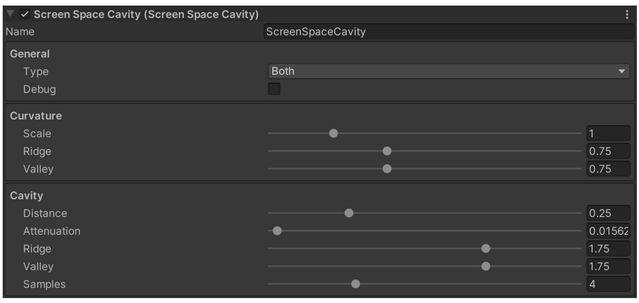
URP-ScreenSpaceCavity

How to preview

- Install Universal Render Pipeline.
- Download and import the Unity package.
- Open scene from **Assets/PolygonStarter/Scenes/Demo.unity**.

Tested with Unity version - 2020.3 URP version - 10.7

Configurable parameters



- **Type**:
  - **Curvature** - highlights only the edges of objects.
  - **Cavity** - highlights the edges with the Ambient Occlusion effect.
  - **Both** - well, it's understandable .
- **Curvature**:
  - **Scale** - effect width.
  - **Ridge** - effect ntensivity for ridge (white).
  - **Valley** - effect ntensivity for valley (black).
- **Cavity**:
  - **Distance** - distance of effect from edge.
  - **Attenuation** - fading out the effect relative to the camera (relevant for nearby objects).
  - **Ridge** - effect ntensivity for ridge (white).
  - **Valley** - effect ntensivity for valley (black).
  - **Samples** - number of passes to calculate the effect.

Shader setup

Here is an example of parts of the code for the shader to work with Cavity:

```
#if defined (_SCREEN_SPACE_CAVITY)
  #include "CavityInput.hlsl"
#endif
```

```
#pragma multi_compile_fragment _ _SCREEN_SPACE_CAVITY
#pragma multi_compile _ _CAVITY_DEBUG
```

```
#if defined (_SCREEN_SPACE_CAVITY)
  if (_CavityEnabled)
  {
    float2 normalizedUV = GetNormalizedScreenSpaceUV(input.positionCS);
    half cavity = SampleCavity(normalizedUV);
    #ifdef _CAVITY_DEBUG
      albedo.rgb = cavity * 2.0;
    #else
      bakedGI *= cavity * 4.0;
      lightColor *= cavity * 4.0;
    #endif
  }
#endif
```

The main thing is to get the cavity value and use it to apply the color:

```
#include "CavityInput.hlsl"
float2 normalizedUV = GetNormalizedScreenSpaceUV(input.positionCS);
half cavity = SampleCavity(normalizedUV);
color *= cavity * 4.0;
```

Attention! A custom shader must have passes for normals and depths.

Notes

- As an example, use the free POLYGON Starter Pack asset from reputable Synty Studios.
- If you are not familiar with the Universal Render Pipeline, you can find the official tutorial here.
- Writing Shaders.

Good to everyone!