

3D场景渲染器

邹林壮 计科拔尖2201 202208040412

本项目是一个使用OpenGL和GLFW、GLEW、GLM开发的3D场景渲染器。

它创建了一个交互式环境，包含多个对象，包括二维对象和三维对象，如地面、河流、树木、太阳、云、人、鱼、山、房屋、小草和星星，构建了一个生活在天圆地方世界的人物的日常场景。

该场景支持通过键盘和鼠标输入进行用户交互，以导航和操作环境，并且支持视角变化。

键盘和鼠标事件

键盘事件

- **ESC:** 关闭应用程序窗口。
- **上、下、左、右:** 移动场景中的人物。移动范围限制在地面上，超出范围时会循环回到另一侧，为了增加趣味性，人物可以穿越山体。
- **P:** 循环切换预定义的颜色集，改变人的身体颜色（影响身体、手臂和腿部）。
- **Q、E:** 调整透视投影的宽高比。**Q** 增加0.1，**E** 减少0.1（最小值为0.1）。
- **W、A、S、D:** 根据当前相机方向分别向前、向左、向后、向右移动相机。
- **空格键:** 触发人物的跳跃动画，在1秒内使用正弦函数平滑地升降。
- **ESCAPE键:** 退出按键。

鼠标事件

- **鼠标移动:** 通过调整偏航角（yaw）和俯仰角（pitch）控制相机方向，让用户环视场景。俯仰角限制在-89°到89°之间，以防止翻转。
- **鼠标滚轮:** 调整视野（fov）进行缩放。向上滚动减小fov（放大），向下滚动增大fov（缩小），范围为10°到90°。

代码架构和模块

代码按模块化方式组织，用于管理3D渲染和用户交互：

- **Mesh结构:** 定义3D对象的属性，包括顶点数组对象（VAO）、顶点缓冲对象（VBO）、元素缓冲对象（EBO）、绘制计数、绘制模式、位置、旋转、缩放和颜色。
- **setupMesh函数:** 初始化网格数据，设置缓冲区和顶点属性（位置和颜色）。
- **renderMesh函数:** 渲染网格，应用变换（平移、旋转、缩放）并为着色器程序设置统一变量。
- **全局变量:** 存储场景范围的设置，如着色器程序ID、屏幕尺寸（1600x1600）、相机参数和人的颜色调色板。
- **generateSphere函数:** 为球形对象（如太阳和星星）生成顶点和索引数据。
- **updateGrassColor函数:** 使用基于时间和位置的正弦函数动态更新小草颜色。
- **工具函数:**
 - `get_OpenGL_info`: 显示OpenGL供应商、渲染器和版本信息。
 - `checkStatus`: 检查着色器或程序的编译/链接状态。

- `readShaderCode`: 从文件中加载着色器源代码。
- `installShaders`: 编译并链接顶点和片段着色器。
- `sendDataToOpenGL`: 初始化并配置所有场景对象。
- `paintGL`: 主渲染函数，更新动态对象并绘制场景。
- **事件处理程序**:
 - `key_callback`: 处理键盘输入。
 - `mouse_callback`: 处理鼠标移动以控制相机。
 - `scroll_callback`: 管理滚轮输入以调整fov。
- `initializedGL`: 设置GLEW和OpenGL配置，包括启用深度测试。
- `main`: 程序入口，初始化GLFW，创建窗口，设置回调并运行主循环。

创建的对象

场景包含以下对象，包括二维对象和三维对象以及带索引的对象，每个对象都有特定的几何形状和属性：

- **地面**: 表示地形的四边形，颜色为沙褐色。
- **河流**: 带有深蓝和浅蓝交替条纹的平面，略高于地面。
- **树木 (3个实例)**: 长方体树干和三角形树叶，放置在不同位置并缩放。
- **太阳**: 红色球体，绕场景固定高度旋转。
- **云 (3个实例)**: 白色三角形，位于天空不同位置和大小。
- **人**: 由多个部分组成，p键可以改变人物颜色，并且可以通过键盘上下左右进行平移：
 - **头部**: 红色长方体。
 - **身体**: 可通过 P 键改变颜色的长方体。
 - **左右手臂**: 附着在身体上的红色长方体。
 - **左右腿**: 从身体下方延伸的红色长方体。
- **鱼**: 在河流中水平来回移动的黄色形状，并且游动到河流边缘会自动旋转反方向游动。
- **山体**: 多层三角形结构，从底部深棕色渐变到顶部浅灰色。
- **房屋**: 带有斜屋顶和窗户的详细结构，颜色为桃色和棕色调。
- **房屋线框**: 白色线框勾勒房屋细节，使用线模式渲染。
- **小草 (10片)**: 三角锥形，颜色随时间动态变化（RGB基于正弦函数）。
- **星星 (5个实例)**: 绕Y轴旋转的小黄色球体，位于云附近和天空更高处。

透视投影参数及其效果

场景使用 `glm::perspective` 定义的透视投影，可调整的参数包括：

- **视野 (fov)**:
 - **范围**: 10°到90°，通过鼠标滚轮控制。
 - **效果**: 较小的fov（例如10°）使视野变窄，物体显得更大更近，适合放大查看。较大的fov（例如90°）拓宽视野，物体显得更小，显示更多场景。极端值可能导致感知扭曲——过小可能引起裁剪，过大可能夸大透视效果。
- **宽高比 (Aspect Ratio)**:

- **默认值:** 1.0 (1600/1600) , 可通过 Q 和 E 键调整。
- **效果:** 增加宽高比会水平拉伸场景, 使物体变宽。减小则压缩场景, 使物体变高。高宽高比 (如2.0) : 场景水平拉伸, 物体变宽, 地面和河流显得更长。
- **近远平面:**
 - **值:** 近=0.1, 远=20.0 (代码中固定) 。
 - **效果:** 近平面过小会牺牲精度, 过大则丢失近处细节。远平面过近裁剪远处对象, 过远则性能下降。当前值 (0.1到20.0) 适合场景规模 (物体距离相机约10单位) 。远平面限制最大渲染距离——过大浪费资源, 过小则远处的物体 (例如星星) 可能被裁剪。当前范围 (0.1到20.0) 在此场景规模下平衡了细节和性能。

深度测试

initializedGL函数里, 通过调用glEnable(GL_DEPTH_TEST)来启动深度测试