WebGL通过着色器进行绘图，来看一下如何通过JS改变着色器的属性

**导出着色器属性**

如下看代码

<!-- 顶点着色器 -->

<script id="vertexShader" type="x-shader/x-vertex">

    // 导出属性

    attribute vec4 a\_Position;

    void main(){

        gl\_Position = a\_Position;

        gl\_PointSize = 50.0;

    }

</script>

...

<script>

    // 获取着色器属性点

    const a\_Position = context.getAttribLocation(context.program, 'a\_Position');

    // 修改属性点

    context.vertexAttrib3f(a\_Position, 0.0, 0.0, 0.0);

    // 绘制顶点

    context.drawArrays(context.POINTS, 0, 1);

</script>

- 首先声明一个变量，该变量的类型是vec4（顶点类型），attribute为表示导出该变量（只限于vec4类型的属性）

attribute vec4 a\_Position;

- 通过上下文获取属性点

const a\_Position = context.getAttribLocation(context.program, 'a\_Position');

- 通过上下文修改属性点

context.vertexAttrib3f(a\_Position, 0.0, 0.0, 0.0);

扩展知识点：

vertexAttrib方法用于修改vertex着色器的属性，其方法有如下

vertexAttrib1f(location,v0)

vertexAttrib2f(location,v0,v1)

vertexAttrib3f(location,v0,v1,v2)

vertexAttrib4f(location,v0,v1,v2,v3)

其命名规则为vertexAttrib+参数个数+参数类型(f代表float除此之外还有i代表整型，v代表数字)，以后会遇到很多这种类型的方法

**导出颜色属性**

如下代码

<canvas id="canvas"></canvas>

...

<!-- 片元着色器 -->

<script id="fragmentShader" type="x-shader/x-fragment">

    // 导出颜色属性

    precision mediump float;

    uniform vec4 u\_FragColor;

    void main() {

        gl\_FragColor = u\_FragColor;

    }

</script>

<script>

    ...

    // 获取属性元素

    const u\_FragColor = context.getUniformLocation(context.program, 'u\_FragColor');

    // 修改颜色

    context.uniform4f(u\_FragColor, Math.random(), Math.random(), Math.random(), 1.0);

    context.clear(context.COLOR\_BUFFER\_BIT);

    context.drawArrays(context.POINTS, 0, 1);

</script>

- 导出颜色属性代码如下，uniform是导出声明

uniform vec4 u\_FragColor;

- 第一行precision mediump float;是必须的，其定义了float的精度