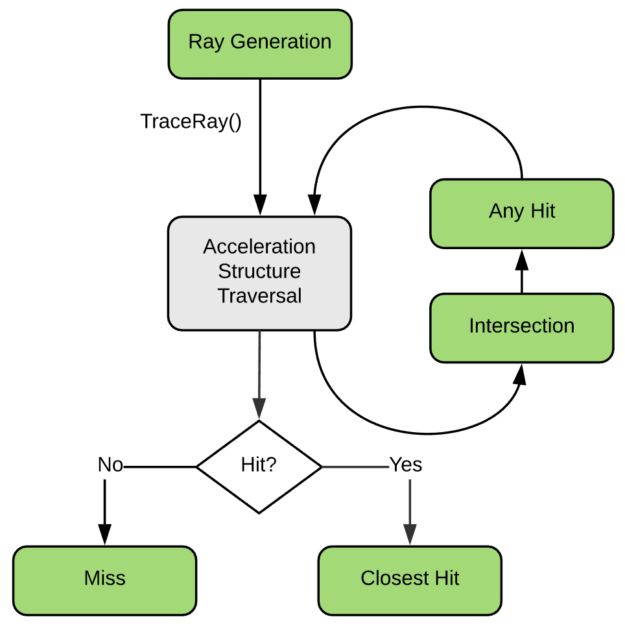
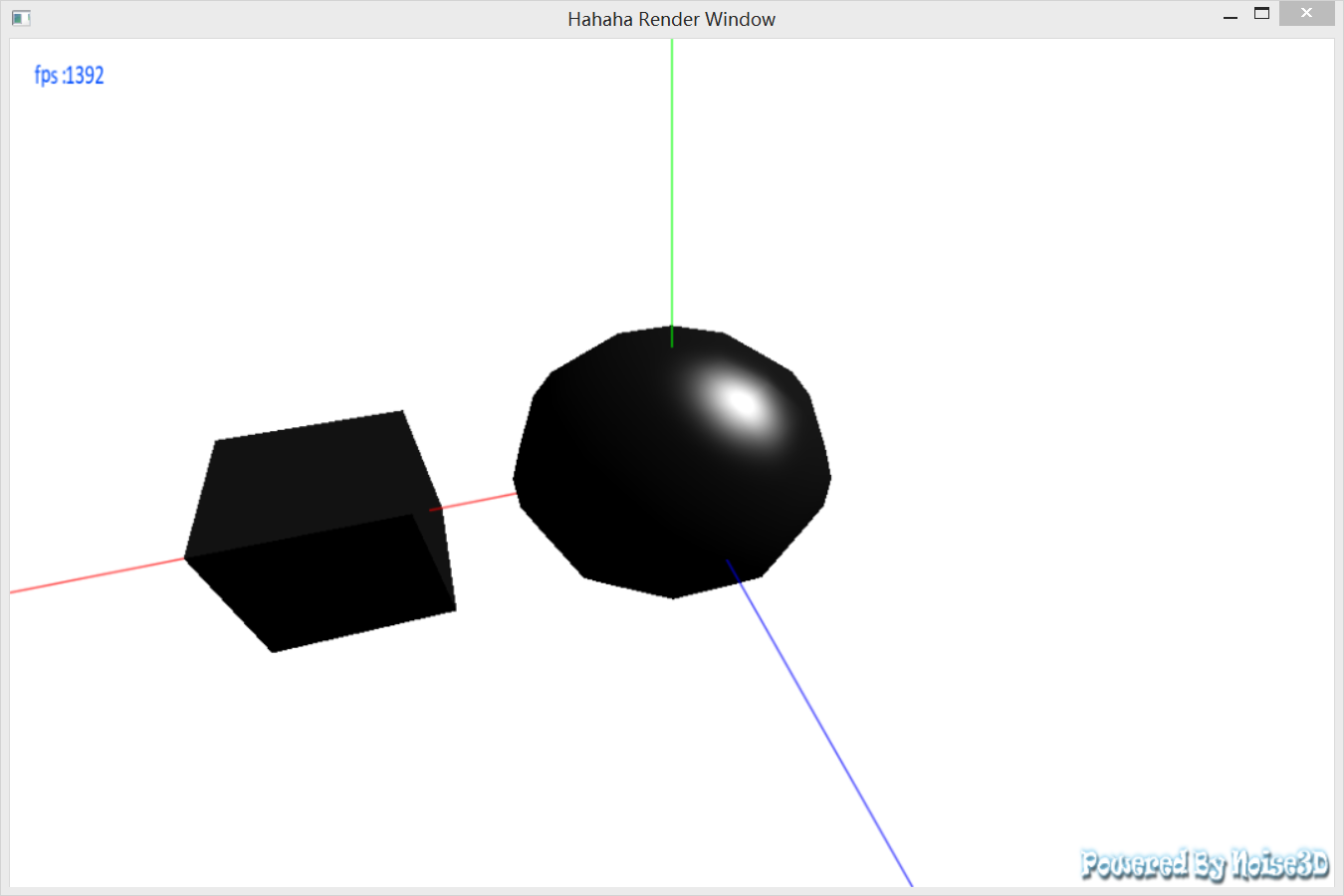
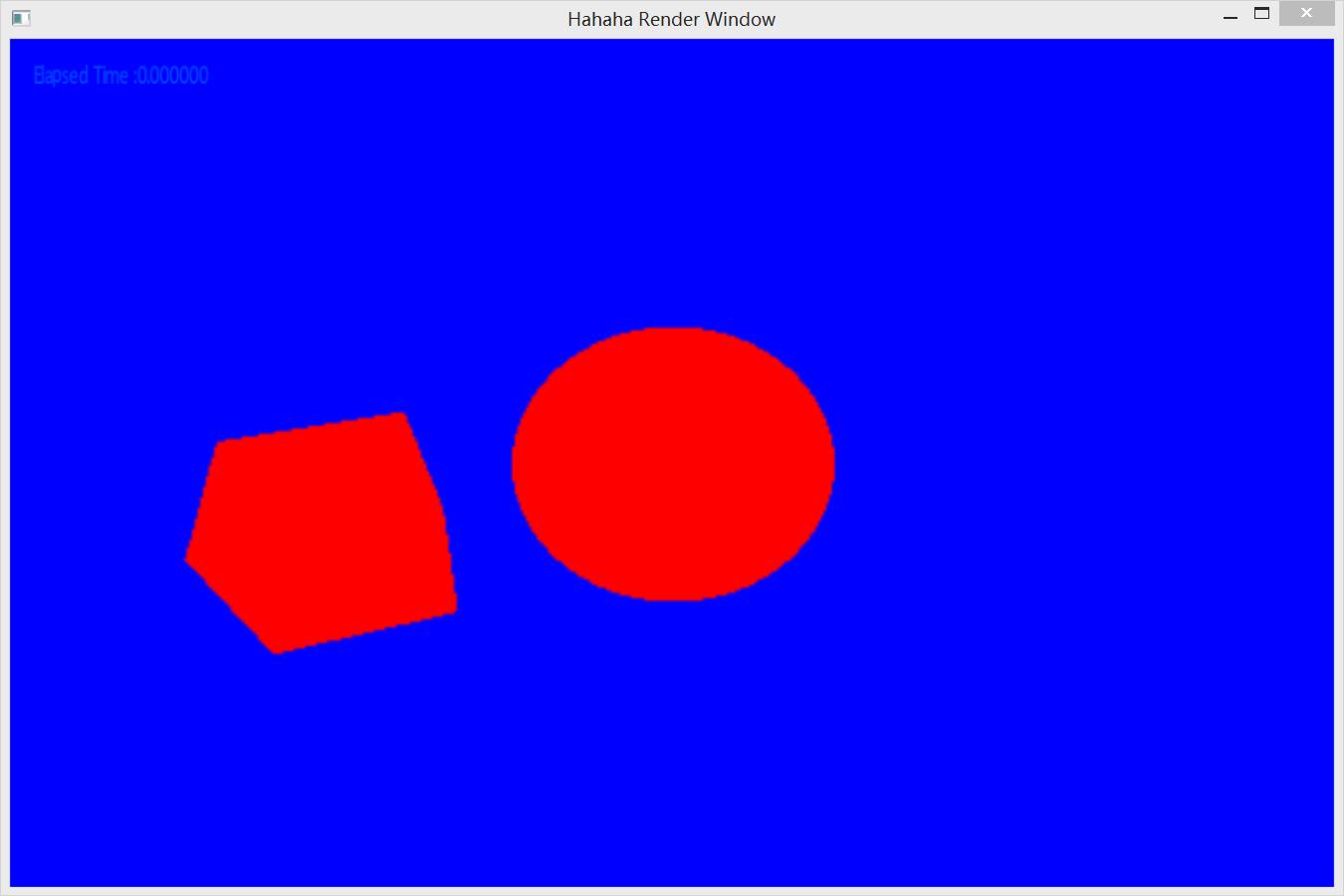
<https://devblogs.nvidia.com/introduction-nvidia-rtx-directx-ray-tracing/>



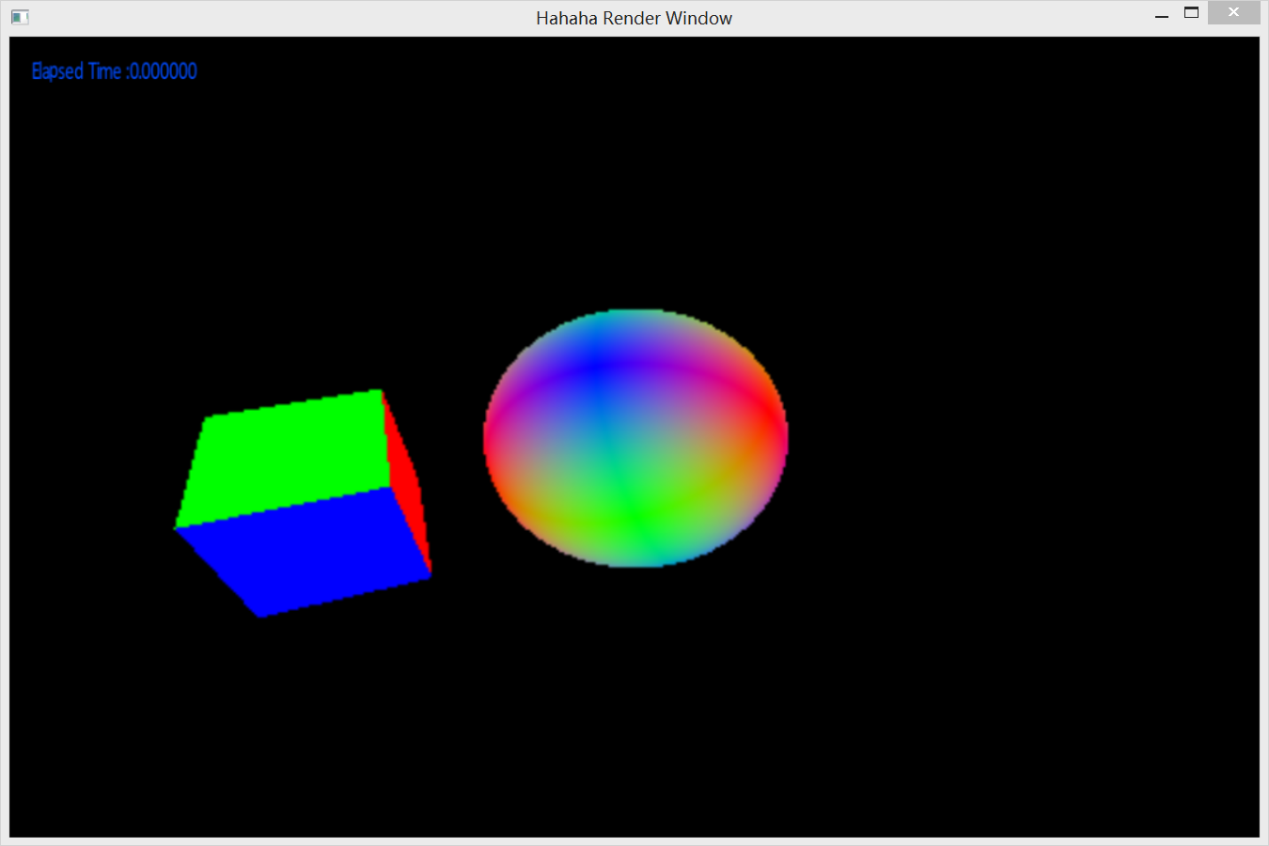
<https://developer.nvidia.com/rtx/raytracing/dxr/DX12-Raytracing-tutorial-Part-1>

**Noise3D Path Tracer Soft Shader Demo：**





|  |
| --- |
| /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*                      Path Tracer Shader: Minimal              Noise3D::PathTracer's soft shader "hello world",              for demonstration and test usage  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  #include "Noise3D.h"  using namespace Noise3D;  void Noise3D::GI::PathTracerShader\_Minimal::ClosestHit(N\_Ray ray, const N\_RayHitInfo & hitInfo, N\_TraceRayPayload & in\_out\_payload)  {      in\_out\_payload.radiance = GI::Radiance(1.0f, 0, 0);  }  void Noise3D::GI::PathTracerShader\_Minimal::Miss(N\_Ray ray, N\_TraceRayPayload & in\_out\_payload)  {      in\_out\_payload.radiance = GI::Radiance(0, 0, 1.0f);  } |



|  |
| --- |
| /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*                      Path Tracer Shader: Minimal              Noise3D::PathTracer's soft shader "hello world",              for demonstration and test usage  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  #include "Noise3D.h"  using namespace Noise3D;  void Noise3D::GI::PathTracerShader\_Minimal::ClosestHit(N\_Ray ray, const N\_RayHitInfo & hitInfo, N\_TraceRayPayload & in\_out\_payload)  {      in\_out\_payload.radiance = GI::Radiance(abs(hitInfo.normal.x), abs(hitInfo.normal.y), abs(hitInfo.normal.z));  }  void Noise3D::GI::PathTracerShader\_Minimal::Miss(N\_Ray ray, N\_TraceRayPayload & in\_out\_payload)  {      in\_out\_payload.radiance = GI::Radiance(0, 0, 0);  } |