

# 实验报告

## 一、实验目的

熟悉 OpenGL 的框架和使用，复习计算机图形学中图形变换和投影的知识。

具体任务：

用OpenGL编写程序，完成一个3D场景的绘制

### 1. 场景构建

- 1) 构建一个简单场景（如秀丽山河、画廊、海底世界等），利用天空盒构建场景的背景。
- 2) 场景中至少包含2个静态3D物体和1个动态3D物体，物体无需纹理贴图，纯色即可。自行设计动态物体的运动路线，随机的和固定的路线即可，无需考虑物理规律。
- 3) 程序支持场景文件（自定义文件格式）的读入，可通过场景文件编辑场景，如修改物体初始位置、颜色等。

### 2. 场景漫游

自行设计交互方式，实现相机位置、视线方向的改变。

### 3. 碰撞检测

- 1) 使用AABB碰撞检测算法实现物体与物体之间的碰撞检测。
- 2) 当发生碰撞时在UI界面给出提示，物体不可穿过物体。

### 4. 滤镜

利用帧缓冲实现一个滤镜，如反相、灰度、核效果等

## 二、实验方法

### 1. 场景构建中：

1) 天空盒是通过渲染一个正方体实现的。正方体的每个面贴图，在移动摄像机时天空盒中心跟随摄像机，同时在渲染天空盒时关闭深度测试，最后渲染，实现雪地场地。

2) 3 个物体均为正方体，渲染方式和天空盒一样，都绑定自己编写的顶点着色器和片元着色器。动态物体在两个静态物体之间做匀速直线往复运动。在初始化时设置一个16ms（60帧每秒）的定时器，定时刷新页面实现动态效果。

3) 场景文件采用键值的方式读取，可以设置静态物体的位置、颜色以及动态物体的颜色。动态物体的位置默认在静态物体中间。读取时会进行文件合法性、位置重叠、颜色重复的检测。

### 2. 场景漫游中：

相机位置及视线方向使用 Demo 中提供的方式，可以使用按键、鼠标操控。

### 3. 碰撞检测中：

使用 AABB 碰撞检测算法，两物体碰撞时速度设为反向，同时发出信号给 UI 界面的槽函数，显示发生碰撞的静态物体的颜色。

### 4. 滤镜中：

使用帧缓冲。首先编写反相滤镜的着色器并初始化，在绘制的过程中，绑定 frameBufferObject，接着正常渲染场景，之后如果是滤镜状态，绑定反相着色器，将 FBO 中的内容存到纹理中，并贴到一个和屏幕大小相同的四边形上；如果是正常状态，绑定正常着色器输出。

### 三、实验结果

见同目录下录屏

### 四、交互方式

双击运行，右上角可以选择原滤镜和反相滤镜。

下方第一个按钮 Load Configuration File 可以选择文件，加载配置文件（TXT），加载成功后会渲染三个正方体。

格式不对或加载失败或点击第二个按钮会加载默认配置。

下方的碰撞检测提醒会在碰撞时输出，同时文本颜色和发生碰撞的静态正方体颜色相同。

配置文件格式示例：

```
leftCubePos: -1.0, 0.0, 8.0
leftCubeColor: 1.0, 0.8, 0.0
rightCubePos: 1.0, 0.0, 4.0
rightCubeColor: 0.3, 0.7, 0.0
middleCubeColor: 0.8, 0.3, 0.6
speed: 0.4
```

如上示例，需要六个键名都存在且正确，位置颜色均为三维向量，每一维都在 -1 到 1 之间的浮点数。速度为浮点数。

### 五、编译环境和编译方式

编译环境

- 操作系统：Windows 10
- IDE: Visual Studio 2022 Community
  - 安装了 Visual Studio 的 Qt 扩展（Qt VS Tools）
- Qt 版本：Qt 6.8.0
- OpenGL 版本：Qt 中的 OpenGL 与 OpenGLWidget 组件

编译方式

Visual Studio 编译运行