

# 计算机图形学期末大作业选题报告

项目名称	基于 OpenGL 的体素化三维游戏 -- MyCraft
小组成员姓名	梁恩睿
学号	23336126
专业	计算机科学与技术
日期	2025 年 11 月 24 日

## 一、项目背景与选题意义

受沙盒游戏《Minecraft》的启发，体素作为一种离散化的三维数据表达方式，能够直观地构建可交互的虚拟世界。本项目旨在利用本学期所学的计算机图形学基础理论，不依赖成熟的游戏引擎，而是基于 OpenGL 底层图形库，采用基于 CMake 的现代 C++ 构建流程与依赖管理，从零构建一个类 Minecraft 体素游戏。

本项目的核心目标是通过实现一个具备方块放置与移除、水面动态波动、过程式地形生成以及基础全局光照效果的体素世界。这将把课程中抽象的渲染管线、坐标变换、光照模型等知识点进行实践与验证。

## 二、结合课程内容的知识点规划

本项目将紧密围绕本学期《计算机图形学》课程的教学大纲进行设计，计划将各个阶段的学习内容应用到系统的不同模块中：

### 1. 基础渲染管线与图元绘制（对应课程第 2-6 周）

项目将放弃古老的固定管线，完全采用 Core Profile 模式进行开发。在项目初期，我计划通过手动管理顶点数组对象（VAO）和顶点缓冲对象（VBO），实现最基础的立方体绘制。这将直接应用到课程中关于光栅化原理和 OpenGL 状态机的知识，编写综合性的 GLSL 着色器来控制顶点的输出颜色与材质混合，验证渲染管线的基本流程。

### 2. 几何变换与视角系统（对应课程第 7-13 周）

为了实现“漫游”功能，项目中需要构建完整的坐标变换矩阵系统。我计划应用课程中讲解的模型变换、视图变换和投影变换矩阵，将体素的世界坐标映射到屏幕空间。同时，通过监听键盘和鼠标输入，动态更新摄像机的欧拉角，从而改变观察矩阵，实现第一人称视角的自由移动。这一部分将是验证几何变换与视角理论的核心环节。

### 3. 进阶渲染与扩展课题（对应课程第 14-17 周）

在完成基础场景搭建后，我计划尝试进阶技术：

**纹理映射：**加载外部高清材质包，并将纹理坐标传递给片段着色器，给立方体贴上草地、泥土或石头贴图。

**地形生成：**为了模拟自然环境，利用噪声函数，生成自然的起伏地貌。

**光照与环境：**探索基础的全局光照模型，实现基于 **Phong 模型** 的昼夜交替光照，结合环境光遮蔽与雾化效果，体现场景的体积感与沉浸感。

### 三、系统设计与实施方案

#### 3.1 总体架构规划

为了保证代码的可维护性，系统拟采用面向对象的设计思想。主要模块规划如下：

- **窗口管理**：使用 **GLFW** 库处理窗口创建和输入响应，使用 **GLAD** 加载 OpenGL 函数指针，并集成 **ImGui** 用于调试与 UI 展示。
- **摄像机模块 (Camera)**：封装视图矩阵计算逻辑，提供第一人称漫游 (Spectator Mode) 的接口。
- **世界管理模块 (World)**：考虑到全地图渲染的性能开销，计划采用“分区块”方式组织数据。将世界划分为若干个 16x16x128 的小区块，仅对摄像机附近的区块进行网格构建和渲染。

#### 3.2 关键技术路线

- **数据结构设计**：在内存中，拟使用一维数组存储每个区块内的方块类型。为了优化渲染性能，将在生成网格时实施**面剔除策略**：仅当方块面与空气或透明物体接触时才生成顶点。此外，引入脏标记机制，确保仅在数据变动时重建网格，避免每帧重复计算。
- **地形生成逻辑**：为了模拟自然环境，利用噪声函数的连续性特点生成高度图。计划采用多层噪声叠加来模拟平原与山脉的过渡，根据高度分层填充基岩、石头、泥土及表层草地，实现过程式的地形生成。
- **交互逻辑**：系统拟实现精准的方块破坏与放置功能。这需要涉及射线检测算法。计划采用 DDA（数字微分分析）步进算法，精确计算视线与体素网格的交点，确定玩家指向的方块及其法线方向，从而实现方块的高亮提示与修改。

### 四、预期成果

通过本次大作业，我预期完成一个可运行的 C++ 程序。该程序能够展示一个由体素构成的三维世界，具备以下特征：

1. **自由漫游**：用户可以通过键盘和鼠标在无限延伸的场景中自由移动与观察。
2. **动态环境**：场景包含由算法生成的自然地形、昼夜交替的光照变化以及程序化的云层渲染。
3. **交互功能**：支持玩家对方块进行破坏与放置，且伴有视觉反馈。
4. **性能表现**：帧率在常规笔记本电脑上保持流畅（60 FPS+）。

这不仅是对本学期计算机图形学知识的一次全面总结，也是对现代 C++ 工程实践能力的一次锻炼。

### 五、项目开发进度安排

根据项目当前的开发状态及后续规划，将剩余工作划分为五个阶段进行。

#### 第 10 周：核心渲染框架与摄像机系统 (Current Phase)

**本周目标**：搭建稳固的 OpenGL 上下文环境，实现三维空间内的自由观察。**具体实施计划**：

1. **基础架构搭建**：使用 CMake 集成 GLFW、GLAD 与 GLM 库。编写 Shader 类，封装 GLSL 编译链接流程，建立 Core Profile 开发环境。
2. **摄像机系统**：基于欧拉角原理实现 FPS 风格摄像机，构建矩阵，实现场景漫游。

#### 第 11 周：数据结构设计与区块管理

**本周目标：**实现体素数据的有效存储与基于区块的网格构建。**具体实施计划：**

1. **区块类设计：**采用一维数组存储  $16 \times 128 \times 16$  规格体素 ID，利用哈希表管理区块索引与动态加载。
2. **网格构建算法：**实现\*\*面剔除 (Face Culling)\*\* 算法，仅生成可见面顶点，大幅减少无效三角形。
3. **方块注册表：**集中管理方块属性（透明度、固体属性），为渲染和物理逻辑提供数据支持。

## 第 12 周：纹理系统优化与材质表现

**本周目标：**解决海量方块的纹理映射问题，提升视觉表现力。**具体实施计划：**

1. **纹理图集：**加载外部材质包，动态拼接大纹理图集。
2. **特殊材质渲染：**在 Shader 中实现透明物体（水、玻璃）与不透明物体的分层渲染逻辑。为水面方块增加动画，模拟水面动态。

## 第 13 周：过程式地形生成与环境系统

**本周目标：**利用算法自动生成自然地形，并构建拟真的环境氛围。**具体实施计划：**

1. **地形生成算法：**引入 **FBM（分形布朗运动）** 与 **Ridge Noise** 叠加算法，生成具有自然起伏感的高度图，并根据高度自动填充生物群系（如雪山、草地）。
2. **动态环境系统：**实现昼夜交替。在 Shader 中动态计算太阳向量，对天空颜色与环境光进行线性插值。
3. **程序化云层：**不使用贴图盒子，而是直接在片段着色器中通过噪声函数计算体积云效果，增强天空的真实感。

## 第 15 周：交互逻辑实现与系统调试

**本周目标：**实现用户与虚拟世界的交互，进行性能优化与演示准备。**具体实施计划：**

1. **光线投射交互：**实现 **DDA（数字微分分析）** 算法，用于精确计算视线与方块的交点。基于此实现方块的破坏与放置功能，并添加高亮选中框。
2. **性能优化：**完善“脏标记 (Dirty Flag)”机制与网格重建队列，防止玩家快速移动或大规模破坏方块时造成主线程卡顿。
3. **UI 与调试：**集成 **ImGui** 库，制作调试面板，实时监控 FPS、坐标、切换渲染模式（线框/实体）及调整环境参数。
4. **最终测试：**进行测试，debug，录制演示视频，撰写结课报告。