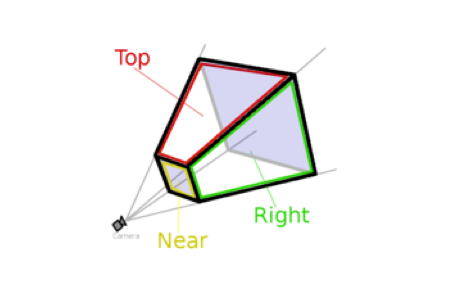
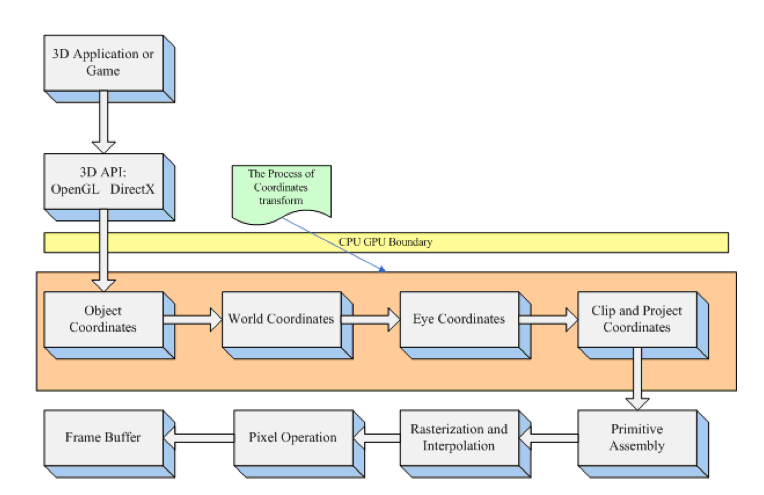
计算机图形学基础：

所谓GPU的渲染管线，听起来好像很高深的样子，其实我们可以把它理解为一个流程，就是我们告诉GPU一堆数据，最后得出来一副二维图像，而这些数据就包括了”视点、三维物体、光源、照明模型、纹理”等元素。   
在各种图形学的书中，渲染管线主要分为三个阶段：应用程序阶段、几何阶段、光栅阶段。



Object space，模型坐标空间；World space，世界坐标空间；Eye space，观察坐标空间；Clip and Project space，屏幕坐标空间。

eye space坐标转换到project and clip space坐标的过程其实就是一个投影、剪裁、映射的过程。因为在不规则的视锥体内剪裁是一件非常困难的事，所以前人们将剪裁安排到一个单位立方体中进行，这个立方体被称为规范立方体（CCV），CVV的近平面（对应视锥体的近平面）的x、y坐标对应屏幕像素坐标（左下角0、0），z代表画面像素深度。所以这个转换过程事实上由三步组成：   
（1），用透视变换矩阵把顶点从视锥体变换到CVV中；   
（2），在CVV内进行剪裁；   
（3），屏幕映射：将经过前两步得到的坐标映射到屏幕坐标系上。



Shader（着色器）实际上就是一小段程序，它负责将输入的Mesh（网格）以指定的方式和输入的贴图或者颜色等组合作用，然后输出。

绘图单元可以依据这个输出来将图像绘制到屏幕上。输入的贴图或者颜色等，加上对应的Shader，以及对Shader的特定的参数设置，将这些内容（Shader及输入参数）打包存储在一起，得到的就是一个Material（材质）。之后，我们便可以将材质赋予合适的renderer（渲染器）来进行渲染（输出）了。

Unity中编写shader的语言叫做ShaderLab，shaderlab就是CG着色器语言。CG：C for graphics。是M家和Nvidia协作开发的协议。HLSL和CG是一种语言，联合推出以打击GLSL。**在Unity中写Shader约等于给DirectX写Shader。**

**需要注意的是，Shader想要使用到游戏物体上，一般得有个媒介，这个媒介就是材质（Material）。我们把Shader作用于材质，接着再把材质对应地作用于给游戏物体，这样写的Shader就间接地给物体表面使用了。**

我们知道，计算机图形学的中渲染管线一般可以分为两种类型：

1.**固定功能渲染管线(fixed-functionrendering pipelines)**

2.**可编程渲染管线(programmablerendering pipelines)**

按这样的分类思路，在Unity中，Shader便可以分成如下三种基本类型：

1.**固定功能着色器（Fixed Function Shader）**

2.**表面着色器（Surface Shader）**

3.**顶点着色器&片段着色器 （Vertex Shader & Fragment Shader）**