**Minecraft组中期总结**

周寒靖、郑昱笙、尹浩霖

一、项目描述

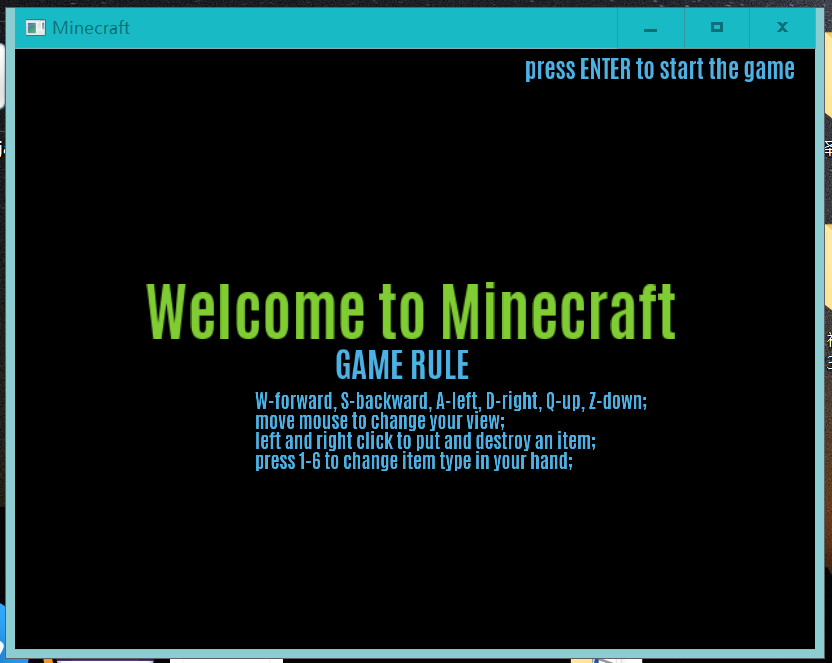
参考3D沙盒游戏minecraft，做一个简易版minecraft。玩家根据第一视角可在生成的有限随机地图中随意移动，损毁方块或放置方块，搭建属于自己的方块世界。

目前进度是第二轮迭代接近结束，基本目标已经接近完成；

地址：

二、部分效果图

开始界面，包含游戏提示等内容：



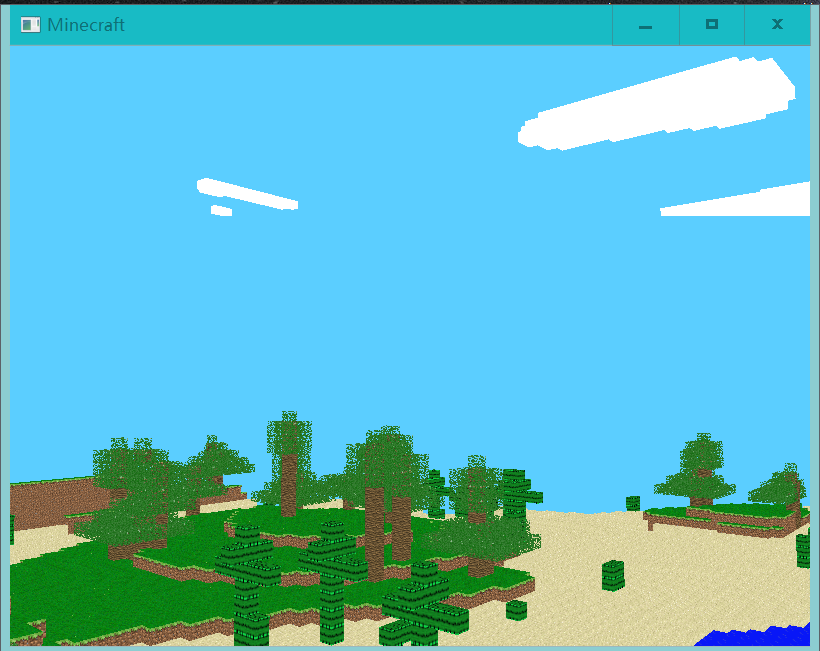
（1）实现基本地形生成，包含多种元素和地形，如山地、海洋、沙滩、树木、仙人掌等；

地图可以根据摄像机的位置进行动态生成和加载；

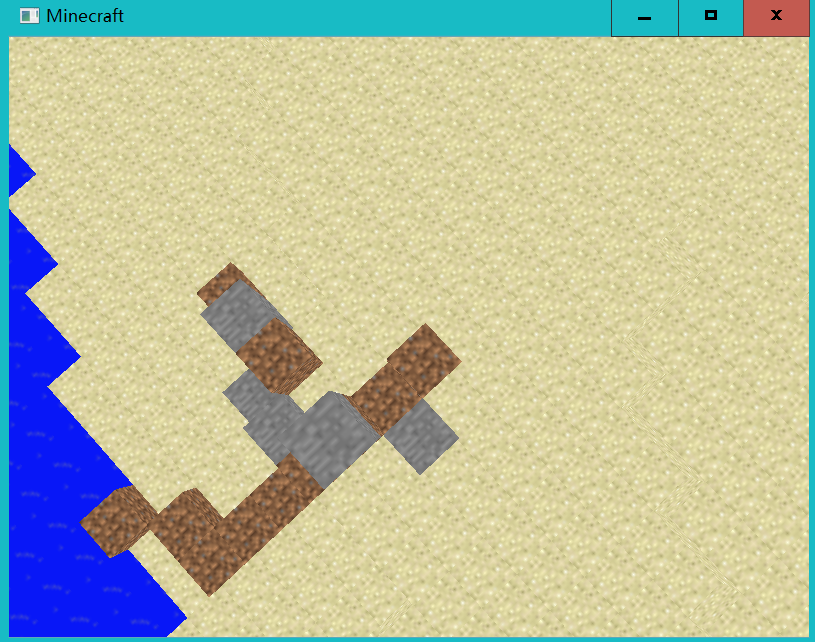


（2）实现摄像机漫游；

可以通过方向键控制摄像机前后左右上下移动，并有碰撞检测保证摄像机不会进入方块中；



（3）实现方块的放置与破坏；

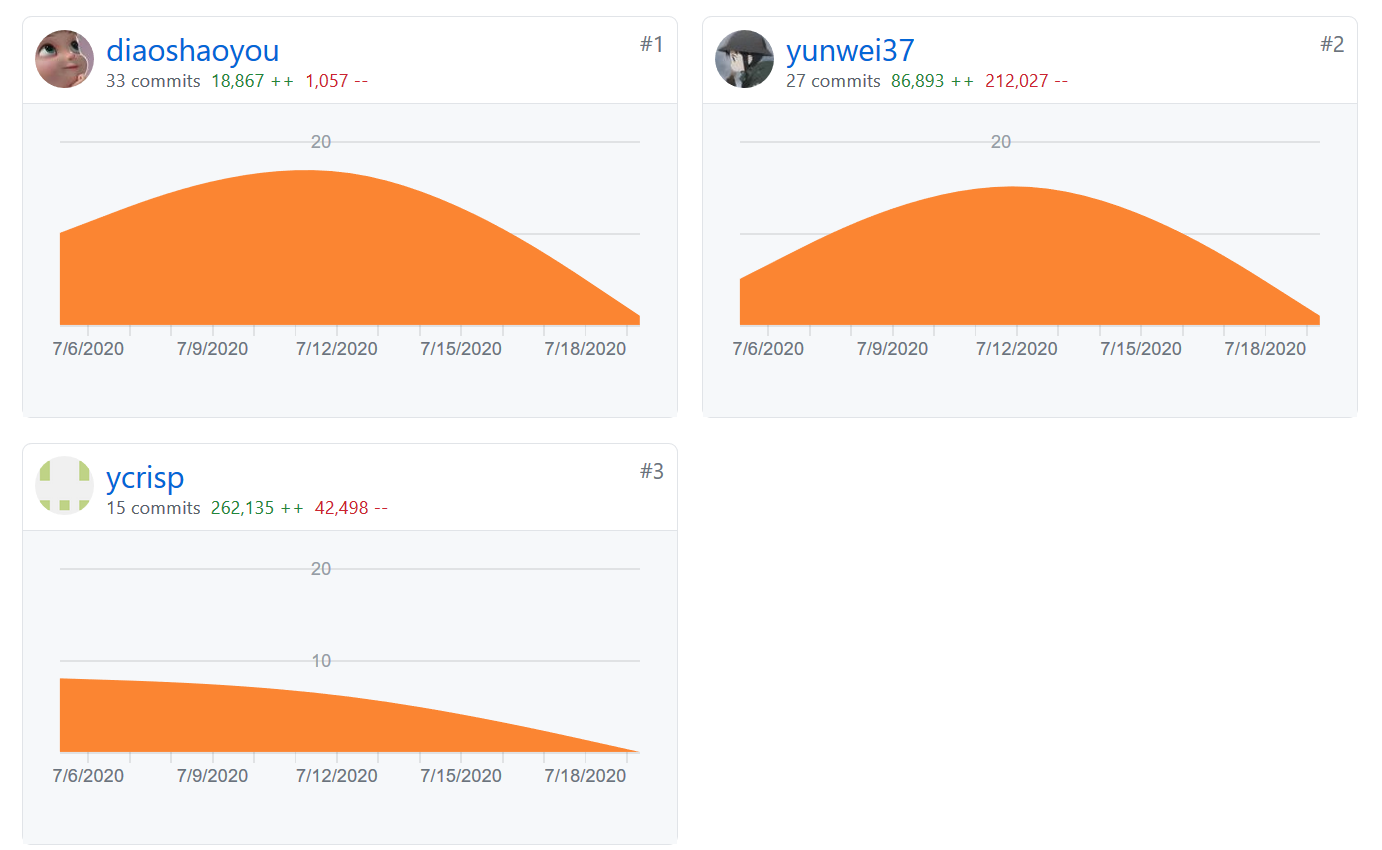


可以通过左键放置方块，右键破坏方块，并可以通过数字键选择方块类型；

三、协作情况

成员之间通过github进行协作开发，配合紧密；

Contributors图谱：



四、技术难点及克服；

1、地图的动态生成算法，以及存储渲染问题：

我们查阅了相关资料，根据原版的生成算法，采用二维perlin噪声实现了一个简易版的地形生成算法，该算法在seed确定的情况下，仅仅与x y位置有关，与生成顺序等无关；在噪声的基础上，我们通过多个步骤，包含一系列的采样、插值平滑，根据地形高度进行分层处理，采用附加噪声生成树木等，完成最终的地形生成；

由于噪声仅仅和地理位置相关，在位置确定的情况下地形一定相同，因此可以通过噪声的多次生成实现地形的可重入性，生成后方块的存储仅作为渲染的缓存。我们将渲染后的方块存储在 chunk 类的数组三维中。

2、无限地形

这里采用根据摄像机位置来动态生成和加载方块的策略；即只渲染和存储摄像机周围的部分chunk，当摄像机移动的时候

2、放置和破坏方块：

放置和破坏需要涉及openGL中的世界坐标和map中的三维坐标的转换(已达成)，另鼠标的屏幕坐标没法转成map的坐标，导致项目目前的用户体验不太好，只能根据用户的位置进行放置删除方块；

3、方块的破坏特效：

目前实现的特效生成的由于游戏是3d环境始终有坐标bug存在，并且特效的渲染和地图的渲染总是会有莫名其妙的冲突；这个问题需要进行更深入的挖掘，目前暂未想到良好的解决方案。

4、渲染速度问题：

渲染加载速度可以从两个方向解决问题，一方面是减少渲染物体数量，一方面是采用GPU加速，需要使用openGL缓存功能，目前还未完全实现。

这里采用了一个简单的方法，对于不需要渲染（即六个面都被其他方块覆盖的方块）进行标记，在渲染的时候跳过这些方块。

5、还存在部分层次之间的耦合情况

在下一个阶段中，我们需要对代码进行部分重构解决相关问题。

五：总体心得

这次项目起源于我们在github上面发现的一个很好玩的repo：<https://github.com/Hopson97/MineCraft-One-Week-Challenge> 在七天写一个简单的minecraft出来，有不少挑战性，又在力所能及的范围内，因此就希望完成一个类似的project。由于对图形学背景知识不够熟悉，我们也花费了不少时间在openGL的学习上面，但更重要的还是对项目管理、设计模式有了一个良好的认识和初步的实践，包括用github进行版本控制、协作开发，软件的mvvm模式，敏捷开发流程、迭代等等。

六：个人心得

尹浩霖：

这次project我们组做的是我的世界游戏的简略版本，过程中我对项目管理有了初步的认识，包括用github进行版本控制，与队友协作开发和软件的mvvm模式，除此之外，为了完成软件设计，我们自学了opengl，这个过程中让我对opengl基础有了掌握，完成了计算机图形学的入门。在几轮迭代中，这些技能和知识的掌握越来越熟练，这次的项目合作让我体会到了与以前小组合作完全不同的感受和效率，也提升了我的c++代码水平，让我受益匪浅。

周寒靖

这次大作业的难点有二：图形学知识和MVVM模式分工的划定。对于前者，在项目前期我们狂补了openGL官方教程，终于补充了必备的图形学知识，为项目的打造奠定基础。而对于后者，由于我们第一次尝试MVVM模式联合开发，不太熟悉，所以第一轮迭代并未很严格地按照这个模式分工。经过直播展示和老师提建议后，我们迅速做出调整，后面的工作都按照MVVM模式来分工，较大提高了团队效率。目前我们的工程已接近尾声，基础功能具备，细节优化尚未完成，我在这一次次迭代开发中获益良多。

郑昱笙

这次小学期的开发让我深刻理解到了现代项目管理工具与良好的设计模式的重要性，在MVVM模式模式中对于各层有个清晰的划分，对于项目的实际开发能起到事半功倍的效果；同时采用git协作开发，也能清晰地看到团队的进度，完成一次次迭代，同时对于每一个模块每一行代码也都是可追溯的，遇到问题也可以回退到之前的版本。另外，自己对于图形学的知识也有一些欠缺，这次项目也让自己更好地了解和使用了现代openGL。总之，这次工程实践收获还是非常丰富的。