**计算机图形学大程序实验报告与程序说明文档**

课程名称： 计算机图形学

实验项目名称： 晴天雪原主题的风景图像生成

指导教师： 张宏鑫 完成时间： 2024.2.4

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **组长** | 张睿腾 | **专业** | **计算机科学与技术** | **学号** | 3210104884 |
| **组员** | 丛博义 | **专业** | **信息与计算科学** | **学号** | 3210102507 |
| **组员** | 段逸翔 | **专业** | **电子科学与技术** | **学号** | 3210105038 |
| **组员** | 江凯翔 | **专业** | **地理信息科学** | **学号** | 3210104620 |

**一、环境与配置：**

设备：Windows10，64位操作系统，基于x64的处理器

开发平台：Microsoft Visual Studio 2022 Community

Opengl API：GLFW3.3+GLEW7.0

其它辅助库：

Assimp库： 用于模型导入

GLM库： 用于辅助数学计算

IMGUI库： 提供图形界面框架API

GLUT库： 用于辅助部分简单内容的绘制

**二、主要代码文件及功能：**

**场景渲染模块（Layout.h）：**

该文件包括所有会出现在CG画面中的物体的绘制函数，及一小部分特殊物体的额外GUI信息，并为所有会出现在CG画面中的物体提供了与控制模块、GUI模块的接口。

**重要内容有：**

**天空盒（SkyBox）类：**初始化和绘制天空盒，天空盒是一个六面分别为不同材质的立方体，始终跟随人物相机。天空盒绘制时禁用深度缓冲区，因而不会遮挡物体且在任何位置看到的天空盒纹理都相同。

**地面（Terrain）类：**初始化和绘制地面，决定了相机漫游能到达的最低y坐标值。

**模型载入（ObjHandler）类：**用于载入模型文件，模型载入后其所有网格与材质信息与一同被封装进模型（Model）类的实例。模型载入代码的主要部分来自learnopengl.com教程的Model Loading章节。

**可操作物体（OperableLayout）类：**包括物体在CG场景中的各种属性，包括坐标、旋转角度、伸缩倍率等，同时是Model类实例的一个容器。对于载入的模型，在调用OperableLayout::Draw时调用其绘制函数。对于直接用opengl代码绘制的物体通过继承并重载虚函数OperableLayout::Draw进行绘制。OperableLayout类为场景中的物体提供了与控制模块的接口。**OperableLayout或其子类的一个实例对应场景中的一个物体类型**。

**被添加物体（PlayerAddedLayout）类：**是OperableLayout的子类。同时也是OperableLayout的一个容器。OperableLayout或其子类的一个实例就是场景中的一个物体类型，而**PlayerAddedLayout或其子类的一个实例直接对应场景中的一个物体**（天空盒与地面除外）。PlayerAddedLayout提供了场景中的物体与GUI模块和控制模块的接口。其子类PlayerAddedLight另外提供了场景中的发光物体与光源模块的接口。

**可编辑棱台（GeneralPrismCreator）类：**是OperableLayout的子类的子类。它以PlayerAddedLayout的子类PlayerAddedPrism作为容器。可以通过在GUI中调节绘制任意棱数（但不能超过200或小于3）的棱台与棱柱。

**场景物体表（std::map<int, void\*> AddedLayout）：**存储目前被添加到场景中的物体，同时为控制模块选中判定函数提供信息

**控制模块（Control.h）：**

该文件包含用于键盘控制和鼠标控制的函数，具体的按键作用会在第三部分说明

**主要内容有：**

**鼠标点击函数（MouseCtrl::MouseClickProc4GLFWLoop）：**会调用选中判定函数等。左击和右击都有功能，会在第三部分说明。

**选中判定函数（MouseCtrl::ProcessHit4GLFW）：**为所有物体（主要是无法通过函数gluPickMatrix建立拾取矩阵选中的物体，例如利用着色器渲染的导入模型）提供选中判定。通过PlayerAddedLayout存储的坐标信息和相机坐标信息，计算其与视线方向的夹角及与相机的距离实现，并非lab6中的拾取矩阵方式。

**鼠标滚轮函数（MouseCtrl::MouseScrollProc4GLFWLoop）：**通过调整视场角视线Zoom In和Zoom Out

**键盘输入处理函数（KeyboardCtrl::KeyProc4GLFWLoop）：**处理键盘输入

**控制状态（static int GLOBAL\_CONTROL\_STATUS）：**决定了键盘鼠标操作的对象，及GUI显示模式

**相机模块（Camera.h）：**

该文件包括相机（Camera类）

一个Camera类实例主要包括以下字段、成员函数：

相机坐标Position[3]

相机与视点的距离R，它同时间接决定了使用GUI放置物体时会放在面前多远，以及在点击是最远能点到多远范围内的物体

相机与视点的相对高度H，它充当俯仰角的作用

偏航角Yaw

Camera类的成员函数中比较重要的有：

**相机漫游函数（Camera::ProcessInput）：**根据键盘输入移动相机

**鼠标移动函数（Camera::SightMouseFollow）：**鼠标移动（非拖动）时，调整相机角度

**GUI模块（GUI.h）：**

该文件包含一个图形界面（GUI）类

该类主要包括以下内容：

可添加物体表（std::vector<std::pair<std::string, OperableLayout\*>> ObjList）：存储GUI按钮名称与对应OperableLayout实例的指针

GUI渲染函数（GUI::Draw）：用于渲染GUI，当鼠标在选中GUI中需要放置某个物体的按钮时，它会修改所需的控制模块信息，当再次用鼠标左键点击屏幕时就会放置对应的物体。

**光照模块（DualLight.h）：**

该文件包含一个光源（DualLight）类

该类主要作用如下：

包含一个光源列表（std::vector<PlayerAddedLight\*> LightList）：当通过鼠标左键点击添加了一个光源物体时，主循环会根据控制模块的信息将这个光源添加到光源列表中。

除了初始设置的主光源外，最多可以有七个额外的点光源参与光照渲染，超出这个数的光源物体不会参与光照渲染。添加的光源物体的光源属性可以在GUI中修改。

**截图功能（ScreenShot.h）：**

提供截图功能

其它工程文件

LoadTexture.h：用于加载材质图片

Shader.h：包括着色器（Shader）对象，提供将着色器代码编译成着色器程序的函数，其包含的成员变量主要有着色器描述符

utils.h、txts.h：一些与程序主干无关的杂项

**三、场景功能：**

**键盘操作方法：**

在任何状态下，按下Esc可退出程序，按下shift+p可截图，但e键每0.5s只能触发一次，截图每1s只能触发一次；截图保存路径为：解决方案路径/OpenGL/ScreenShots

在除了场景布置状态的三个状态下，按鼠标**右键**可显示/隐藏鼠标，但在按e打开GUI的同时会自动切换为显示鼠标

**初始进入时为漫游状态**，漫游状态下GUI无按键，但会显示“Press e to furnish the scene.”，此时的控制方式为：

wasd移动、space/z上下移动

鼠标移动（非拖到）改变视角

按e可打开GUI，并进入场景布置状态

鼠标**左键**成功选中物体，可进入物体放置状态，漫游状态下可以选中除天空盒与地面外的任何物体。（**注：选中判定算法使用计算视线夹角判定，故选中的是屏幕中间的物体而非鼠标位置所在的物体**，这一点与一些游戏（如Minecraft类似））

**在场景布置状态时**，视角不可变换，此时的控制方式为：

wasd移动、space/z上下移动

当**左键**点击中GUI中的按钮后，再**右键**点击屏幕可放置选中的物体

按e可回到漫游状态

**在物体放置状态**，GUI无按键，但会显示“Selecting xxx. Press e to edit.”，此时的控制 方式为：

相机位置不可变，但可以通过鼠标移动（非拖动）改变视角，此时只能移动被选中的物体

wasd移动、space/z上下移动

x/c放大缩小

shift+z/x/c可使其绕x/y/z轴旋转（但如果需要旋转物体到某个角度，直接在物体编辑模式下会更快）

左键点击屏幕可回到漫游状态

按e键可打开GUI，并进入物体编辑状态

**在物体编辑状态**，键盘控制与物体放置状态完全相同，但视角不可变换，控制方式为：

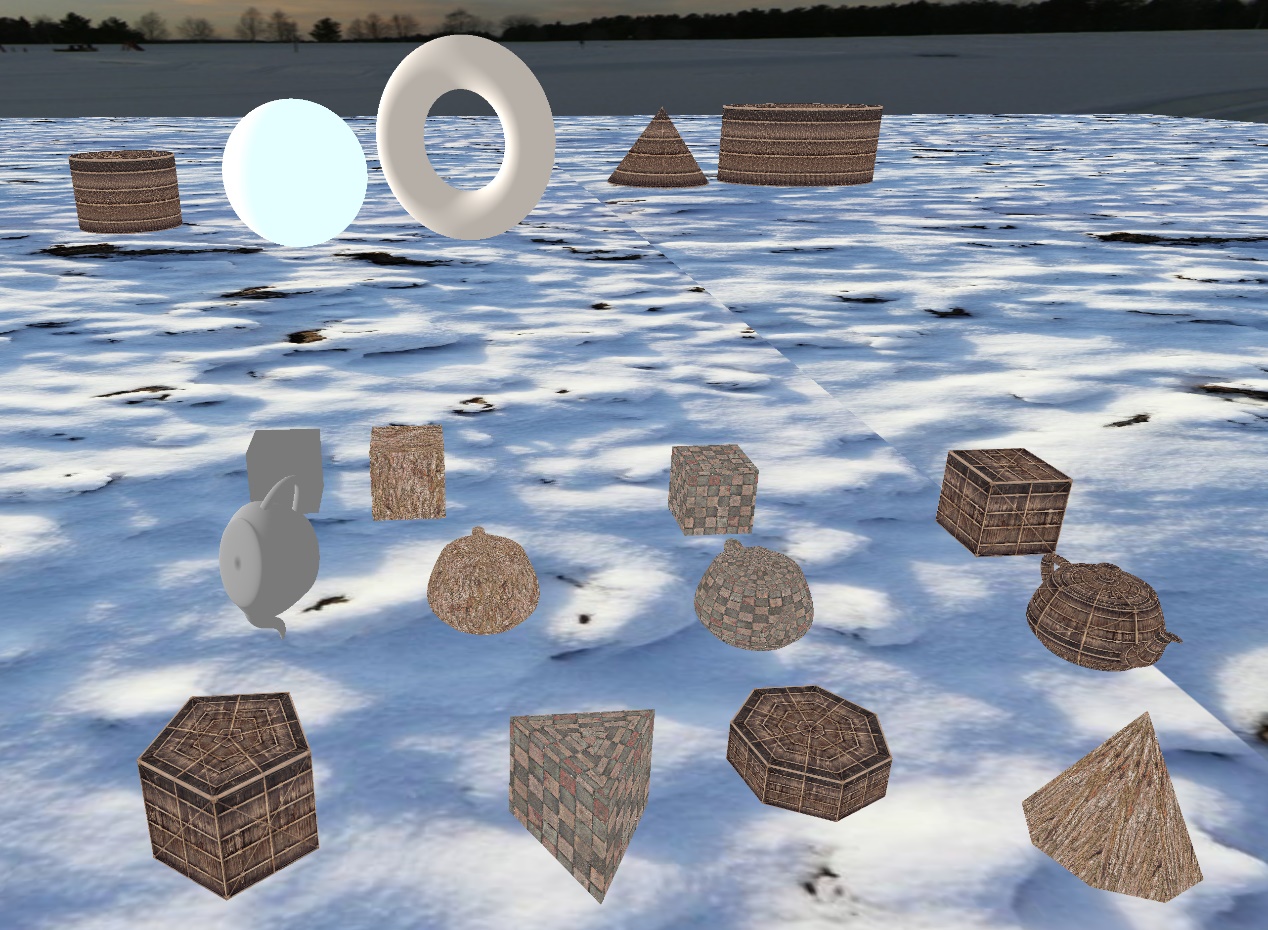
按e键可回到物体放置状态

物品编辑状态下，GUI的界面随不同种类的物体而不同，但至少包括以下几项：删除键、绕X、Y、Z轴的旋转速度及对应的清除键、沿X、Y、Z方向的伸缩变换因数及对应的重设键

**可添加的场景物体：**

**非载入模型的物体：**

**基本体素：立方体、球、圆柱、圆锥、多面棱柱、多面棱台**、**圆环**与glut库的**茶壶**。



上图下三排为立方体、茶壶、可编辑棱台。

立方体、球、圆柱、圆锥、圆环、茶壶都是通过GUI中的“Solidxxx”按钮添加。

由于实验要求中并没有说清楚多面棱柱、多面棱台要几条棱，因此我在程序中是以可编辑棱台（通过“Editable Prism”按钮添加）的形式实现的。

这些非载入模型的物体在物体编辑状态下都有以下几个功能：



删除、更换材质（这些物体初始状态都默认无材质，有三种材质可供选择，球和圆环除外）、绕x/y/z轴的旋转动画速度、沿x/y/z方向拉伸倍率、表面光照反射参数，及第三部分提到的键盘操作（移动、旋转、放大缩小）。

而可编辑棱台则多出几个按键



多出来的几个按键为：加/减棱数、一键设定为圆台（棱数100）、一件设定为立方体（棱数4，此处因棱数恰好为4未显示）、调整高、调整上下底面半径的比。

因此可编辑棱台可以任意加减棱数（至少3，至多200），同时可以调节高与底面半径的比和上底面和下地面的半径的比（这里“半径”指的是从底面的中心到底面端点的距离）。

实际上可以通过这个物体实现圆柱、圆台、圆锥、多面棱柱、多面棱台。（当棱数为100时，其效果与上面直接添加的圆柱完全相同，因为绘制圆柱的代码就是从这里复制过去的。）

**载入的模型：**

**光源物体：篝火（光源开/关状态为两个材质）、灯笼**



光源物体在物体编辑状态下都有以下几个操作功能



删除、光源开/关、绕x/y/z轴的旋转动画速度、沿x/y/z方向拉伸倍率、调节光源参数，及第三部分提到的键盘操作（移动、旋转、放大缩小）。另外，篝火通过“light on”/“light off”按钮点燃、熄灭有两种不同的材质。不难看出上图中的茶壶明显右侧比左侧更亮。

**三种小屋**



两种原木堆



**背包、松树、雪人**



**山（非常巨大，慎用）**

****

普通物体在物体编辑状态下只有以下几个基本功能



删除、光源开/关、绕x/y/z轴的旋转动画速度、沿x/y/z方向拉伸倍率，及第三部分提到的键盘操作（移动、旋转、放大缩小）。

**四、其它信息**

Github仓库地址：<https://github.com/HotoriArashiyama/CG-Project/tree/Final>

分工信息：

张睿腾：场景渲染模块、相机模块、控制模块（主要是相机相关及鼠标和键盘操作场景物体的部分）、截图功能、代码整合、报告撰写

段逸翔：GUI模块、控制模块（主要是与GUI相关的部分）、部分简单物体绘制

江凯翔：光照模块（包括前文未提到的复杂光照着色器渲染，但很遗憾没有整合成功）、天空盒

丛博义：组织分工、搜集.obj模型

事实上，在这个大工程的编写过程中，不同的分工模块边界其实并没有那么清晰，例如因为需要测试自己的代码，段逸翔和张睿腾都写了控制模块与相机模块的一些基本功能，而这两个模块最终采用的都是我（张睿腾）的版本，这主要是因为我的这两部分在代码整合前就已经和场景渲染模块建立了联系，整合时乃至后续添加针对.obj载入模型的控制和选中功能时也更方便。再如江凯翔的光照模块和张睿腾的场景渲染模块都包含了天空盒的实现，最终采用后者的版本也是如此。