

项目报告

钱磊

2024 年 1 月 16 日

1. 引言

本项目为 SI100B 课程的最终项目，项目文件已上传到 https://github.com/QLL-ABAB/pj_game。我制作了一款 Roguelike 类型的游戏。玩家初始出生在森林，拥有 6 金币，3 攻击力和 10 血量。你可以通过打怪或者玩游戏获得金币来武装自己。游戏的最终目标是击败 boss 外星人。

2. 项目实施（基本得分点）

2.1 场景

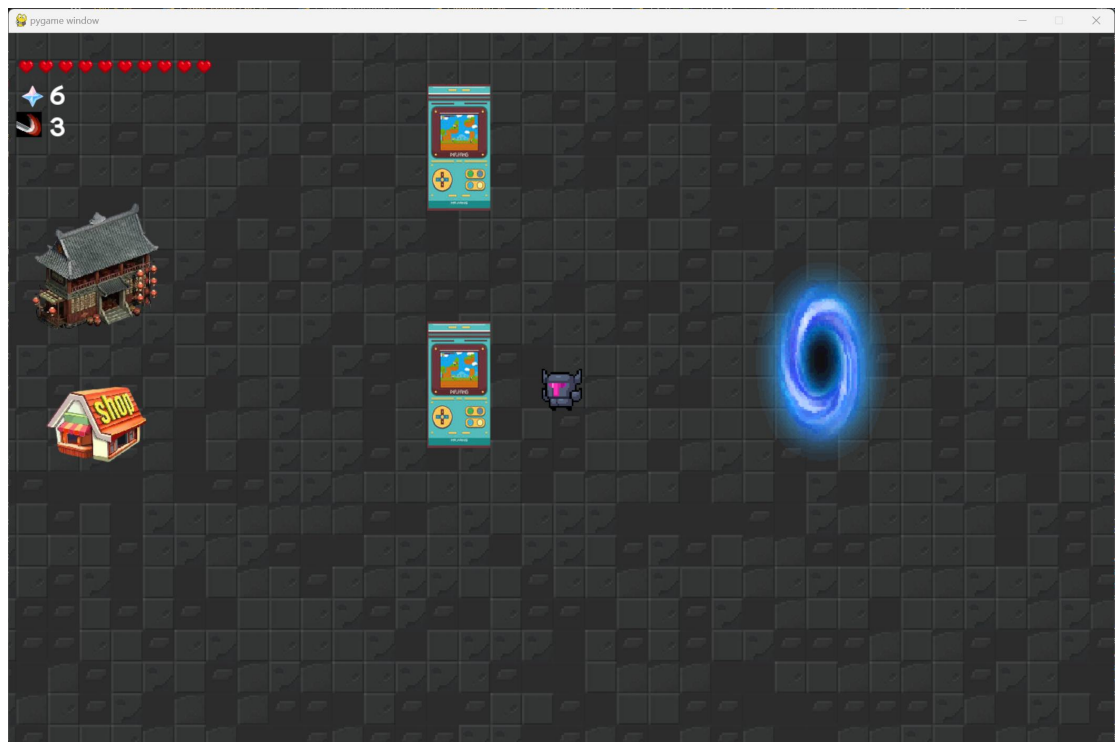
2.1.1 主要场景简介：

我们的游戏中一共有三个大场景：

- 1、城市：玩家可以在此处与 NPC 对话，在商店中用金钱进行交易，刷新森林地图或者游玩小游戏，见图 1；
- 2、森林：玩家可以在此处与怪物战斗获取金钱，或者进行存档，见图 2；
- 3、BOSS 房：玩家进入 BOSS 房后只能与 Boss 决一死战，见图 3。



（图 1）



(图 2)



(图 3)

2.1.2 满足镜头移动的大型场景：

我们游戏中的 BOSS 房，森林和城市都是大型场景，其中城市 and 森林有着可以跟随玩家移动而进行视角移动的摄像头。

镜头移动的原理：每当满足镜头移动的条件时，所有物体（含地砖、NPC、玩家）都会获得 一个偏移量，该偏移量等于在这一帧玩家试图移动的距离，在

渲染时所有物体的位置都会加上这个偏移量的相反数。因此，玩家移动了 $(dx, dy) + (-dx, -dy) = (0, 0)$ ，即在原地不动，其它物体全部移动偏移量的相反数，达到了镜头移动的效果。

2.1.3 互动物品

我们的游戏中有四种互动物品，分别为树、墙壁、火堆和存档祭坛，如图。



其中，树与墙壁均起到障碍物的作用，玩家无法跨越这些障碍物。而火堆无法阻挡玩家走动，但玩家在火堆上走动时会持续受到伤害。与祭坛交互可以实现存档功能。这些互动物品会在地图砖块后被渲染，形成它们坐落于地图砖块上的效果。

2.2 角色

2.2.1 主要角色

在我们的游戏中，玩家可以通过 wasd 四个按键操控主要角色四处移动，通过传送门进入其他场景，通过与其他角色碰撞来触发特殊事件。主要角色的移动通过检测按键按下与抬起这两种事件的方法来实现，当按键被按下，角色就会持续朝指定方向移动，按键抬起后，角色不再朝指定方向移动。按下按键等事件的检测全部在 `GameManager.py` 中的 `run_game` 函数中实现。

角色接收移动后会产生一个即将要移动到的位置，用即将要移动到的位置去检测障碍物，如果遇到障碍物则不再前进，更多的也会在检测碰撞的机制当中说明。

主要角色行走时会播放行走动画，当玩家按下相应按键后，程序会记录下行走朝向，并以此为依据更改主要角色的图片，使图片与行走方向相匹配，并循环播放一组图片以达到动画的效果。行走动画主要通过 `Player.py` 中的 `animation` 函数实现。

2.2.2 友好 NPC

NPC 主要为两个游戏机、商店以及酒店。通过与其碰撞并且按 e 可以进入不同功能的场景。

两个游戏机的主要内容分别在 `Game_scene.py` 和 `Game_scene2.py` 中，而商店则在 `Shop.py` 中定义，酒店较为简单，直接写在了 `Scenes.py` 中的 `Scene_city` 中。



2.2.3 简单敌人

野外场景存在着两种敌人：小型机器哥布林和大型机器人。

小型机器哥布林血量和攻击力都较低，掠夺金币和造成伤害的能力都较弱；而大型机器人则全面更高。森林当中，敌人的类都在 `Wild_scene_entitylike` 中定义。回合制对战场景的代码在 `Add_windows.py` 中。



2.2.4 特殊敌人

Boss 房中存在着 Boss 外星人，击败他就能获得游戏胜利。

Boss 可以发射子弹、召唤随从。Boss 和随从的定义都在 `Boss_and_fellows.py` 中完成。更多的内容也会在创意模块当中讲解



2.3 游戏机制

2.3.1 核心机制

2.3.1.1 代码：

代码的核心为 `listen-post` 体系

2.3.1.2 游戏：

玩家通过操作主要角色移动来与要互动的角色碰撞，碰撞发生后加上按 E 等指令即会触发相应事件。小怪战斗采用回合制，玩家与怪物交替发起攻击。而 Boss 则采用偏向飞机大战的玩法。玩家需要击败怪物或者打小游戏以获取金币，玩家的最终目标是变得足够强大去挑战 Boss 外星人。

2.3.2 碰撞系统

程序通过 `pygame` 内置的碰撞检测方法来检测玩家与实体间的碰撞。

当玩家与障碍物发生碰撞时，玩家则无法发出 `CAN_MOVE` 的指令，因此无法移动。当玩家与火堆、NPC、敌人、传送门等实体碰撞时，如果是复杂的事件，碰撞检测方法会向事件队列发送相对应的事件，在事件队列中被相继处理；如果碰撞事件相对简单，则会通过改变内部变量发生作用。

事件队列的处理在 `Game_manager.py` 中实现

2.3.3 资源系统

玩家击败怪物或者玩小游戏可获得金币，打败大怪物（机器人）玩家获得的金币更多。金币可以用来在商店中购买血量、攻击力和装备等，提高玩家的战斗力。玩家的血量也是一种资源，玩家需要时刻分配好自己的血量，确保自己不被怪物击杀的同时赚取足够多的金钱。

玩家的资源是玩家类的实例变量。

2.4 AI 系统

2.4.1 对话系统：

你可以在商店中与爱尔奎特进行对话，具体对话的操作详见 README 当中的指导。AI 的引入操作在 `Openai.py` 文件当中完成，具体的对话代码写在了 `Shop.py` 中

2.4.2 对战系统：

AI 决策系统加在了与小怪的回合制游戏中，点击 RTPS 则可进入 AI 代理。AI 会选择进攻或者防御，与敌人战斗。

2.5 游戏性

2.5.1 菜单

我们的游戏有三种场景菜单，主菜单、失败界面与胜利界面。

进入游戏与游戏重新开始时会显示主菜单，此时玩家需要选择存档，然后按下 Q 键以进入游戏。

当玩家死亡时，会出现失败界面，显示游玩时间，按下 R 键可重新开始游戏。

当玩家击败外星人时，会出现胜利界面，按下 R 键可重新开始游戏。

同时各个场景之内也有各种选择菜单。

2.5.2 BGM

我们的游戏在主菜单、失败界面、胜利界面、城市场景、野外场景与 Boss 房都会播放不同的 BGM。

BGM 的切换通过 `Bgm_player.py` 中的类实现。

2.6 代码

用编辑器打开文件夹后运行 `Main.py`，或直接通过在文件夹目录打开终端并输入 `python Main.py` 运行，即可进入游戏。

程序采用面向对象编程的编程方式，代码中难以理解的地方全都添加了注释。下面罗列程序中的各个文件的相应作用：

1. Settings.py:
游戏设置文件，包括游戏窗口大小、背景音乐、字体等。也包括了存档系统。
2. BgmPlayer.py
该文件负责处理背景音乐的播放。
3. Gamemanager.py
切换场景与接受处理键盘的输入。
4. Main.py
该文件是游戏主程序，包含了游戏主循环，负责运行游戏。
5. Player.py
该文件定义了 Player 类，该类负责处理玩家移动、更新玩家参数、播放人物动画等。
6. Scenes.py
该文件定义了 forest 和 city 这两个包含镜头移动的场景类。
7. Shop.py
该文件定义了商店类，负责处理商店的显示、购买道具、结算价格以及 AI 聊天的窗口等。
8. Add_Windows.py
该文件定义了游戏里的文字和文字框，同时也负责定义应用文字和浮窗较多的回合制战斗景。
9. Collections.py
该文件定义了基本的父类以供继承，也定义了可以使用的代码工具。
10. Portals.py
该文件定义了传送门类。
11. Boss_and_fellows.py
该文件定义了 Boss 类和小兵，并包含其需要的子弹类等。
12. Boss_Scene.py
该文件定义了 Boss 战斗场景，包含里面实体的各种互动。
13. Boss_Scene_EntityLike.py
该文件定义了 Boss 战斗场景里的场景实体类。
14. Wild_Scene_EntityLike.py
该文件定义了 Forest 场景里的场景实体类。
15. City_Scene_EntityLike.py
该文件定义了 City 场景里的场景实体类。
16. Openai.py
该文件定义了 OpenAI 聊天机器人，负责处理 AI 的聊天。
17. Game_scene.py :
贪吃蛇
18. Game_scene2.py :
扫雷
19. Restart.py :
游戏开始和结束界面
20. save1.json / save2.json / save3.json :
存档文件
21. assets :

游戏素材文件夹。

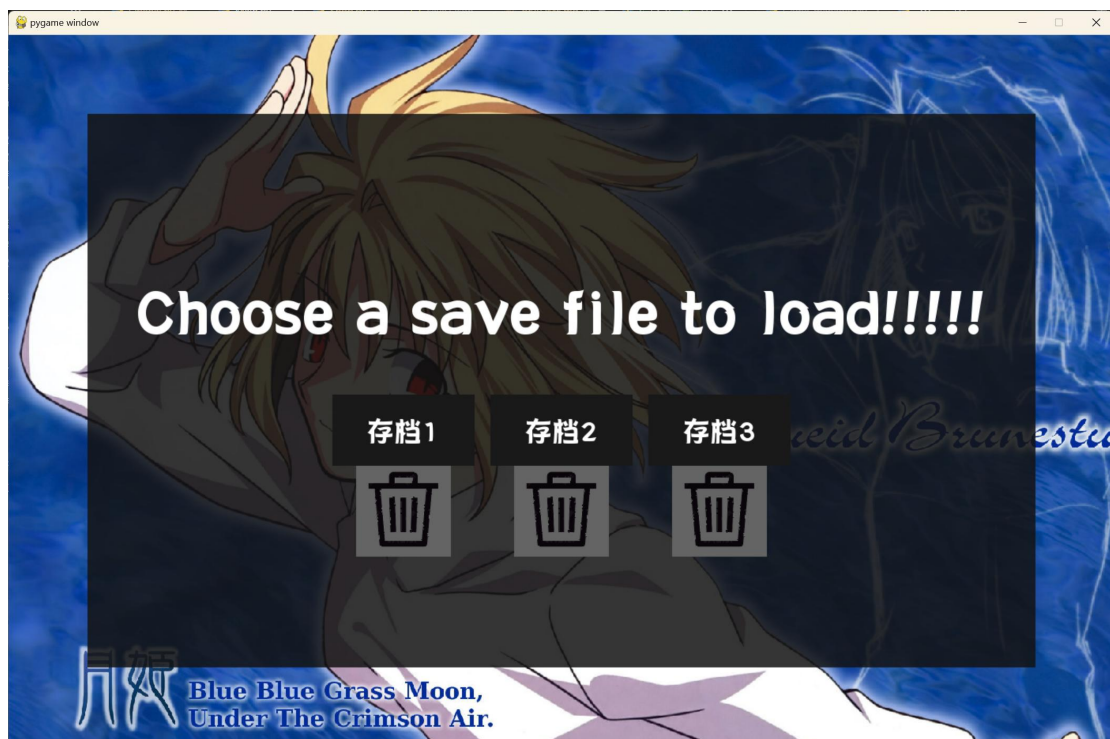
3. 创意部分

3.1 两种小游戏

在城市当中与游戏机碰撞并按 e 可以进入小游戏，丰富了游戏内容。

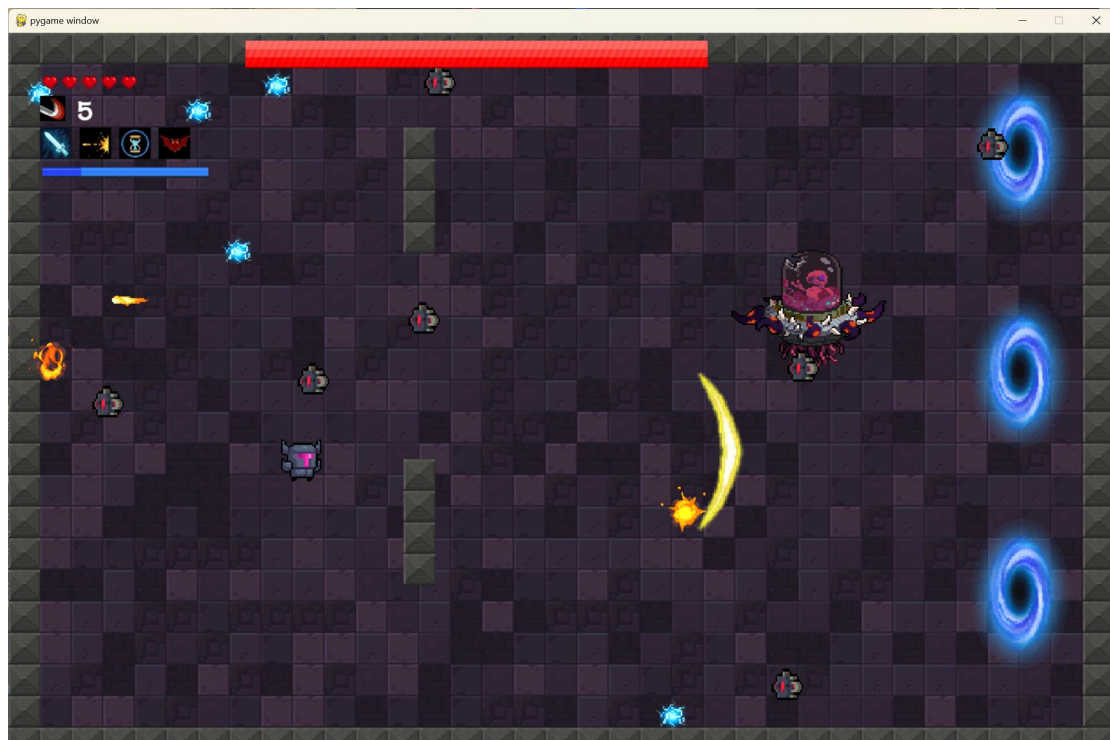
3.2 存档系统

游戏设置的存档系统，使玩家能够退出并且保留进度。（在游玩时也可以通过直接改变存档文件中的数值来实现作弊，减少游戏体验的难度）



3.3 妙趣横生的 Boss 战

游戏的 boss 战包括了大量的内容



3.3.1 光剑系统

在商店中购买咖喱棒可以解锁光剑技能。蓄能完成后左键可以蓄力增大光剑的宽度，右键发射威力巨大的剑气。剑气可以扫除绝大部分子弹和敌人，是 boss 战获胜的巨大助力

3.3.2 场中掩体

场中有两处掩体，可以供玩家躲避部分子弹。掩体也改变了小兵的移动路径，为玩家提供了多种选择。

3.3.3 穿透、冷却等被动技能

可以在商店处购买穿透、冷却和吸血的被动技能。这些技能可以使子弹穿墙，加快攻击频率和增加续航。多重技能的存在丰富了 boss 战的可能性。

3.3.4 灵活的小怪

Boss 战中的小怪极其灵活，他们会从三个传送门中随机出现，并且在经过掩体之前做无规则运动。在到达掩体之前，他们会调整方向来通过掩体。通过掩体之后，就会对角色进行自爆式的袭击，大大提高了游戏的紧张程度。

3.3.5 黑洞的吸入

角色在靠近出怪的黑洞时，会被牢牢吸住，直至死亡

3.4 动画与特效

游戏当中有许多动画和特效。角色受击时会变红，移动有移动动画，死亡有倒地动画。Boss 和小怪死后都会有爆炸动画。传送门也有对应的动画。

3.5 属性显示

角色的各个属性都显示在左上角。包括角色的血量，攻击力，技能和装备，使角色的数据变得直观。

3.6 商品繁多的商店



商店中可以提升两项基本属性、购买四种技能和一个道具。多样的商品为游戏提供了更多选择。

3.7 随机生成的湖泊

场中有通过代码随机生成的湖泊块，丰富了地图的组成成分，增加了随机地图的复杂性。

3.8 火堆爆条与熄灭机制



场中的火堆可以进行交互，通过积攒热量条的方式进行扣血，热量会随着远离火堆而下降。购买桶之后，你也可以通过从湖中取水来浇灭火堆，使游戏更贴近现实。