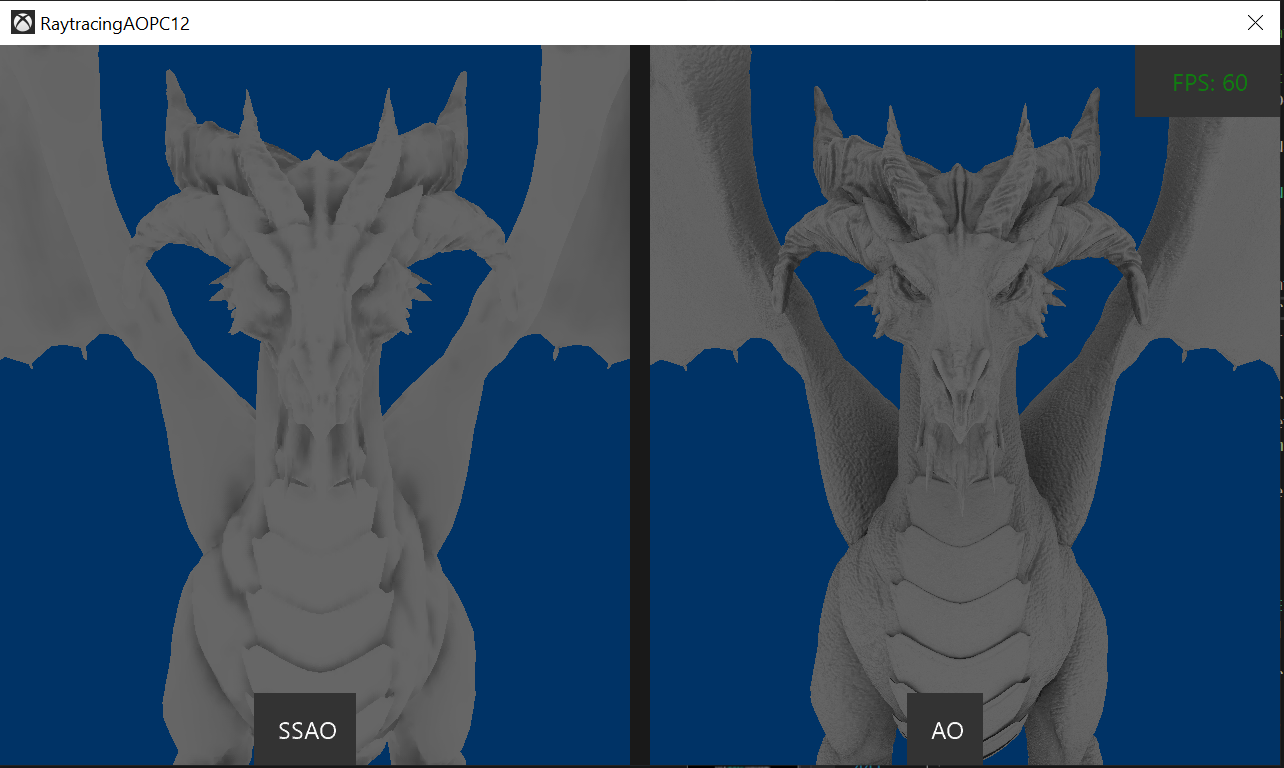
Raytracing Ambient Occlusion PC (DXR)

*This sample is compatible with the Windows 10 May 2019 Update SDK (18362)*

# Description

Games currently utilize Screen Space Ambient Occlusion (SSAO) to add definition and shadow to models during execution. However, with the introduction of DXR, the algorithm that SSAO approximates, Ambient Occlusion (AO), is becoming more feasible at interactive framerates. This sample demonstrates the visual differences between Screen Space Ambient Occlusion and Ambient Occlusion using DirectX 12 with DXR.



# Using the sample

This sample uses the following controls.

|  |  |
| --- | --- |
| Action | Keyboard |
| Toggle Comparison View | S key |
| Zoom In / Out | A / D keys |
| Swap Between SSAO  (Only applies outside of Toggle Comparison View) | Tab key |
| Options Menu | F1 key |
| Change Selected Option  (Only applies when the Options Menu is onscreen) | Up / Down arrow keys |
| Increase / Decrease Selected Option  (Only applies when the Options Menu is onscreen) | Left / Right arrow keys |
| Reset Options  (Only applies when the Options Menu is onscreen) | R key |
| FPS Toggle | F key |
| Exit | Escape key |

# Setup notes

# DirectX Raytracing requires:

* Windows 10 October 2018 Update (17763) or later
* DirectX12 video card & driver that supports DirectX Raytracing
  + e.x. nVIDIA: Pascal, Volta, or Turing card

# Implementation notes

## Normal Maps

# The implementation of SSAO is based on that of the MiniEngine Sample. A spherical sampler is used to collect the data necessary to compute the occlusion value per pixel. Additionally, SSAO does not normally support normal maps due to the lack of screen space information as well as the spherical sampler. Contrariwise, AO can utilize the normal maps in lieu of the surface normal to bias ray casts. Currently, the only sample to utilize normal maps is the Dragon mesh.

## AO Performance

# Only one thread is used per pixel, meaning that all AO rays spawned from a collision are launched sequentially. To improve performance, a Rasterization Render call can be added prior to running the AO shader. The Rasterization call would create a list of hit locations for rays spawned from the camera, allowing multiple threads to be launched per pixel to trace the AO rays. Lastly, a parallel reduction shader must be created to average the results of the multiple threads.

# Update history

Initial release December 2018. Removed support for the fallback layer in June 2019.