# Creator3D:基础1\_一起学shader\_红色小球

# 前言

Shader，中文名为着色器（本人目前初步了解顶点着色器和片元着色器），对很多开发者来说它是一个神秘的存在，想学但是又不知道如何去学。看着网上的各种雾化，水纹波浪的效果着实眼馋，作为小白的我决定自研一下。主要会以例子的方式来和大家分享。本篇为基础篇，那么就从最简单的来，坚决不能想着一口吃个胖子。

---

# 效果展示

![](https://gitee.com/carlosyzy/Creator3D\_Mesh\_Basics/raw/master/file/base1\_1.png)

----

# 正文

# 1.文档资料

[YAML 101](https://docs.cocos.com/creator3d/manual/zh/material-system/yaml-101.html)

[Effect 语法](https://docs.cocos.com/creator3d/manual/zh/material-system/effect-syntax.html)

[Pass 可选配置参数](https://docs.cocos.com/creator3d/manual/zh/material-system/pass-parameter-list.html)

[常用 shader 内置 Uniform](https://docs.cocos.com/creator3d/manual/zh/material-system/builtin-shader-uniforms.html)

文档是必须要读的，哪怕你是和我一下的小白，第一次不要求看懂，但是最好过一遍，有个印象，方便后边回过头来查找

# 2.effect格式

新创建一个effect 大概的格式就是下面这样。

```

// Effect Syntax Guide: https://github.com/cocos-creator/docs-3d/blob/master/zh/material-system/effect-syntax.md

CCEffect %{

techniques:

- name: opaque

passes:

- vert: general-vs:vert # builtin header

frag: unlit-fs:frag

#注释材质和颜色属性

#properties: &props

# mainTexture: { value: white }

# mainColor: { value: [1, 1, 1, 1], editor: { type: color } }

- name: transparent

passes:

- vert: general-vs:vert # builtin header

frag: unlit-fs:frag

blendState:

targets:

- blend: true

blendSrc: src\_alpha

blendDst: one\_minus\_src\_alpha

blendSrcAlpha: src\_alpha

blendDstAlpha: one\_minus\_src\_alpha

#properties: \*props

}%

CCProgram unlit-fs %{

precision highp float;

#include <output>

in vec2 v\_uv;

uniform sampler2D mainTexture;

uniform Constant {

vec4 mainColor;

};

vec4 frag () {

vec4 color=vec4(1.0,0.0,0.0,1.0);

return CCFragOutput(color);

}

}%

```

# 3.给小球一个红颜色

在effect格式中的代码中我们能看见这样三行代码，文档中是这样写的：\*\*每个 Pass 只有两个必填参数：vert 和 frag 声明了当前 pass 使用的 shader, 格式为 片段名:入口函数名

这个名字可以是本文件中声明的 shader 片段名, 也可以是引擎提供的标准头文件。\*\*

其中：

vert 表示的是顶点着色器的入口函数

frag 表示的片元着色器的入口

```

passes:

vert: general-vs:vert # builtin header

frag: unlit-fs:frag

```

本次讲的是上色，那么用到的是片元着色气，也就是下边的这段代码：

```

CCProgram unlit-fs %{

precision highp float;

vec4 frag () {

vec4 color=vec4(1.0,0.0,0.0,1.0);

return color;

}

}%

```

接下来我会详细对上边的代码进行讲解，尽量让每个人都能看懂：

1.precision highp float;

- float类型在 shaders 中非常重要，

- 所以精度非常重要。更低的精度会有更快的渲染速度，但是会以质量为代价。

- 你可以选择每一个浮点值的精度。

- 在第一行（precision highp float;）我们就是设定了所有的浮点值都是高精度。

- 但我们也可以选择把这个值设为“低”（precision lowp float;）或者“中”（precision mediump float;）。

2.vec4 frag () 函数入口

- 大概写一下shader中的数据类型：float, vec2, vec3, vec4, mat2, mat3, mat4, sampler2D and samplerCube，至于具体的区别，我都列出来，大家可以百度查一下。

3.vec4 color=vec4(1.0,0.0,0.0,1.0);

- 定义一个颜色，其中vec4中的x,y,z,w分别代表颜色的r,g,b,a;我用的是红色，大家可以任意设置（\*\*0-1直接的浮点数\*\*）

4.return color;

- 将颜色返回进行上色

5.删除没有用到的代码，最终的代码是：

```

CCEffect %{

techniques:

- name: opaque

passes:

- vert: general-vs:vert # builtin header

frag: unlit-fs:frag

}%

CCProgram unlit-fs %{

precision highp float;

vec4 frag () {

vec4 color=vec4(1.0,0.0,0.0,1.0);

return color;

}

}%

```

# 4.模型显示

- 1.创建一个material

- 2.在material的effect下选择你上边写的effect文件

- 3.将material拖到模型cc.ModelComponten组件下的materials中

- 4.进行预览，效果如上图

# 5.地址

- gitee：https://gitee.com/carlosyzy/Creator3D\_Mesh\_Basics

- 微信公众号：搬砖小菜鸟

![](https://gitee.com/carlosyzy/cocos\_creator\_physical\_cutting/raw/master/file/4.png)

# 6.推荐

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