

项目编号: T030SRIP12009

# 上海交通大学

## 暑期科研见习岗位

### 总结报告

岗位名称: 基于简单图形引擎的交互式程序设计

见习时间: 2023 年 6 月 19 日 至 2023 年 9 月 8 日

学院(系): 电子信息与电气工程学院

专 业: 信息工程

学生姓名: 赵楷越 学号: 522031910803

2023 年 9 月 8 日

## 1. 见习目的

根据 <https://github.com/OneLoneCoder/olcPixelGameEngine/wiki> 网站中的资料，学习开发者 javidx9 的视频，配置环境，并熟悉开发环境，接着利用基于 C++ 的 olcPixelGameEngine 框架从零创建一个游戏或者是可视化的交互界面应用程序。在此过程中，体会基于简单图形引擎的交互式程序设计项目编写的过程并锻炼代码实现的能力。

## 2. 见习任务

见习主要任务分为必选任务和扩展任务。必选任务为制作一个 Battlefield 战棋游戏。扩展任务可从图形化数据结构，分布式算法模拟 Lamport 时钟，模拟电路实现，进阶战棋游戏中自主选择，也可根据自己的兴趣自由寻找自己喜欢的课题进行自主实现。

### 2.1 任务一 Battlefield 战棋游戏

Battlefield 是一个回合制战棋类游戏。在大小为  $M \times N$  的战场上，玩家 A 和玩家 B 交替操作自己控制的单位移动和攻击对方单位，如果一方的单位全部被消灭，那么对方将获胜。任务要求实现五种不同的地形和四种不同的单位，并按要求实现其攻击方式和技能释放的效果。

### 2.2 任务二 图形化数据结构

选择一类特定的数据结构，如链表、二叉树、优先队列等，设计图形化展示数据并通过鼠标拖拽等方式实现数据结构的基本操作。例如，通过鼠标拖拽链表节点的指针更改链表数据结构、通过鼠标拖拽实现树结构和图结构的调整等。

### 2.3 任务三 Mini-Racing 赛车模拟

此项任务为从自己的兴趣出发设定的额外小任务。目标是学习利用 olcPixelGameEngine 扩展库中的 Camera2D 库实现一个迷你赛车的小游戏初步模拟。预计效果能实现小车随运动方向自由转动，小车和赛道的碰撞检测和实时跟随小车移动变换的自由小地图镜头实现。

### 3. 见习主要工作内容与成果

#### 3.1 预备内容

首先，进行对 olcPixelGameEngine 的环境配置和理解。

olcPixelGameEngine 是一个用于创建游戏和原型的 C++ 框架。它可以在 Windows, Linux 和 Mac 多种操作系统环境中编译运行。它也不需要任何外部依赖，只需要一个现代的 C++ 编译器。它可以让你快速地开发原型和游戏。通过创建一个窗口，并快速地绘制到该窗口来实现这一点。在其中，屏幕被认为是一个二维数组的像素，因此它也提供了一些基本的绘图工具来操作屏幕。此外，它也可以实时感知键盘和鼠标的输入。

学习对应视频后，本次任务中的环境配置选择利用 Visual Studio 2022 进行项目的编写和搭建。

#### 3.2 Battlefield 战棋游戏

根据要求，首先实现了从外部控制台读取相应命令行代码的形式进行地形和单位数据的读取，每行中前两个数字代表对应元素的坐标，后位的字母相应表示不同的地形种类，单位种类和单位所属玩家。

图 1 为控制台输入预设地图的界面：

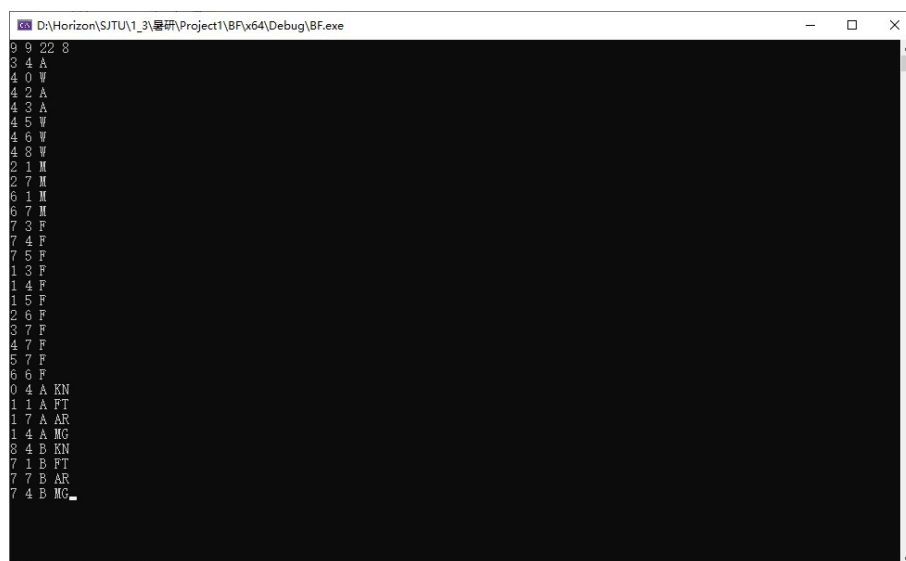


图 1 控制台输入预设地图

接着，先后进行了游戏 UI 及信息界面的设计，地图，单位，地形的绘制，和不同单位，地形的效果实现，同时使其图像和棋盘大小能自适应不同的窗口规格。对于绘制部分，先利用 DrawLine 函数画出了整个棋盘战场，利用 DrawString 函数显示战场的实时信息，再使用了 Sprite 类型和指针的搭配使用，实现了单位和地形的绘制。

其中，不同地形种类为：平原，森林，高山，深渊，洪水。

不同单位种类为：脚男，骑士，弓兵，法师。

图 2 为初始化后的地图预览：

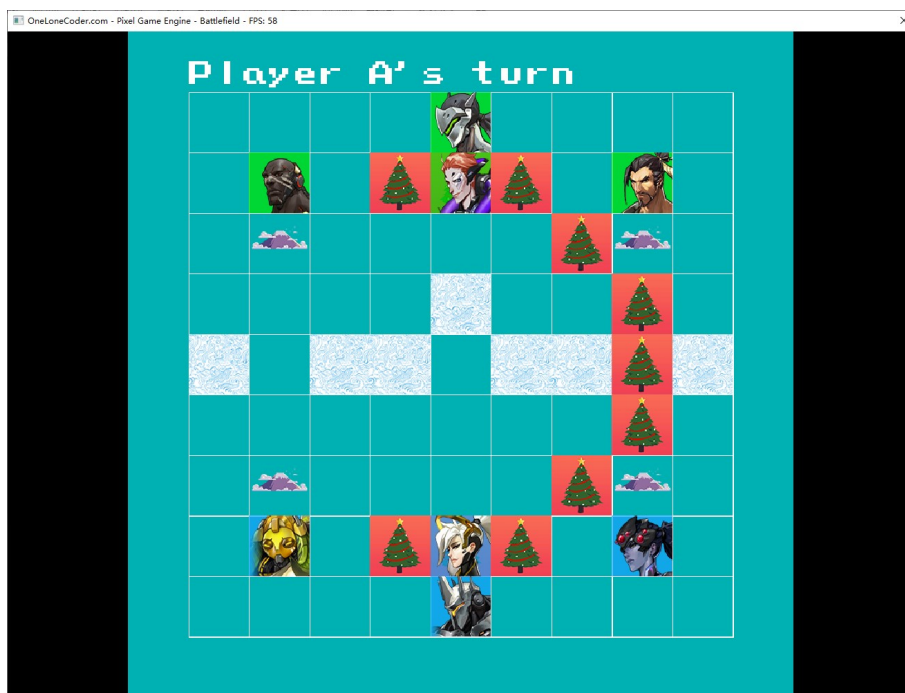


图 2 地图初始化预览

而到了单位部分，编写了一个 units 类作为借口，来存储与棋盘单位有关的信息并进行对应操作。其中包含单位种类，单位移动力，单位坐标，单位所属玩家等关键信息。同时也编写了 field 类，利用指针存储了战场和单位的对应信息。

对于单位的选中和移动，在 engine.cpp 中编写了 mouse\_button\_is\_valid 函数进行不同状态下的判断，并用 current\_state 变量存储当前单位不同选中状态信息，以便于实现单位的操作。

最终，程序效果为使用鼠标左键点击选中单位，程序会自动高亮显示单位并在右上角显示单位在本回合剩余的移动力，再用右键选中不同地形，单位便消耗了对应的移动力并进行移动。

图 3 为单位移动功能的演示：

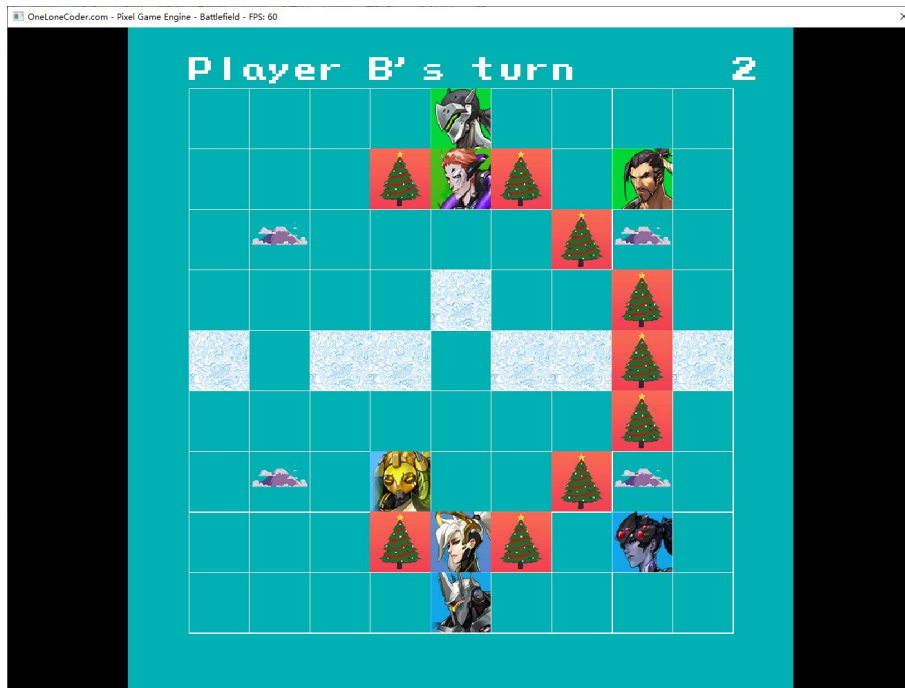


图 3 单位移动功能演示

当选中单位时，右键点击攻击范围内的其他单位（无论友方或敌方单位）便可进行单位攻击操作，击杀单位后，程序会利用 Decal 绘制功能显示击杀特效。

图 4 为近战击杀效果的演示和特效：



图 4 近战击杀效果演示

然后，也实现了法师单位的两个特殊技能的实现：地震术和火球术。当选中法师单位后，按下键盘 E 键和 F 键便能切换至不同法术的释放状态，再右键点击法术施放范围和方向便可释放法术，按 ctrl 返回移动模式。在 units 类的派生 mage 类中分别实现了其对应功能。

图 5 和图 6 分别展示了地震术和火球术的效果：

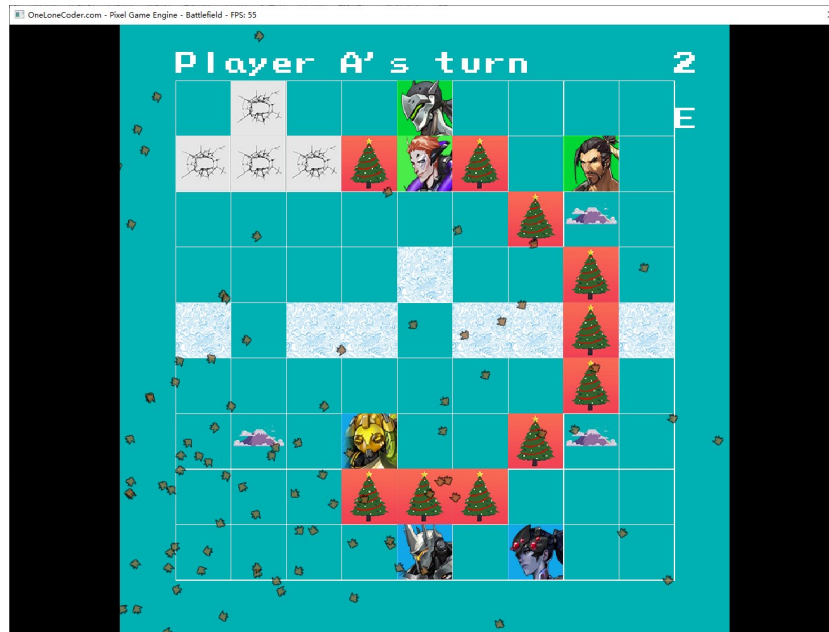


图 5 地震术效果演示

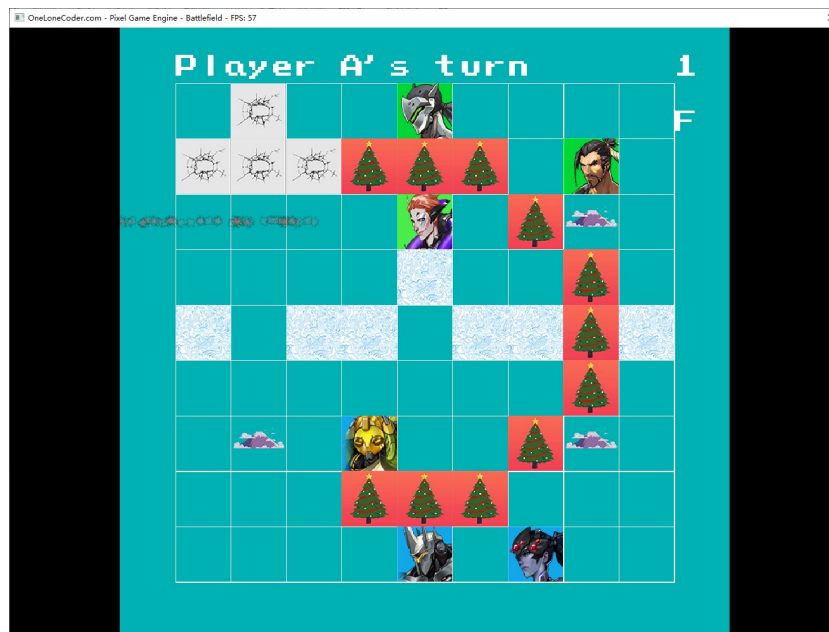


图 6 火球术效果演示

同时，基于深度优先搜索的连通性判断，实现了火球术连锁烧毁树林的效果，以及洪水自动填满附近深渊的效果。特殊效果也均为利用 Decal 绘制函数进行绘制。

图 7 和图 8 分别为火球术连锁烧毁树林和洪水填满深渊的效果演示：



图 7 火球术连锁烧毁树林效果演示



图 8 洪水至深渊的连通性演示



最终，当一方玩家胜利时，切换至如图 9 所示的胜利界面：

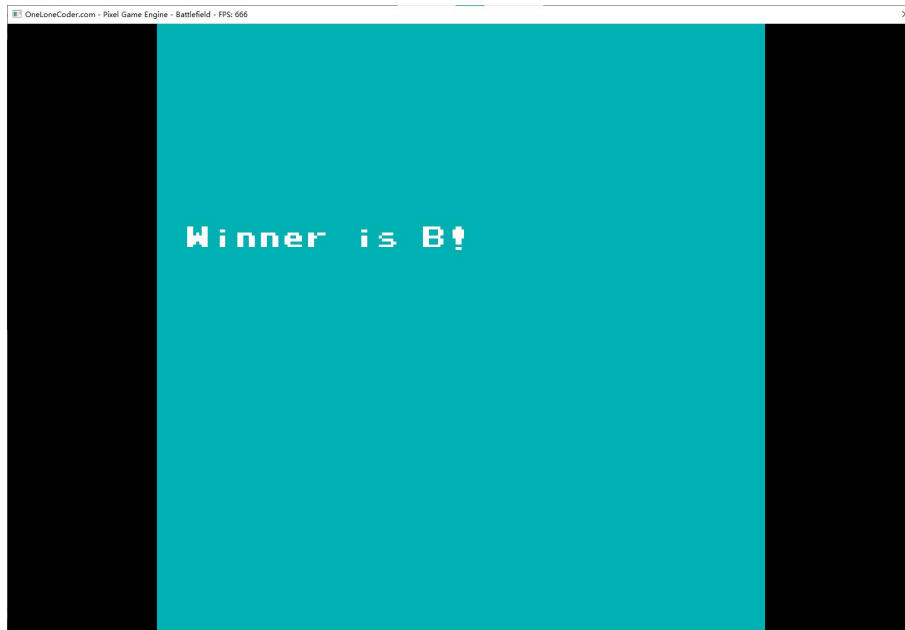


图 9 玩家胜利界面

### 3.3 图形化数据结构

对于图形化数据结构的任务，采用了菜单+分界面式的设计思想，首先，绘制了一页目录，鼠标移动选中对应栏便可进入对应的图形化数据结构模拟。为了稳定帧率，在程序中实时维护当前鼠标位置和四块按钮区域边界选中效果的绘制，以减少维护的像素区域以达到稳定帧率的目的。同时也做到了自适应字体大小及界面大小的内部设计。菜单如图 10 所示：



图 10 开始菜单



当选中 Linklist 块后，进入单链表模拟的页面。利用 stl 库中的 list 链表存储并实时维护自己编写的 Linklistnode 类型，便于管理和绘制。当按下 N 键时，会在界面上随机位置生成单链表的节点，单链表节点的对应编号初始定义为其生成的顺序编号。单链表随机生成的效果如图 11 所示：

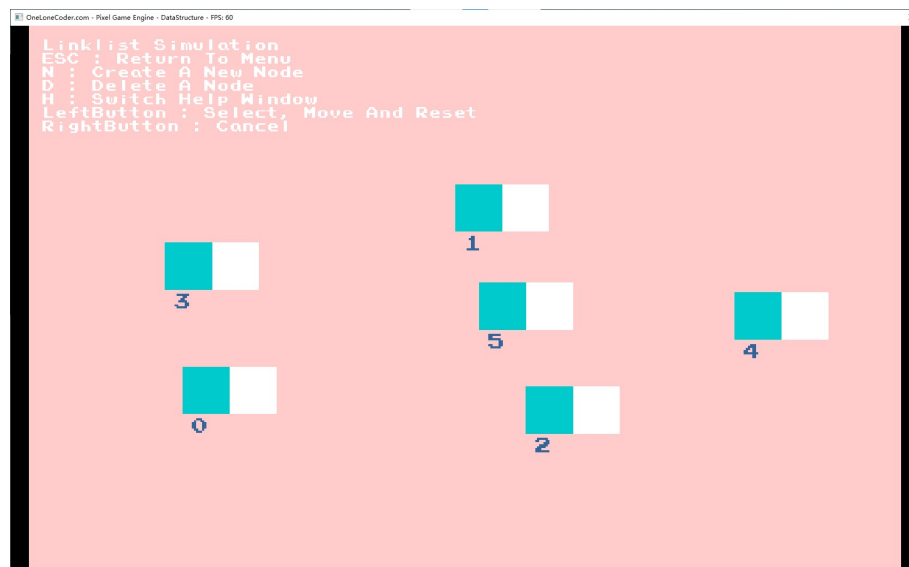


图 11 单链表随机节点生成

操作能灵活响应键盘和鼠标的输入。其各自对应操作方式为：

移动：按住鼠标左键拖拽节点的深色区域移动鼠标进行移动节点操作。

连接：鼠标左键点击浅色区域开始连接节点，再右键点击另一块的深色节点进行连接。

输入：鼠标左键点击节点深色区域时，屏幕右上角显示“Entering...”字样便能进行该节点的字符串类型 data 变量的输入和修改操作。

删除：鼠标移至节点深色区域后按下 D 键便能实现删除操作。

对于小箭头的三角形绘制，由于并没有自带的箭头绘制函数，通过了向量计算确定小三角形三个顶点位置后，也能通过绘制三角形的方式实时绘制出小箭头的效果。

图 12 和图 13 分别为单链表拖拽后整理顺序和连接过程和单链表的最终移动后（经过了编号重新输入）连接效果图：

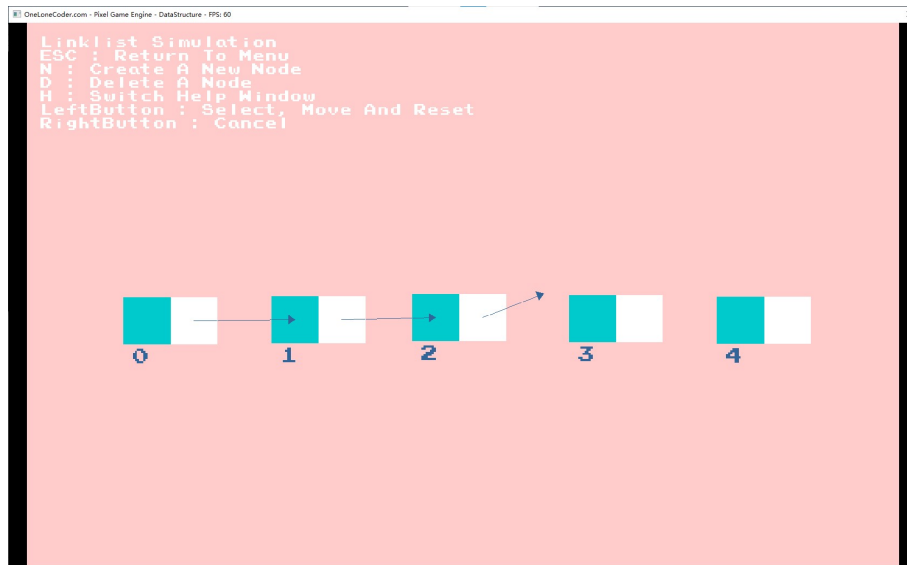


图 12 单链表拖拽后整理顺序和连接过程

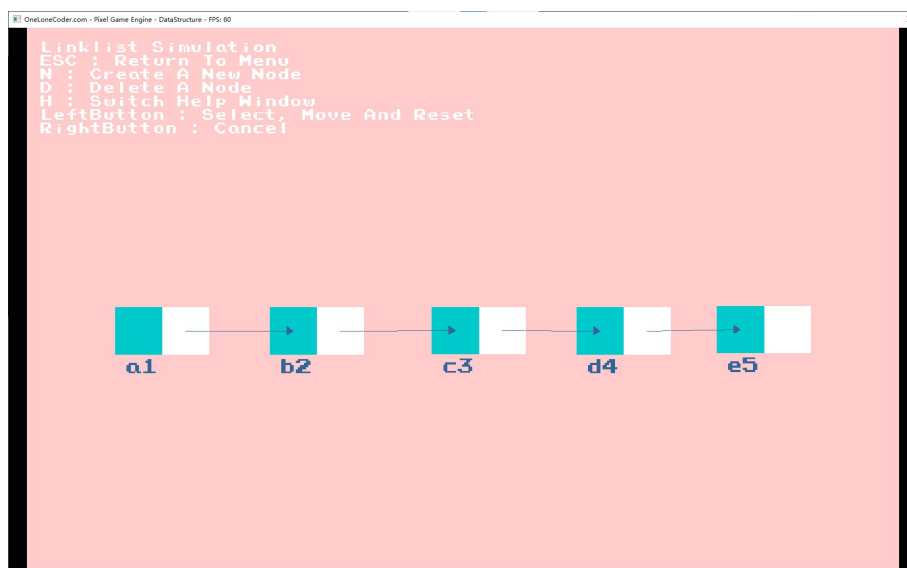


图 13 单链表效果图

类似地，编写了一个 Duallinklistnode 类来作为双向链表节点与引擎的借口，其功能与操作方法与单链表类似。

对于单链表节点和双向链表节点，为了避免直接删除节点后内存分配出现错误，我们采用了隐式删除的方式，即当该节点的字符串类型 data 变量的值为“gg”时，隐式删除该节点：在绘制和选中的时候设置为不能选中，便达到了删除节点的效果。

图 14 和图 15 分别为双向链表的效果图和按下 D 键后删除节点的效果演示：

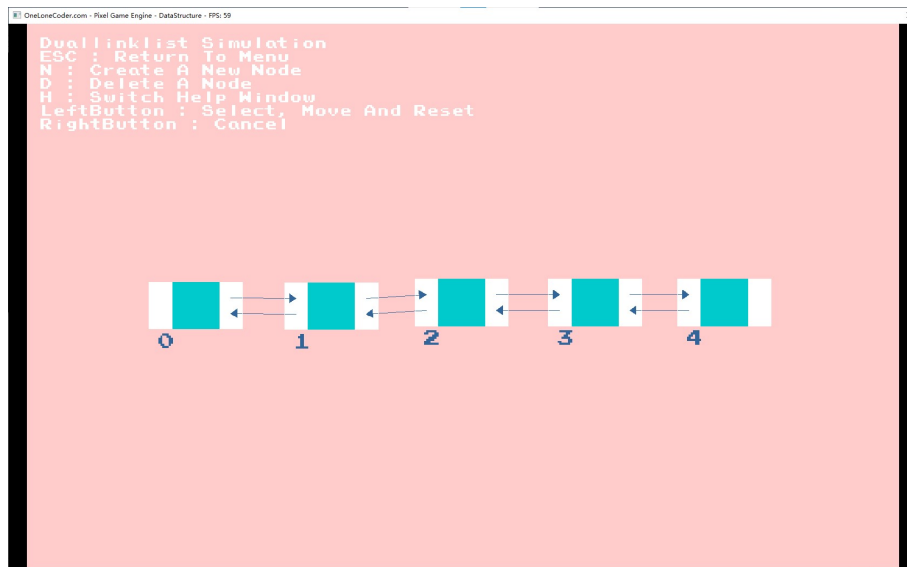


图 14 双向链表效果图

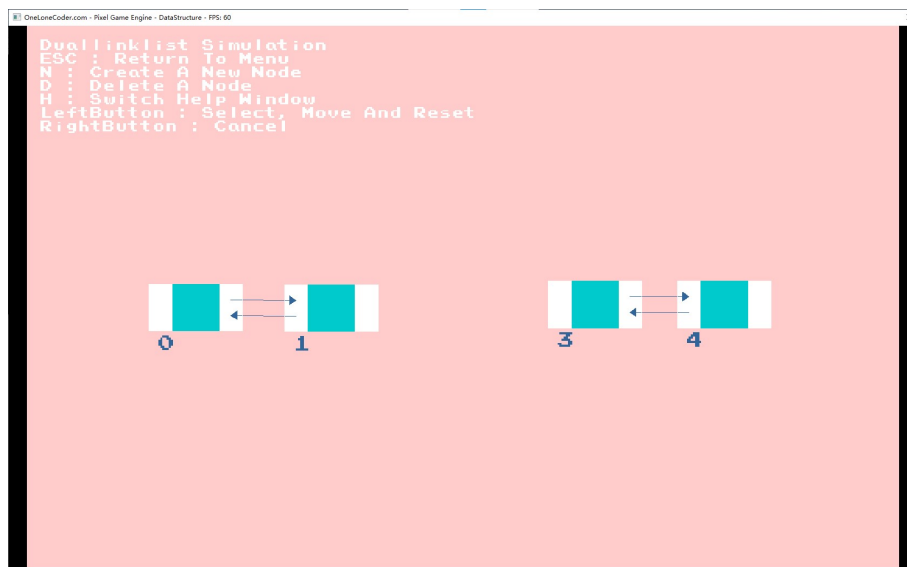


图 15 双向链表节点删除

类似地，也进行了 Graphnode 类的编写，其不同在于一个节点的出度可以是很大的，因此选择了用 stl 中的 vector 数组实时维护一个节点所有的出边，这样既能做到多出边的实现和绘制，也能做到单个边删除的效果。

图 16 即为图数据结构的效果图，可以看到其也能表示有向图的关系，若需表示无向图，只需控制程序不绘制小箭头即可。

图 17 即为图的单条边删除操作演示，一条条删除边的操作更能方便使用者进行图结构的修改和重新整理。

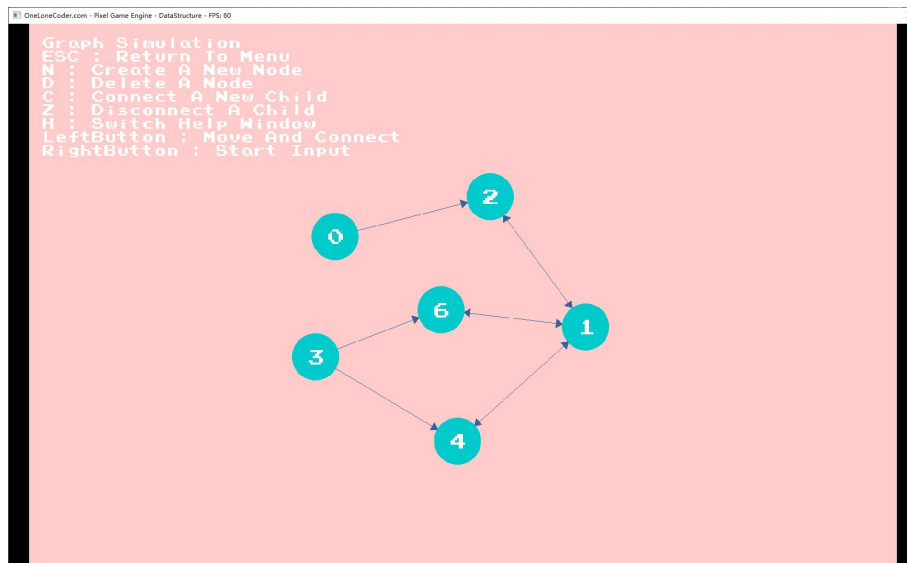


图 16 图数据结构效果图

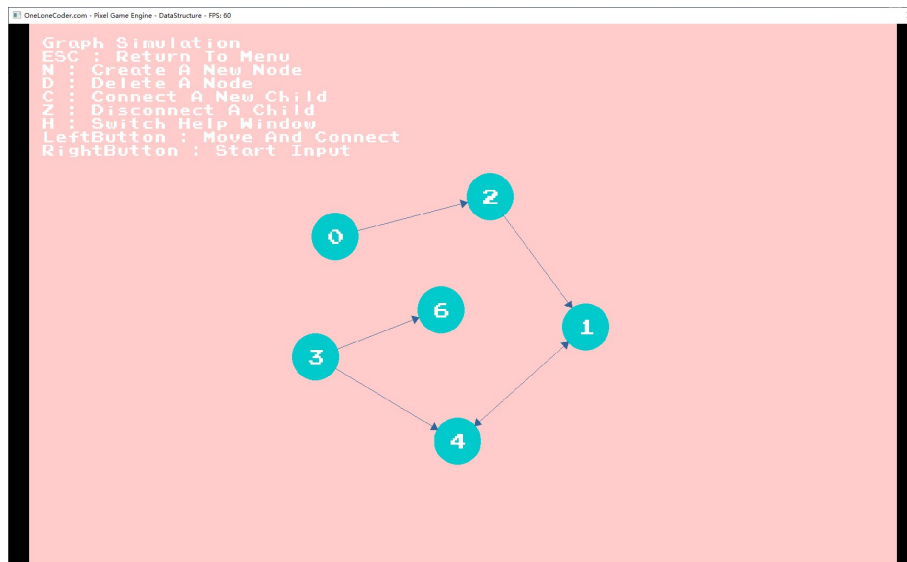


图 17 图的边删除操作演示

### 3.4 Mini-Racing 赛车模拟

首先，实现了赛车运动状态随时间改变的物理建模。当按下方向键时，小车会产生一个固定的加速度，又引入了一个类似摩擦力的阻力，使得小车受到的阻力大小和其速度成正比，因此，小车最终会趋于一个稳定大小的速度。这里运用到了引擎内置的 `vifd` 向量类型，方便进行向量运算。

在此，也利用接入了 `Camera2D` 库中的两种相机运动方式。按下 1 键对应 `Normal` 类型的相机移动方式；按下 2 键对应 `Lazy Move` 类型的相机移动方式。相机会自动跟随小车进行移动，达到在大地图上的一部分小车真实移动的效果。

同时如图 18 所示，实现了小车随其运动方向自由转动的效果。同时，编写了 Has\_hit 函数实现了对小车的碰撞的实时检测，并作出对应运动状态的改变。

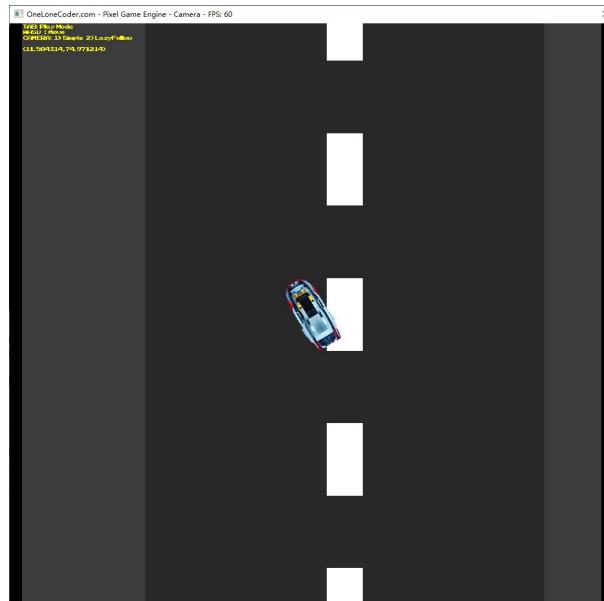


图 18 赛车自由方向运动演示

对于地图，我们使用一个一维的 vector 数组代替二维数组进行存储，利用 int 变量的不同值表示该位置的不同地形类型。对于视角的转换，当我们按下 tab 键时，transformView 函数便会发生作用，此时利用鼠标滚轮便可放大地图，看清地图的更多样貌。

图 19 便为可视化大地图的部分展示：

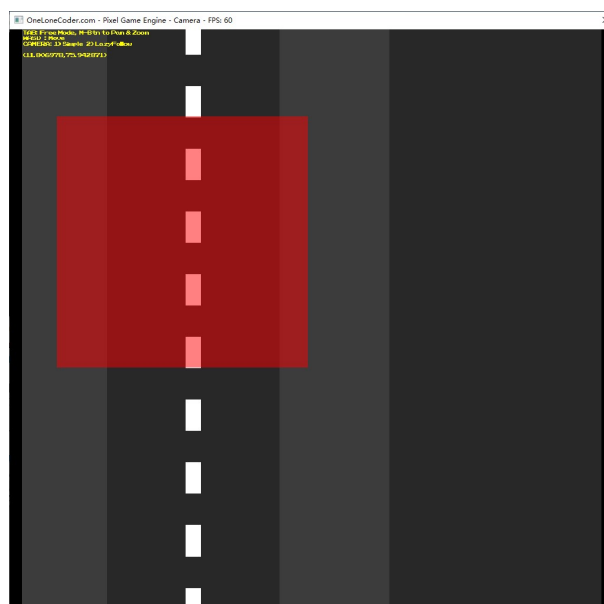


图 19 可视化大地图部分展示

## 4. 见习收获与体会

### 4.1 思考与收获

在这次暑期科研见习后，我学习了如何使用 `olcPixelGameEngine` 进行项目的开发，锻炼了我的能力，拓宽了我的视野，同时也让我知道了如何通过自己自主学习，研究并应用各式各样的开源库和框架，提升了我对代码编写和项目管理的一定经验。

同时我也体会到了在一个软件编写项目中，最关键的是搭建好软件整体的结构，并厘清程序内部的思路和关联。以本次见习任务出发，我也自行观看学习了网络上的 GAMES104 课程，对现代游戏引擎的搭建知识有了初步的了解和认识，结合本次见习实践，总体增进了我对于项目型软件的整体概念重要性的理解。

因此，本次项目让我受益匪浅。

最后，也感谢汪老师在本次暑期科研见习岗项目中做出的指导！

## 5. 附录

### 5.1 项目界面（部分）

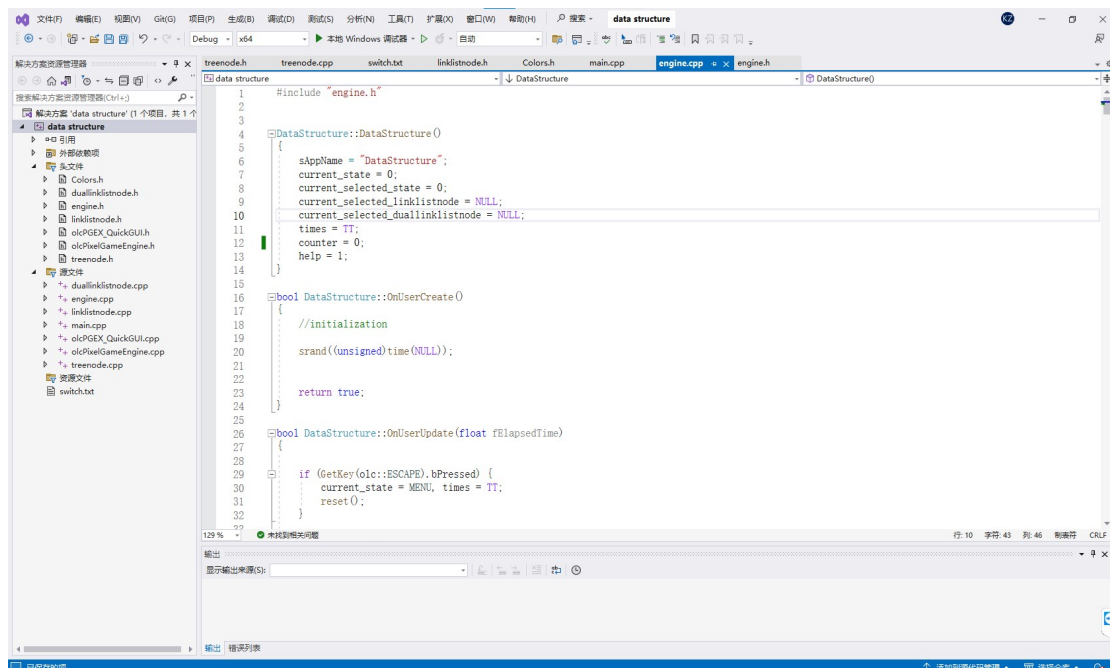


图20 项目界面（部分）

### 指导教师评价意见及成绩评定

赵楷越同学在本次暑期实习中态度认真，表现出了极高的学习热情和创造力，能够独立完成各项任务，并在几次讨论下不断改进和优化自己的作品，展示了 olc 简单图形引擎的完备功能和灵活性。作品也具有良好的交互性和美观的界面，体现了学生对用户体验的关注和理解。学生的作品也为其他实习生提供了借鉴和启发，促进了项目组内的交流和合作。

本人对实习生在本项目中所取得的成绩表示肯定，认为学生出色地完成了本次暑期实习项目。

见习成绩： ☐通过      ☐不通过

☐ 同意学生获得 1 学分

指导教师（签名）：

年    月    日

### 院（系）意见

负责人（签字）：

院（系）公章：

年    月    日