COM的全称是Component Object Model，它让DX称为独立的编程语言并且向下兼容。我们通常使用特定的函数或者其它COM接口来获取COM接口引用的指针。另外，COM对接口是引用计数的，使用完毕后，需要调用Release方法来释放它们。为了管理COM对象的生命周期，Windows Runtime Library提供了wrl.h来管理COM对象的智能指针。龙书中主要用到了ComPtr三个方法：

1.Get:返回其包含的COM接口。

2.GetAddressOf:返回其包含的COM接口的指针指向的地址。

3.Reset:将ComPtr接口设置为空指针，并且减少其包含的COM接口的引用计数。

Inline RayTracing:

需要相应的设置pipeline，也需要产生对应光线

TMin与TMax是用来确定光线的有效发生碰撞的位置,一条光线除此之外还有原点和方向

当完成计算后，TMax会变成最近的交叉点位置对应的T

需要使用BuildRaytracingAccelerationStructure()在init中构建加速结构且其位于commandList中

//Inline RayTracing function illustration

//The autogenerated index of the instance in the top - level structure for the closest hit committed so far.

//在顶层加速结构上自动生成的closet hit的index of instance

//instance index

//query.CommittedInstanceIndex();

//The autogenerated index of the primitive within the geometry inside the bottom-level acceleration structure instance for the closest hit committed so far.

//triangle index within the geometry object.

//query.CommittedPrimitiveIndex();

//The autogenerated index of the geometry in the bottom-level acceleration structure for the closest hit committed so far.

//geometry index

//query.CommittedGeometryIndex();

//This is a float representing the parametric distance at which the closest committed hit so far lies.

//给出了现在的TMax，可由此计算出当前hit position

//defines the current TMax point along the ray according to the following formula: Origin + (Direction \* CommittedRayT).

//query.CommittedRayT();

//Given attributes a0, a1 and a2 for the 3 vertices of a triangle

//返回一个float2 float2.x is the weight for a1 and float2.y is the weight for a2

//a = a0 + float2.x \\* (a1-a0) + float2.y\\* (a2 -- a0).

//query.CommittedTriangleBarycentrics();

//返回一个bool值

//TRUE means front face, FALSE means back face.

//query.CommittedTriangleFrontFace();