The Book of Shaders by *Patricio Gonzalez Vivo & Jen Lowe*

Bahasa Indonesia - *Tiếng Việt* - 日本□語 - 中文版 - 한국어 - *Español* - *Portugues* - *Français* - *Italiano* - *Deutsch* - *Русский* - *English*

运行你的 shader

现在你可能跃跃欲试，想在你熟悉的平台上小试牛刀了。接下来会有一些比较流行的平台的示例代码，展示如何在这些平台上配置 shader。（在这个 github 仓库 中有本章的三种平台的示例代码）

注释 **1**：如果你不想用这些平台来运行 shader，且你想在浏览器外使用 shader，你可以下载glslViewer。这个 MacOS+树莓派程序直接在终端运行，并且是为本书的例子量身打造的。

注释**2**：如果你想用 WebGL 显示 shader，并不关心其他平台，你可以用glslCanvas 。这个 web 工具本来是为本书设计的，但是太好用了，所以我常常用在其他项目中。

Three.js

为人谦逊而非常有才华的 Ricardo Cabello (也就是 MrDoob)和许多贡献者 一起搭了可能是 WebGL 最知名的平台，Three.js。你可以找到无数程序示例，教程，书籍，教你如何用这个 JavaScript 库做出酷炫的 3D 图像。

下面是一个你需要的例子，教你用 three.js 玩转 shader。注意 `id="fragmentShader"`脚本，你要把下面的代码拷到里面。

下面是一个 HTML 和 JS 的示例，

```
<body>
    <div id="container"></div>
    <script src="js/three.min.js"></script>
    <script id="vertexShader" type="x-shader/x-vertex">
        void main() {
            gl_Position = vec4( position, 1.0 );
        }
```

```
</script>
<script id="fragmentShader" type="x-shader/x-fragment">
    uniform vec2 u_resolution;
    uniform float u_time;

    void main() {
        vec2 st = gl_FragCoord.xy/u_resolution.xy;
        gl_FragColor=vec4(st.x,st.y,0.0,1.0);
    }
</script>
<script>
    var container;
    var camera, scene, renderer, clock;
    var uniforms;

    init();
    animate();

    function init() {
        container = document.getElementById( 'container' );

        camera = new THREE.Camera();
        camera.position.z = 1;

        scene = new THREE.Scene();
        clock = new THREE.Clock();

        var geometry = new THREE.PlaneBufferGeometry( 2, 2 );

        uniforms = {
            u_time: { type: "f", value: 1.0 },
            u_resolution: { type: "v2", value: new THREE.Vector2() }
        };

        var material = new THREE.ShaderMaterial( {
            uniforms: uniforms,
            vertexShader: document.getElementById( 'vertexShader' ).textContent,
            fragmentShader: document.getElementById( 'fragmentShader' ).textContent
        } );

        var mesh = new THREE.Mesh( geometry, material );
        scene.add( mesh );

        renderer = new THREE.WebGLRenderer();
        renderer.setPixelRatio( window.devicePixelRatio );

        container.appendChild( renderer.domElement );

        onWindowResize();
        window.addEventListener( 'resize', onWindowResize, false );
    }

    function onWindowResize( event ) {
        renderer.setSize( window.innerWidth, window.innerHeight );
        uniforms.u_resolution.value.x = renderer.domElement.width;
        uniforms.u_resolution.value.y = renderer.domElement.height;
    }

    function animate() {
```

```
            requestAnimationFrame( animate );
            render();
        }

        function render() {
            uniforms.u_time.value += clock.getDelta();
            renderer.render( scene, camera );
        }
    </script>
</body>
```

Processing

2001年由Ben Fry 和 Casey Reas 创建，Processing是一个极其简约而强大的环境，非常适合初尝代码的人（至少对于我来是这样）。关于 OpenGL 和视频，Andres Colubri为 Processing 平台做了很重要的更新，使得环境非常友好，玩 GLSL shader 比起以前大大容易了。Processing 会在你的 sketch 的 `data` 文件夹搜索名为 `"shader.frag"` 的文件。记得把这里的示例代码拷到你的文件夹里然后重命名 shader。

```
PShader shader;


void setup() {
  size(640, 360, P2D);
  noStroke();

  shader = loadShader("shader.frag");
}


void draw() {
  shader.set("u_resolution", float(width), float(height));
  shader.set("u_mouse", float(mouseX), float(mouseY));
  shader.set("u_time", millis() / 1000.0);
  shader(shader);
  rect(0,0,width,height);
}
```

在 2.1 版之前的版本运行 shader，你需要在你的 shader 文件开头添加以下代码：`#define PROCESSING_COLOR_SHADER`。所以它应该看起来是这样：

```
#ifdef GL_ES
precision mediump float;
#endif

#define PROCESSING_COLOR_SHADER

uniform vec2 u_resolution;
uniform vec3 u_mouse;
uniform float u_time;

void main() {
    vec2 st = gl_FragCoord.st/u_resolution;
    gl_FragColor = vec4(st.x,st.y,0.0,1.0);
}
```

更多 Processing 的 shader 教程戳 tutorial。

openFrameworks

每个人都有自己的舒适区，我的则是openFrameworks community。这个 C++ 框架打包了 OpenGL 和其他开源 C++ 库。在很多方面它和 Processing 非常像，但是明显和 C++ 编译器打交道一定比较麻烦。和 Processing 很像地，openFrameworks 会在你的 data 文件夹里寻找 shader 文件，所以不要忘记把你的后缀 `.frag` 的文件拷进去，加载的时候记得改名。

```
void ofApp::draw(){
    ofShader shader;
    shader.load("","shader.frag");

    shader.begin();
    shader.setUniform1f("u_time", ofGetElapsedTimef());
    shader.setUniform2f("u_resolution", ofGetWidth(), ofGetHeight());
    ofRect(0,0,ofGetWidth(), ofGetHeight());
    shader.end();
}
```

关于 shader 在 openFrameworks 的更多信息请参考这篇excellent tutorial，作者是 Joshua Noble。

< < *Previous* *Home* *Next* > >