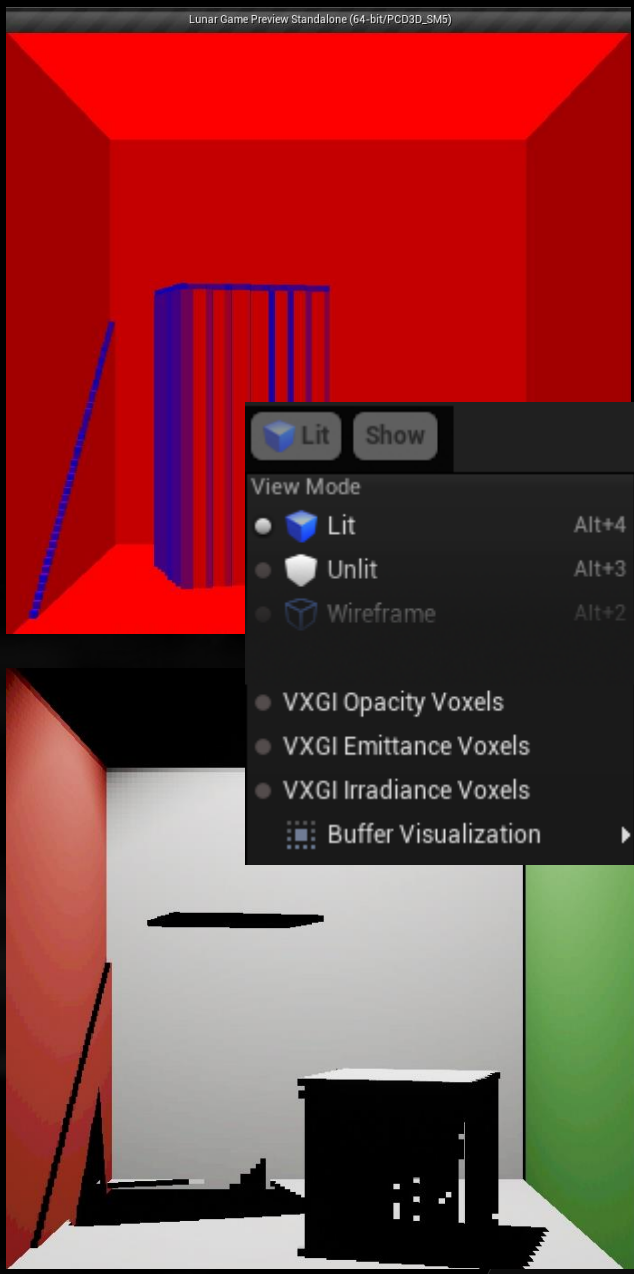
VXGI IN UE4

- Beta version now available on NvPhysX Github to UE4 licensees
- Revoxelizes the whole scene on every frame (for simplicity)
- Can cast multi-bounce GI from emissive materials & multiple lights (shadow mapped or not)
- With the default parameters, should work and look the same on all DX11 GPUs
- Example scenes (Cornell Box and SciFiHallway) are provided

NVIDIA Voxel Global Illumination (VXGI)

VXGI BRING UP

- How to enable it
  - Check “VXGI Diffuse / Enable Diffuse Tracing” in the PostProcessVolume
  - Check “VXGI Indirect Lighting” on real lights and make them Movable
  - Set console variable “r.VXGI.DiffuseTracingEnable 1” (default)
  - Check “Used With VXGI Voxelization” on surface materials (default)
- Use “VXGI Opacity Voxels” view mode to see if all objects are represented as occluders
- Use “VXGI Emittance Voxels” view mode to see if directly lit surfaces and emissive objects are represented as emitters



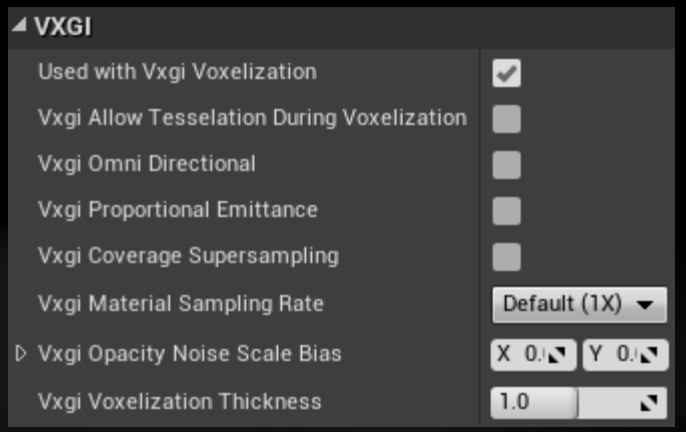
VXGI PARAMETERS

Main Console Variables

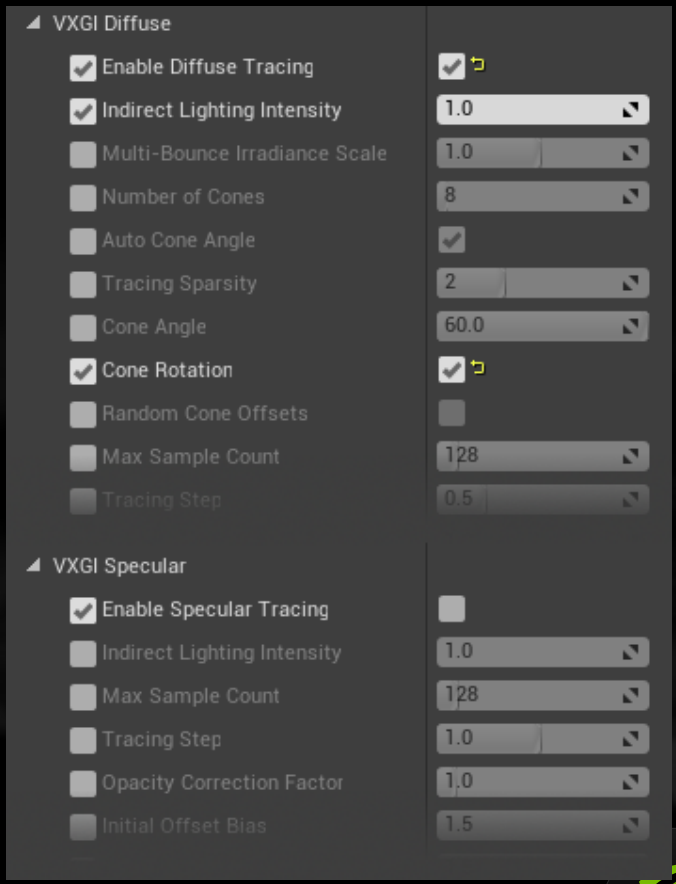
r.VXGI.MapSize
r.VXGI.Opacity6D
r.VXGI.Emittance6D
r.VXGI.NvidiaExtensionsEnable
r.VXGI.StoreEmittanceInHdrFormat
r.VXGI.EmittanceStorageScale
r.VXGI.EmittanceInterpolationEnable
r.VXGI.HighQualityEmittanceDownsamplingEnable

r.VXGI.DiffuseTracingEnable
r.VXGI.SpecularTracingEnable
r.VXGI.EmissiveMaterialsEnable
r.VXGI.DiffuseMaterialsEnable
r.VXGI.Range
r.VXGI.DebugClipmapLevel
r.VXGI.AmbientOcclusionMode
r.VXGI.MultiBounceEnable

Material Parameters



Cone Tracing Parameters



VXGI REFLECTIONS

- How to enable VXGI Reflections
  - Set console variable “r.VXGI.SpecularTracingEnable 1”
  - Check “VXGI Specular / Enable Specular Tracing” in the PostProcessVolume
- When enabled, VXGI Specular Tracing
  - Disables SSR
  - Replaces SSR & light probes with the VXGI Specular Tracing result
  - Can be combined with SSR if “r.VXGI.CombineSpecularWithSSR 1” is set
- Limitation
  - VXGI reflections are meant to render glossy reflections (roughness >= 0.2 or so)
  - VXGI cannot render non-glossy reflections well (e.g. perfect mirrors)

PERFORMANCE KNOBS

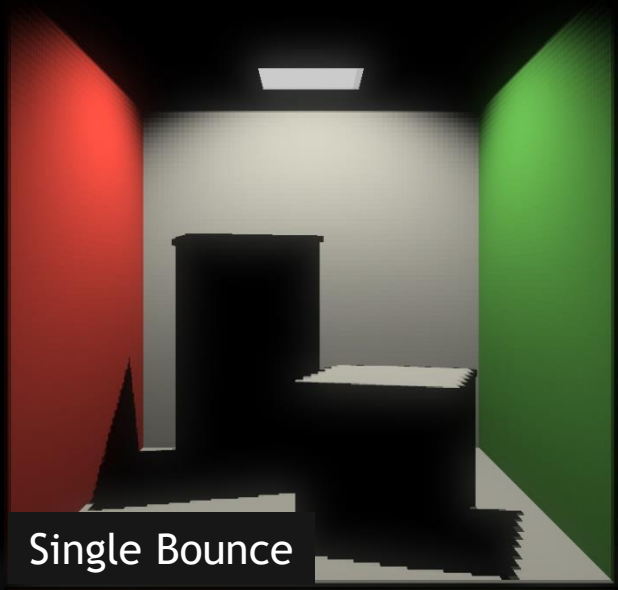
- You can type the "ProfileGPU" command to get a breakdown of the GPU time
  - Output of "stat unit" command also includes VXGI WS (World-Space) and VXGI SS (Screen-Space) GPU times
- To improve cone tracing performance, edit the PostProcessVolume settings:
  - Disable specular tracing
  - Set "Diffuse Tracing / Number of Cones" to 4
  - Set "Diffuse Tracing / Tracing Sparsity" to 4 (quarter-resolution tracing)
  - Enable "Diffuse Tracing / Use Temporal Filtering" (to remove flickering artifacts)
- To improve voxelization performance, set these console variables:
  - Set "r.VXGI.MapSize 64" (to improve both voxelization and tracing performance)
  - Set "r.VXGI.MultiBounceEnable 0" (to improve voxelization performance)
  - Set "r.VXGI.HighQualityEmittanceDownsamplingEnable 0" (to improve voxelization performance)

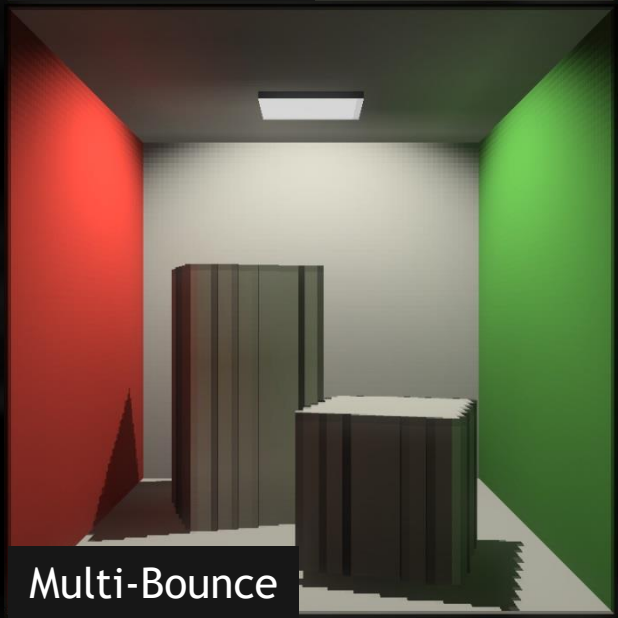
MULTI-BOUNCE GI

- When multi-bounce support is enabled, VXGI computes a 3D indirect irradiance map after voxelization
- Uses the irradiance map during voxelization on the next frame
  - Adds one more bounce on every frame
  - Makes the whole scene appear in specular reflections

- How to enable it:
  - Set console varuable "r.VXGI.MultiBounceEnable 1"
  - Tune "VXGI Diffuse / Multi-Bounce Irradiance Scale" in PostProcessVolume until it looks right
    - Irradiance may blow up if this value is too high
  - Use "VXGI Irradiance Voxels" view mode to see the indirect irradiance map

- Pictures on the right: visualization of emittance voxels





AREA LIGHTS

- VXGI is good at computing lighting (including soft shadows) from an arbitrary number of area lights of arbitrary shapes.
- Using actual emissive surfaces is often better than faking them with traditional local lights.
- For both performance and quality reasons, we recommend to not use any fill lights when lighting scenes with VXGI.
- With multi-bounce support, the entire scene can be lit by area lights only.
- Pictures on the right: Cornell Box lit by one emissive object