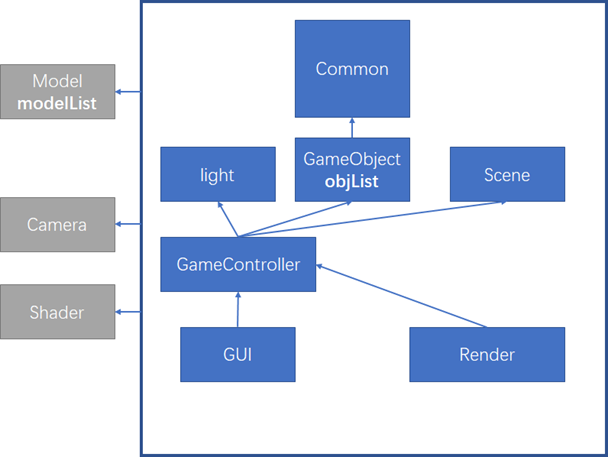
**Github地址：<https://github.com/KoonanHyakukeiZu/2020-ZJUCG-Final-Project>**

**演示视频地址：<https://www.bilibili.com/video/BV1Hy4y1p76d>（求三连QAQ）**

**整体设计思路图**

****

使用glfw+glad进行开发

**1.游戏控制器（GameController）类：**

单例模式

重要成员变量：

主相机mainCamera

创造模式原始位置相机oriCreatingCamera

重要成员函数：

initGameController(GLFWwindow\* window) //为GLFW窗口配置回调函数

updateGameController(GLFWwindow\* window) //在GLFW主渲染循环中调用，用于处理逐帧操作

changeGameModeTo(GameMode newmode) //由GUI类对象调用，进行游戏状态转换

revertGameMode() //由GUI类对象调用，游戏状态将返回上一个状态

**2.图形界面类（GUI）：**

单例模式

重要成员函数：

initEnv(GLFWwindow\* window) //初始化ImGui环境

updateModelTextures(Shader& shader) //在GLFW主渲染循环中调用，用于绘制GUI

**3.相机（Camera）类对象**

重要成员变量:

相机位置Positoin

相机前朝向向量Front

相机上朝向向量Up

相机右朝向向量Right

相机俯仰角Pitch

相机摇动角Yaw

重要成员函数：

ProcessKeyboard(float deltaTime, Camera\_Movement direction = NONE) //根据键盘输入的参数进行变换

ProcessMouseMovement(float xoffset, float yoffset) //根据鼠标移动输入的参数进行变换

ProcessMouseScroll(Camera\_Scroll scrollAct, float yoffset) //根据鼠标滚轮输入的参数进行变换

**4.场景(Scene)类场景的对象：**

重要成员变量：

含有所有地形类(Terrain)的vector

含有所有水面类(Water)的vector

天空盒类(Skybox)

重要成员函数：

Draw(float deltaTime, Camera& cam, glm::vec4 clippling\_plane, bool draw\_water, bool draw\_shadow = false)//输入时间、相机、裁剪平面进行场景绘制

float getTerrainHeight(float x, float z)//根据xz坐标获取地面高度

**5.地形(Terrain) 类的对象**：绘制地面，可从高度图生成（当前）或者生成随机地形

**6.水面 (Water) 类的对象：**绘制水面

**7.水面帧缓冲 (Water\_Frame\_Buffer) 类的对象：**负责管理水面渲染的帧缓冲

**8.灯光(Light)类的对象：**设置全局平行光和点光源，并且负责为各个场景和物体元素的Shader中的灯光

**9.天空盒(Skybox) 类的对象：**绘制天空盒

**10.阴影(Shadow) 类的对象：**负责管理阴影的深度缓冲

**11.渲染(Render)类的对象：**

重要成员变量：

场景对象、灯光对象、水面帧缓冲对象、拾取缓冲对象、阴影缓冲对象的引用

重要成员函数：

InitLighting(Shader& shader)//初始化某个Shader中的灯光变量

DrawReflection(Shader& modelShader)//绘制水面反射材质

DrawRefraction(Shader& modelShader) //绘制水面折射材质

DrawAll(Shader& pickingShader,Shader& modelShader, Shader& shadowShader)//绘制场景和所有实体

**12.投射拾取（MousePicker）类的对象：**

从鼠标点击位置投射光线进行地面位置判断

**13.帧缓冲拾取（MousePicker）类的对象：**

从鼠标点击位置进行物体拾取判断

**14.视频录制（VideoRecord）类的对象：**

抓取当前默认帧编码录制成为视频

**15.模型(model)类的构建：**

重要成员变量：

含有图形纹理的vector;

含有所有网格的vector;

模型文件（.obj模型文件以及.mtl材质文件）储存目录

重要成员函数：

void Draw(Shader& shader)：利用着色器画出所有网格。

void loadModel(string const& path)：从模型文件的目录读取文件及纹理载入模型。

void processNode(aiNode\* node, const aiScene\* scene)：以递归方式处理节点。

Mesh processMesh(aiMesh\* mesh, const aiScene\* scene)：处理网格。

vector<Texture> loadMaterialTextures(aiMaterial\* mat, aiTextureType type,

string typeName)：检查并加载材质纹理

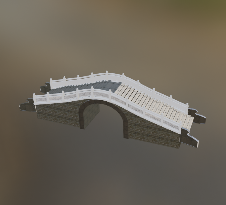
**场景内模型建模：**

①基本体素

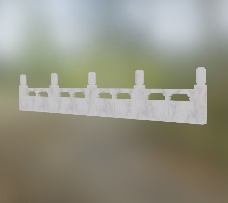
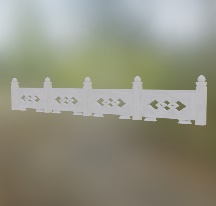
正方体、圆锥、圆柱、球体、六棱柱、六棱台六种基本体素，初始纹理为纯灰，为没肿模型提供了gold、wood、marble、glass和texture五种可选材质纹理，可以通过选择自由切换。

②场景内建筑

1）桥梁（三种，以中国古代桥梁风格为蓝图）

2）围栏（三种材质的围栏，木质、石质、花岗岩）

3）Memorial Gate



4)树木、灌木丛

5）岩石



6）寺庙（两种）

7）木舟



8）建筑



为了和模型类兼容，每个模型的每个部分均设置了材质和纹理。