intensity	3
Scatter	0.606
✓ Tint	
Clamp	65472
High Quality Filtering	
Skip Iterations	1

Lens Dirt

Dirt Texture	None (Texture)
Dirt Intensity	0

✓ Vignette

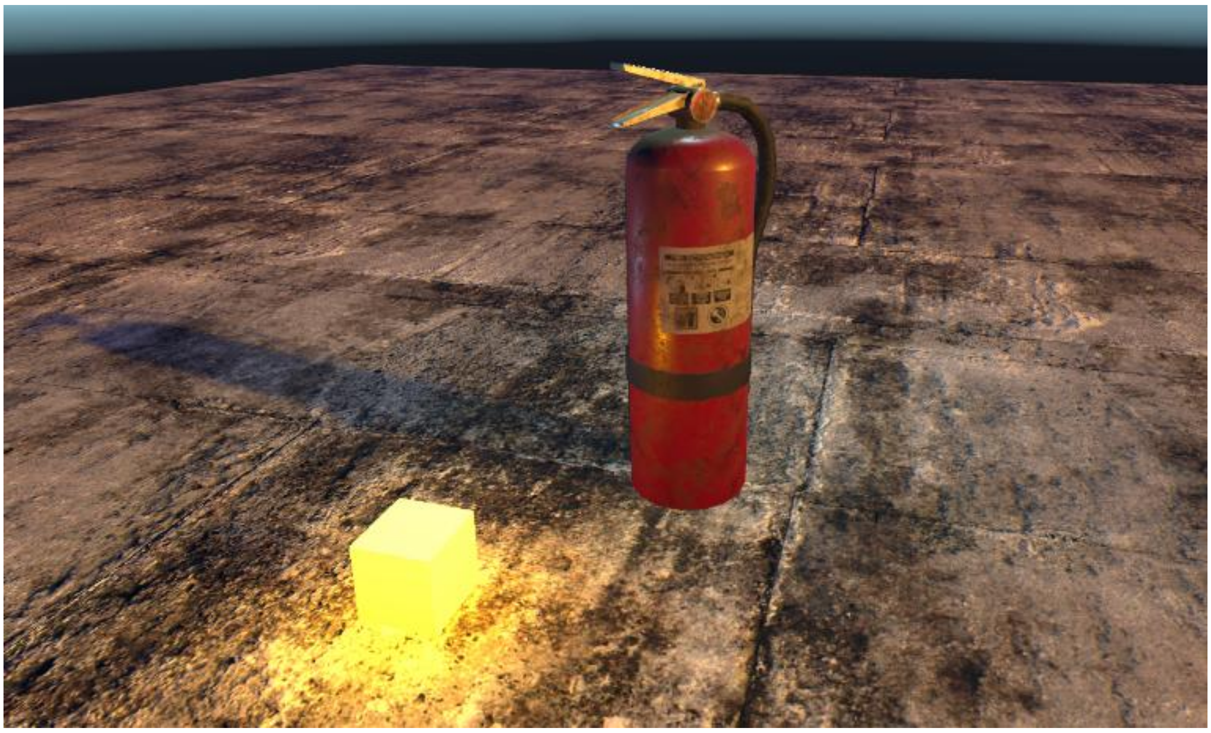
ALL NONE

✓ Color	
Center	X 0.5 Y 0.5
✓ Intensity	0.294
✓ Smoothness	0.418
Rounded	

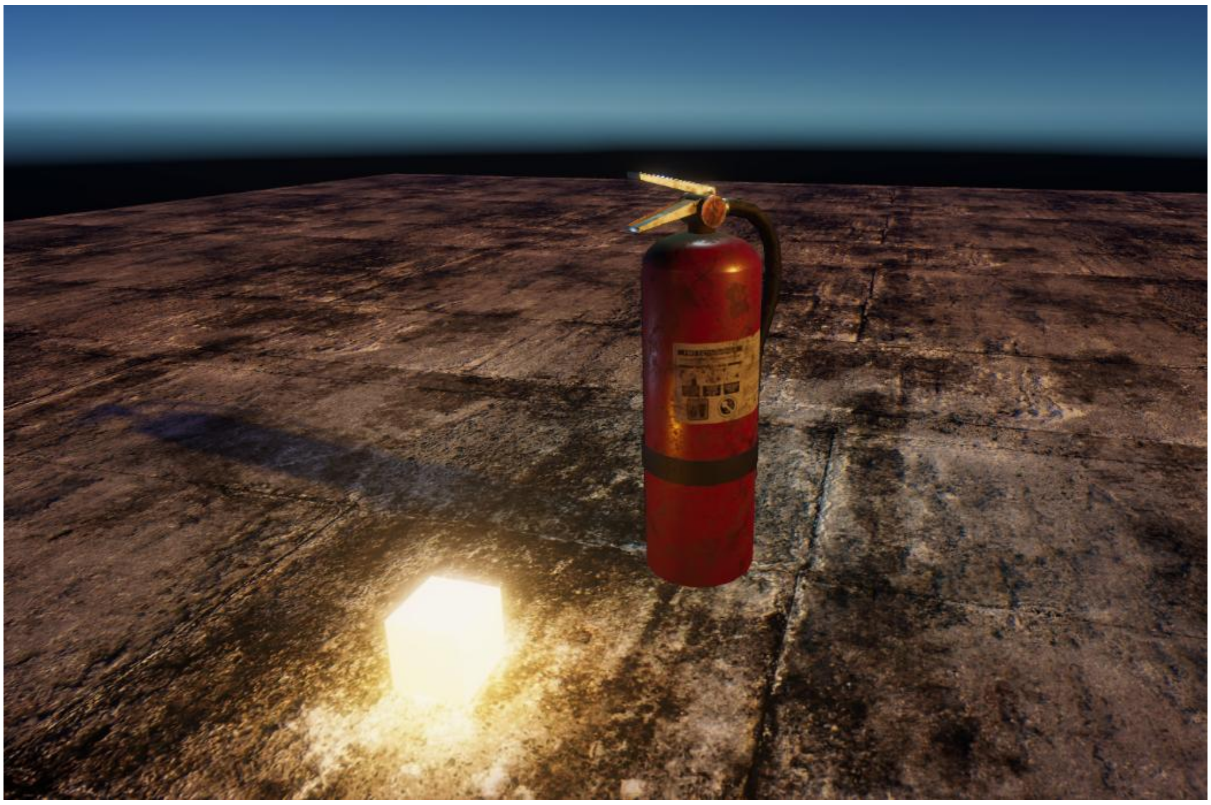
Add Override

A Global Volume was added to the scene with ACES tonemapping, a subtle vignette, and moderate bloom. ACES tonemapping ensures physically correct colour grading and preserves highlight detail under the dusk lighting. The vignette darkens the edges slightly, helping to focus the viewer's attention on the hero asset without being intrusive. Bloom adds a gentle glow to bright areas, particularly the emissive cube and the directional sunlight, enhancing realism and supporting the dusk atmosphere. A before/after comparison shows improved contrast, more visually pleasing highlights, and a clearer mood when post-processing is enabled.

Before



After



Unity vs Unreal Material and Workflow Comparison

Texture Mapping and PBR Inputs

One key difference is Unreal expects Roughness as a 0–1 map, whereas Unity's URP may interpret Smoothness instead; this required minor inversion adjustments to maintain consistent surface response.

Dissolve/Burn-Out Effect

Unreal's Material Editor terminology differs: "Opacity Mask" corresponds to Unity's AlphaClipThreshold, and "Emissive Colour" corresponds to URP's Emission input. Functionally, the visual result is similar, but Unreal requires all opacity-controlled materials to have Blend Mode set to Masked to achieve the hard dissolve effect.

Lighting Considerations

For minimal dusk lighting in Unreal, a Directional Light was placed at a low angle with warm tint, and a Sky Light provided cooler ambient fill, mirroring the Unity setup. Reflection captures were added to evaluate the hero asset's PBR response. Both engines support real-time shadows, but defaults differ: Unreal tends to produce softer shadow bias by default, whereas Unity's URP requires explicit tweaking of shadow strength and colour for dusk ambience.

Key Differences and Pitfalls

The most noticeable differences lie in terminology and default material behaviour. Unity separates "Smoothness" and "Metallic" inputs, while Unreal often combines them in a metallic-roughness workflow. Normal maps must be imported as Linear in Unity, whereas Unreal automatically interprets them. Shader Graph allows procedural nodes directly in the graph, whereas Unreal uses Material Functions or nodes like Noise and Lerp, which may require more connections to replicate the same effect.

Pipeline Standardization

To streamline a production workflow across both engines, it would be best to maintain consistent PBR texture naming and import settings, clearly document how each map is interpreted, and standardize procedural effect parameters (e.g., dissolve thresholds, edge colour/intensity). Overall, the exercise highlights that while both engines share the same physically-based rendering principles, terminology, default material behaviour, and procedural workflows differ.