

图形渲染管线 (GRAPHICS PIPELINE)

指的是一堆原始图形数据途经一个输送管道，期间经过各种变化处理最终出现在屏幕的过程

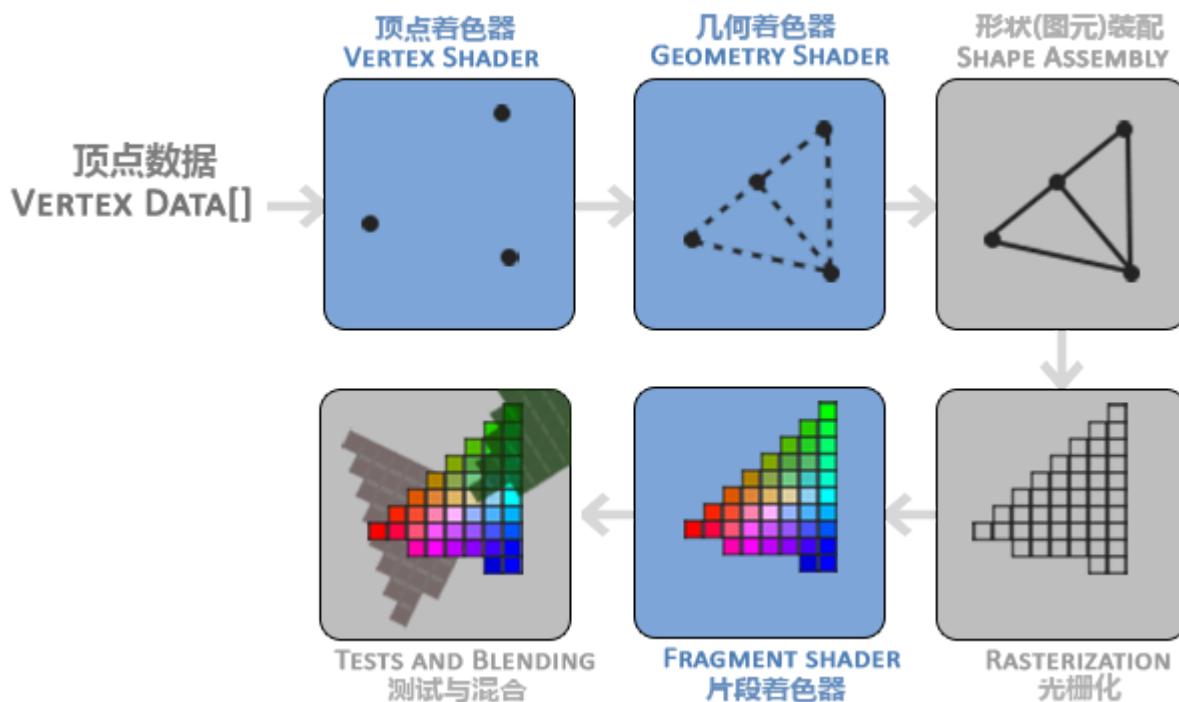
图形渲染管线可以被划分为两个主要部分：

- 第一部分把**3D坐标转换为2D坐标**，
- 第二部分是**把2D坐标转变为实际的有颜色的像素**。
- **输入：一组3D坐标**
- **输出：屏幕上的有色2D像素输出**

图形渲染管线被划分为几个阶段，每个阶段将会把前一个阶段的输出作为输入

当今大多数显卡都有成千上万的小处理核心，它们在GPU上为每一个（渲染管线）阶段运行各自的小程序，从而在图形渲染管线中快速处理你的数据。

这些小程序叫做**着色器(Shader)**



0. 顶点数据

以数组的形式传递3个3D(XYZ)坐标作为图形渲染管线的输入，用来表示一个三角形，这个数组叫做**顶点数据(Vertex Data)**；

顶点数据是一系列**顶点**的集合。

一个**顶点(Vertex)**是一个**3D坐标**的数据的集合。

1. 顶点着色器

主要的目的是把3D坐标转为另一种3D坐标
可以对顶点属性进行一些基本处理

- **输入**: 一个单独的顶点
- **输出**: 当前顶点的另一种3D坐标

(2 .) 几 何 着 色 器

顶点着色器阶段的输出可以选择性地传递给几何着色器

输入: 一组顶点作为, 这些顶点形成**图元(primitive)**

通过发出新的顶点来形成新的(或其他)图元来生成其他形状

3 . 图 元 装 配 (P R I M I T I V E A S S E M B L Y)

输入: 顶点着色器/几何着色器输出的**所有顶点**

将所有点装配成指定图元的形状

4 . 光 栅 化 R A S T E R I Z A T I O N

- 把图元映射为最终屏幕上相应的像素,
- 生成供**片段着色器**使用的**片段(Fragment)**
- 在片段着色器运行之前会执行**裁切(Clipping)**。
 - 裁切会**丢弃超出视图以外**的所有像素, 用来提升执行效率。

5 . 片 段 着 色 器

计算一个像素的最终颜色

包含3D场景的数据 (比如光照、阴影、光的颜色等等), 用来计算最终像素的颜色。

6 . A L P H A 测 试 和 混 合 (B L E N D I N G) 阶 段

- 检测片段的对应的**深度** (和**模板(Stencil)**) 值,
 - 用它们来判断这个像素是其它物体的前面还是后面, 决定是否应该丢弃
- 检测**alpha**值 (**alpha**值定义了一个物体的透明度) 并对物体进行**混合(Blend)**