Creator3D：shader10\_解析【Creator3D shader】的正确姿势（汇总1）

## 前言

不知不觉自己研究shader已经快一个月了，一路靠爱发电坚持到现在。

菜鸟按照自己对新知识的自学方式，先从简单示例入手进行学习，然后再回过头来了解理论知识。在前边的9篇文章中，菜鸟从最简单的效果入手，一步步的去实现更好看的效果。那么在这篇文章中，菜鸟就将一些理论性的知识进行补充。

## 回归

首先总结一下之前的文章，有些伙伴可能没有看过，不妨点击去看一下

a

[Creator3D:基础1\_一起学shader\_红色小球](https://mp.weixin.qq.com/s/-Q7Obw\_1zFjWEzV2aleQPw)

[Creator3D:基础2\_一起学shader\_变色小球](https://mp.weixin.qq.com/s/8mZ9MU4w\_MPnJQ4cALAvRQ)

[Creator3D:基础3\_一起学shader\_波浪小球](https://mp.weixin.qq.com/s/2CzOaGu3RFefrDjFl044fg)

[Creator3D:基础4\_一起学shader\_没有尽头的路](https://mp.weixin.qq.com/s/S4uxhOIcaOuWz0mdMymxKA)

[Creator3D\_shader5\_代码如何控制effect中的属性](https://mp.weixin.qq.com/s/g21-HJUlvHF1nqrJoa-43Q)

[Creator3D:shader6\_程序员也是会心动的](https://mp.weixin.qq.com/s/RZ9PBQXwYHPKMHAJsFOeqg)

[Creator3D:shader7\_尽然还有双pass这波操作](https://mp.weixin.qq.com/s/-3Eq\_d6KQuW7Ck2CQ0KdwQ)

[Creator3D:shader8\_这种shader怎样配标题](https://mp.weixin.qq.com/s/5oi0fyalYBq-Fop3dxCnwA)

[Creator3D:shader9\_这样的内发光你喜欢不](https://mp.weixin.qq.com/s/O4lj3PL5ORuSA4Rt-4uq8Q)

## 正文

### 1.什么是shader

Shader，翻译过来是：着色器。网上对他的解释是：用来实现图像渲染的，用来替代固定渲染管线的可编辑程序。其中Vertex Shader主要负责顶点的几何关系等的运算，Pixel Shader主要负责片源颜色等的计算。

### 2.Creator3D的shader 着色器

- 在unity中，shader 的写法有三种：Surface Shaders 表面着色器，Vertex/Fragment Shaders 顶点/片断着色器，Fixed Function Shaders 固定管线着色器。

- 而在Creator3D shader写法是：Vertex/Fragment Shaders 顶点/片断着色器

- Vertex Shader：顶点着色器，输入的单位是顶点，也就是说每个顶点都会调用一次顶点着色器，顶点着色器的本身是不可以创建或者销毁任何顶点，同时也没有办法得到顶点与顶点之间的关系。其主要工作为：坐标转换与逐顶点光照，还可以后续阶段（片元着色器）所需要的数据

- Fragment Shader：大家也可以叫它片段着色器或者像素着色器。他输入的是上一个阶段（顶点着色器）对顶点信息插值得到的结果，而他输出的是一个或者多个颜色值。

### 3.Creator3D的shader 文档

在[材质系统](https://docs.cocos.com/creator3d/manual/zh/material-system/overview.html)的文章中对着色流程，顺序，内置结构，语法框架，可配参数，内置参数等都做了说明。菜鸟的建议是不要求都看懂，但是最起码要过一遍，菜鸟看了好几遍了，目前为止还有好多看不懂的。

### 4.渲染

既然说起shader，那么也有必要了解一下渲染。

![](https://gitee.com/carlosyzy/Creator3D\_Mesh\_Basics/raw/master/file/base10\_1.png)

渲染的流程可以分为三个阶段：应用阶段，几何阶段，光栅阶段；

- 应用阶段从名字找们可以看出，这个阶段是由找们的应用主导的，因此通常由 CPU 负责实现。换句话说，我们这些开发者具有这个阶段的绝对控制权。在这一阶段中，开发者有 3 个主要任务：首先，我们需要准备好场景数据，例如摄像机的位置、视锥体、场景中包含了哪些模型、使用了哪些光源等等：其次，为了提高渲染性能，我们往往需要做 一个粗粒度剔除（ culling ）工作，以把那些不可见的物体剔除出去，这样就不需要再移交给几何阶段进行处理：最后，我们需要设置好每个模型的渲染状态。这一阶段最重要的输出是渲染所需的几何信息，即渲染图元。通俗来讲，渲染图元可以是点、线、三角面等。这些渲染图元将会被传递给 下一个阶段 ― 几何阶段．

- 几何阶段用于处理所有和我们要绘制的几何相关的事情。例如，决定需要绘制的图元是什么，怎样绘制它们，在哪里绘制它们。这一阶段通常在 GPU 上进行。几何阶段负责和每个渲染图元打交道，进行逐顶点、逐多边形的操作，几何阶段的一个重要任务就是把顶点坐标变换到屏幕空间中，再交给光栅器进行处理。通过对输入的渲染图元进行多步处理后，这一阶段将会输出屏幕空间的二维顶点坐标、每个顶点对应的深度值、着色等相关信息，并传递给下一个阶段。

- 光栅化阶段这 · 阶段将会使用上个阶段传递的数据来产生屏幕上的像素，并渲染出最终的图像。这一阶段也是在 GPU 上运行。光栅化的任务主要是决定每个渲染图元中的哪些像素应该被绘制在屏幕上。它需要对上一个阶段得到的逐顶点数据（例如纹理坐标、顶点颜色等）进行插值，然后再进行逐像素处理。和上一个阶段类似，光栅化阶段也可以分成更小的流水线阶段。

从上面的介绍中，我们可以到的应用阶段是CPU负责操作，而几何阶段和光栅阶段都是GPU进行操作。

- 应用阶段：CPU

- 其实就游戏开发的过程，场景，光源，模型，包括设置各类参数。

- 几何阶段：GPU

- 顶点着色器，逐顶点进行操作，进行顶点坐标的转换等操作

- 光栅阶段：GPU

- 片元着色器。

那么此时就验证了我在别的地方看见的一句话：着⾊器其实就是一段在GPU运行的程序。我们平时的程序，是在CPU运行。

### 5.CPU和GPU直接的通信

- 首先咱们看一下CPU在与GPU通信之前自身所需要的操作：

- 1.将数据加载到显存中

- 2.设置渲染状态

- 当CPU完成上边的操作后会发送一个命令，这个命令就是Draw Call，接收者是GPU。当给定一个Draw Call，GPU将会根据渲染状态和所有输入的顶点数据进行计算，最终显示到屏幕上。

## 6.地址

- 微信公众号：搬砖小菜鸟

![](https://gitee.com/carlosyzy/Creator3D\_Mesh\_Basics/raw/master/file/gzh.jpg)

扫码关注公众号，发送"gitee"可获取源码