

---

# Project 3

## 一、项目目录及文件说明

/PJ3 目录: Project 3 项目根目录

/PJ3/lib 目录: 项目提供的库文件目录

/PJ3/config 目录: 项目提供的库文件目录

/PJ3/image 目录: 项目提供的贴图文件目录

/PJ3/model 目录: 项目提供的模型文件目录

/PJ3/project3.html 文件: 可执行的 HTML 文件

/PJ3/project3.js 文件: 实现本 Project 需求功能的 JS 文件

## 二、开发及运行环境

开发环境: IntelliJ IDEA 2020.3.1 及 Google Chrome 89.0.4389.90 (设置允许跨域访问), 带有 NVIDIA GeForce RTX 2060 with-Max Q Design 显卡的计算机。

运行环境: Google Chrome 浏览器 (支持 HTML5, JavaScript 和 WebGL 的版本, 设置允许跨域访问), 带有支持 OpenGL ES2.0 显卡的计算机。

## 三、运行及使用方法

运行方法: 确保 project3.html、project2.js 和 lib 目录、config 目录、image 目录、model 目录在同一目录下, 用设置了允许跨域访问的 Chrome 浏览器打开 project3.html 文件即可运行。

使用方法: 运行后, 可以使用键盘上的 W/S 控制摄像机前后移动, 使用 A/D 控制摄像机左右移动 (移动的速度为 30 个单位/秒), 使用 J/L 控制摄像机以 Y 轴为轴左右旋转 (旋转速度为 60 角度/秒), 使用 I/K 控制摄像机以 X 轴为轴上下旋转 (旋转速度为 60 角度/秒), 使用 F 控制打开/关闭点光源。

## 四、项目中的亮点

1. 实现了纹理和光照的结合: 在纹理的着色器中传入光源的相关参数, 通过与 model obj 中计算光源类似的计算方法 (纹理的顶点着色器中没有 a\_Color 参数, 计算主要在片段着色器中完成), 在片段着色器中, 通过 `vec4 color = texture2D(u_Sampler, v_TexCoord)` 获取的纹理颜色即可作为 a\_color, 以此实现了纹理和光照的结合。

2. 实现雾化效果: 同时实现了 obj 对象和纹理的雾化, 通过在着色器中加入 foggyColor、foggyDist 等参数, 通过计算、混合颜色实现雾化效果。

## 五、开发过程中遇到的问题及解决办法

1. 问题: 实现纹理与光照结合时, 难以在纹理的顶点着色器中获得 a\_color;  
解决办法: 在纹理的片段着色器中的纹理颜色即可代表 a\_color, 因此在实

---

现纹理与光线结合时,可以将计算纹理光线颜色等操作放在片段着色器中进行。

2. 问题: 动画效果较难实现;

解决方法: 在绘制 `scene obj` 时, 先判断当前 `obj` 是否需要动画, 若需要则计算旋转角度、移动距离等变量, 再设置 `model` 变化矩阵, 最后进行绘制, 实现动画效果。

3. 问题: 进行移动、旋转等操作时的时候, 更新向量参数较为困难;

解决方法: 结合一定的数学知识, 在每次更新时获取移动或旋转方向上的单位向量, 再根据该单位向量更新 `eye` 和 `at` 向量。

## 六、项目可能存在的缺陷及你的思考

思考: 由于需要加载 `obj` 和图片文件, 在打开 `project3.html` 页面时会有一瞬间的卡顿, 虽然时间很短, 几乎不影响使用, 但仍然有进一步优化的空间。

## 七、对本课程 Project 的意见及建议

老师和助教讲得很详细, 文档和项目需求也很清晰, 没有什么建议。