# Creator3D:shader5\_代码如何控制effect中的属性

# 前言

之前做了一个球形的波浪水球，当时提到一句有点像加载进度条，那么下去我就实现了一下这个功能，游戏的加载进度用它来显示。

# 效果展示

![](https://gitee.com/carlosyzy/Creator3D\_Mesh\_Basics/raw/master/file/base5\_1.gif)

# 正文

## 1.概述

在上边的效果显示中，大家可以看见，球形的波浪高度，随着游戏加载的进度逐渐变高，游戏加载到100%的时候，球也正好被填充满。那么大家就会想到，波浪小球的shader实现效果是在effec文件中，而游戏的加载进度是在游戏代码中，应该怎样在代码中去设置effect中的属性。

## 2.实现

Effect代码：

```

CCEffect %{

  techniques:

  - name: opaque

    passes:

    - vert: general-vs:vert # builtin header

      frag: unlit-fs:frag

      properties: &props

        mainTexture:    { value: white }

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

        mScale:         { value: [1, 1, 1,1] }

        mPos:           { value: [1, 1, 1,1] }

        mHeight:           { value: 0 }

  - 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>

  #include <cc-global>

  #include <output>

  #include <cc-local-batch>

  in vec3 v\_position;

  in vec2 v\_uv;

  uniform sampler2D mainTexture;

  uniform Constant {

    vec4 mainColor;

    vec4 mScale;

    vec4 mPos;

    float mHeight;

  };

  vec4 frag () {

    //顶点坐标，法线坐标都是基于世界坐标系的

    vec4 color=mainColor;

    if(v\_position.y+sin((v\_position.x+cc\_time.x)\*7.0)/40.0>mPos.y-mScale.y/2.0+mHeight){

        color = vec4(1.0,1.1,1.0,0.1);

    }

    return CCFragOutput(color \* texture(mainTexture, v\_uv));

  }

}%

```

新建一个effect文件,我们需要下面3个地方的改动

- 1.添加可编辑的属性，mScale,mPos,mHieght 是我们需要添加的。

```

 properties: &props

        mainTexture:    { value: white }

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

        mScale:         { value: [1, 1, 1,1] }

        mPos:           { value: [1, 1, 1,1] }

        mHeight:           { value: 0 }

```

- 2.声明添加属性的类型

```

  uniform Constant {

    vec4 mainColor;

    vec4 mScale;

    vec4 mPos;

    float mHeight;

  };

```

- 3. 实现波浪效果

```

vec4 frag () {

//顶点坐标，法线坐标都是基于世界坐标系的

//mPos 小球的坐标

//mScale 小球的大小

//mHeight 波浪在小球中的高度

    vec4 color=mainColor;

    if(v\_position.y+sin((v\_position.x+cc\_time.x)\*7.0)/40.0>mPos.y-mScale.y/2.0+mHeight){

        color = vec4(1.0,1.1,1.0,0.1);

    }

    return CCFragOutput(color \* texture(mainTexture, v\_uv));

  }

```

游戏脚本中的代码

```

//获取小球的材质

this.loadSphereCm = this.loadSphere.getComponent(ModelComponent);

let material: Material = this.loadSphereCm.material;

//设置材质中mainColor属性值

material.setProperty("mainColor", new Vec4(0.0, 0.7, 0.8, 1.0));

let pos: Vec3 = this.loadSphere.position;

let scale: Vec3 = this.loadSphere.scale;

//设置材质中mPos属性值

material.setProperty("mPos", new Vec4(pos.x, pos.y, pos.z, 0.0));

//设置材质中mScale属性值

material.setProperty("mScale", new Vec4(scale.x, scale.y, scale.z, 0.0));

this.mProgress = 0;

this.schedule(() => {

this.mProgress += 1;

if (this.mProgress > 100) {

this.mProgress = 100;

}

//根据加载进度设置mHeight的值

material.setProperty("mHeight", scale.y \* (this.mProgress / 100));

this.labelLoad.getComponent(LabelComponent).string = "资源初始化进度" + this.mProgress + "%";

}, 0.07);

```

## 3.地址

- 微信公众号：搬砖小菜鸟