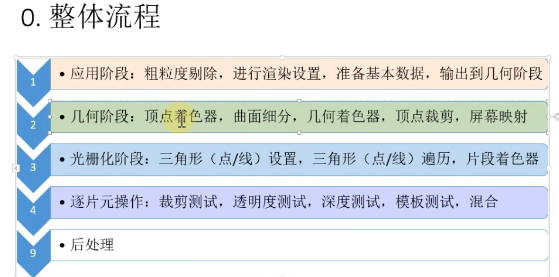
渲染管线的流程



[渲染管线](https://docs.unity.cn/cn/2020.3/Manual/render-pipelines.html)确定场景中对象的显示方式，分为三个主要阶段。

第一步是剔除；它列出了需要渲染的对象，最好是那些对摄像机可见的对象（[视锥体剔除](https://docs.unity.cn/Manual/UnderstandingFrustum.html)）和其他对象不遮挡的对象（[遮挡剔除](https://docs.unity.cn/Manual/OcclusionCulling.html)）。

第二个阶段渲染是指将这些对象绘制到基于像素的缓冲区中（通过正确的光照以及它们的一些属性）。

最后，可以在这些缓冲区上执行后期处理操作，例如，应用颜色分级、泛光和景深，从而生成发送到显示设备的最终输出帧。

主要是顶点着色器和片段着色器

[选择和配置渲染管线和光照解决方案 - Unity 手册](https://docs.unity.cn/cn/2020.3/Manual/BestPracticeLightingPipelines.html)

1. 着色器是在图形处理单元 (GPU) 上运行的程序或程序集合的通用名称。
2. 计算着色器：计算着色器允许程序员利用 GPU 的大量并行处理能力，用于任何类型的数学运算，如光照剔除、粒子物理或体积模拟。
3. 前向渲染根据影响对象的光源在一个或多个通道中渲染每个对象。光源本身也可以通过前向渲染进行不同的处理，具体取决于它们的设置和强度。
4. 帧率：每秒的帧数，等于帧数/时间
5. Shader语言

- HLSL：基于DirectX的High Level Shading Language

- GLSL：基于OpenGL的OpenGL Shading Language

- CG：NVIDIA（英伟达）公司的C for Graphic，更底层，和HLSL差不多

Unity用HLSL、CG或基本的GLSL

1. GPU计算不依赖其他类型数据，且独立运算快（控制弱，计算强）
2. Unity中的坐标系：

- local：左手系，受对象rotation影响

- global：左手系，全局的，不受对象rotation影响

- view port：右手系，原点摄像机位置，z为镜头之后

- screen：左手系，左下角是原点，→x，上y

7. 光栅化：像素中心是否在三角形内

点是否在矩阵内：xMin <= x < xMax && yMin <=y < yMax

1. Unity内的Tex Normal: 最高 MainTex：中等 Emit：最低
2. sRGB是人眼看的才开启（基本纹理、UI），像法线（Normal、紫、细节）、高度（形变）、反射（发光）、粗糙度这些不用开启。
3. UE贴图类型：SRGB的会标注，没有就不开 （连线的节点）

使用2的幂次方尺寸的, 有助于生成Mipmap并进行流送。

**基础颜色贴图（Base Color） SRGB Diffuse、BaseColor、Albedo、D、BC**

基础颜色贴图定义了物体在无光照情况下的颜色。它储存了非导体的漫反射色和金属导体的镜面反射值。（使用RGB）

**法线贴图（Normal Map、紫色）**

法线贴图用于增加表面的细节，使其看起来更加立体。它通过影响模型表面的阴影来表达凹凸效果。(使用RGB)

**高光贴图（Specular Map）**

高光贴图用于表现光线照射到模型表面时的反射属性，从而区分不同材质的反光程度。

**粗糙度贴图（Roughness Map、灰色）**

粗糙度贴图使用黑白颜色来定义材质的粗糙度。白色表示完全粗糙，黑色表示光滑。（使用R/G/B）