SI100B Project Final Report

潘佑邦.宋梓冬.吴俊阳

1 引言

本项目为 SI100B 课程的最终项目，项目文件已上传至 https://github.com/TossACoinTAC/Team-SAVE-MY-LINEAR-ALGEBRA，其中 **dev** 分支中记录了项目的全部开发历史，**main** 分支则保存了项目的最终版本。

本项目是一款 Rougelike 类型游戏，主要玩法是玩家操纵 Isaac 发射子弹、放置炸弹击杀怪物，最终击败 boss，在游戏过程中，玩家也可以通过抽奖、与 NPC 对话等方式获取隐藏奖励。

当然，Isaac 的血量是有限的，当血量清零时，Isaac 将会死亡，游戏也随之结束。

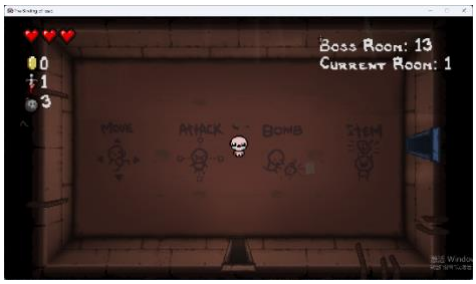
游戏的原型和素材来自于《以撒的结合》（The Binding of Isaac）。

2 项目实施

2.1 Scenes

2.1.1 主场景

StartRoom 为玩家的出生房，每轮游戏玩家都会在此出生（或复活），在这里玩家可以了解可以进行的操作（移动、攻击、放置炸弹）。



游戏的奖励房间有 **Lucky Room** 和 **NPC Room** 两种，奖励房间不设有墙体，且门不会关闭。玩家在此可以通过与 NPC 对话获得相应的奖励。

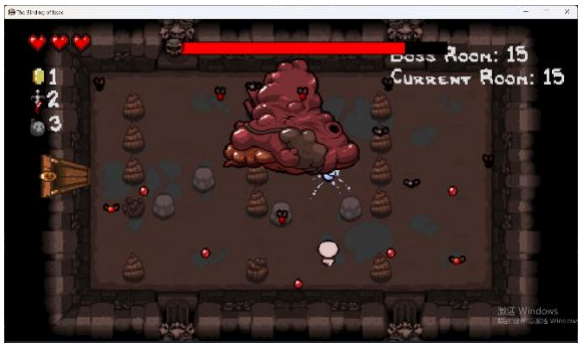
在 **NPC Room** 中，玩家可以靠近 NPC 按 **q** 键与 NPC 对话，NPC 是无敌的，不会受到玩家攻击。

在 **Lucky Room 中**，玩家可以通过抽奖机以金币*5 为代价进行抽奖（靠近抽奖机按 q 键抽奖），抽奖可以获得血量、攻击力、金币增益，抽奖机同样无法被破坏。

游戏具有多种战斗房。在战斗房中，玩家可以与怪物交战，击杀怪物可以获得金币奖励。

战斗房中会随机生成 Shit 和 Rock，对玩家和普通怪物进行阻挡，墙体的具体生成逻辑见 2.1.3。

每轮游戏有且仅有一个 BossRoom，玩家可以根据右上角指引寻找 BOSS 房间位置。BOSS 在该房间内生成。在这里玩家可以挑战 BOSS，击杀 BOSS 可以直接获得游戏胜利，进入胜利结算画面。



2.1.2 Doors

Door 可能在房间的上、下、左、右方位出现，进入右侧和下侧的门，你将远离 StartRoom。不同类型的 Door 代表了将要进入的不同房间的种类。

在我们的代码中，判定将要进入房间的种类是通过判定门的种类实现的。

在战斗房间中，战斗进行时，门会关闭，当房间内的所有敌人被击杀时门才会打开。

2.1.3 障碍物

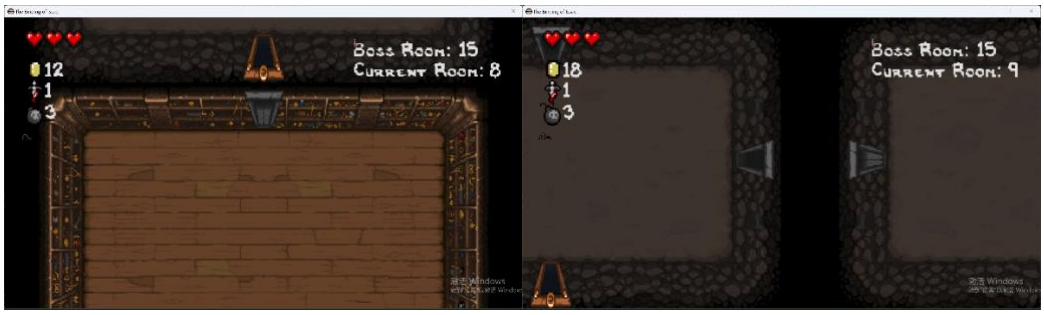
房间中的障碍物有两种：Shit 和 Rock。其中 Rock 无法被子弹破坏，但可以被炸弹破坏，而 Shit 既可以被子弹破坏，也可以被炸弹破坏，它的血量为 5。

墙体在战斗房间中具有多种不同的生成逻辑，其中 Rock 在房间中随机位置、随机数量生成；Shit 的生成在三种模式中随机选择。模式 1：间隔生成随机列数的墙体 模式 2：在房间的四角生成单独的墙体 模式 3：在房间的中心生成 4x4 的墙体。

玩家与敌人实体都会被障碍物阻挡（Fly 除外）。

2.1.4 场景切换（镜头跟随）

游戏主要由 7 种不同的房间构成，虽然在单一房间内镜头是固定的，但是在进行房间切换时，我们设置了过渡动画，镜头会做跟随运动，实际上实现了不同场景之间的转场效果。



2.2 Characters

2.2.1 Main Character

玩家需要扮演 Isaac，一个哭泣的孩子。w,a,s,d 键可以控制 Isaac 向前，左，后，右移动，Isaac 向不同方向移动时都有不同的动画，由于我们的代码中移动操作被保存在一个向量中，因此实际上玩家可以向八个方向移动，即同时实现左右移动和前后移动。按住 LeftShift 键则可以加速。

↑，←，↓，→键可以控制 Isaac 向不同方向发射子弹（tear），子弹，子弹的攻击力可以增加，也有多种不同形态，我们将在“创意”部分中对其进行更详细的介绍。

你可以使用 e 键放置炸弹，炸弹会造成范围伤害，在爆炸时同样也会播放动画。但是请注意，炸弹同样会伤害到你自己（可能会触发隐藏机制），所以请务必谨慎使用。

2.2.2 Normal Enemies

Normal Enemies 不会主动攻击玩家，只会到处移动，如果玩家与怪物发生碰撞，会被扣血。

其中 Bug 具有普通形态和冲刺形态两种，它的初始血量为 3.

Fly 会做无规则的飞行运动，它的初始血量为 1.

初始血量？因为我们还做了多周目系统，具体请见 2.5.2

2.2.3 Gurdy

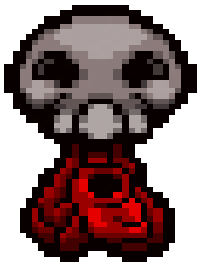
Gurdy 是游戏的最终 Boss，他具有很大的体型，初始有着 100 的血量（显示在游戏界面的正上方）。虽然不会移动，但 Gurdy 会不停地释放 Fly 干扰玩家，同时它自己也会向玩家所在位置散射子弹（琪露诺？什么时候？）。战胜它，你就能获得游戏的最终胜利。

在代码实现中，我们将 BOSS 的身体部分和其攻击机制分装在两个类中进行开发，这使得 boss 机制的实现更为灵活高效，同时也具有一定的整体性。

2.2.4 NPC

在 NPC Room 中，玩家可以靠近 NPC,按 q 键进入对话框与 NPC 交谈。

NPC 有 Trainer（左）和 Merchant（右）两种。




Trainer 的目标是训练玩家的数学水平，他会向玩家循序渐进地提问数学问题并帮助玩家解答，如果玩家回答正确，Trainer 会帮助玩家回复血量或者增强玩家的攻击方式（豌豆射手-->三线射手），而如果玩家不能正确回答简单的数学题，则会面临 Trainer 的惩罚（HP--）。

玩家可以支付金币或血量与 Merchant 进行交易，购买治疗、炸弹、攻击力和攻击速度加成，如果玩家没有足够的货币，NPC 会拒绝与玩家交易。

玩家可以在对话框内通过打字输入对话内容，按 Enter 键发送，玩家的输入信息和 NPC 的回复会显示在上方。输入 quit or exit 退出对话框。

同时，两位 NPC 均接入了 LLM Agent，会根据玩家当前状态（HP、ATK 等）做出合适决策（美愿时代的留恋！（bushi））具体见 2.4

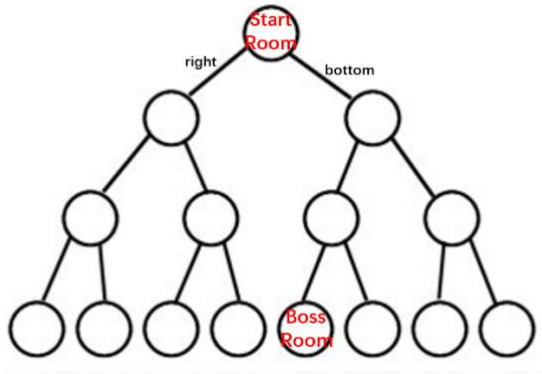
2.3 GameMachine

2.3.1 Core Game System

玩家操纵 Isaac 发射子弹、放置炸弹击杀怪物，最终击败 boss，在游戏过程中，玩家也可以通过抽奖、与 NPC 对话等方式获取隐藏奖励。

对于有怪物的房间，玩家只有在击杀所有怪物后才能进入下一个房间。

我们通过一个二叉树存储了地图的结构，StartRoom 为根节点，之后地图向右方和下方延申四层，因此一共有 15 个房间，房间随机刷新，我们保证 Boss Room 一定在第四层生成。其中的每个节点都保存其父节点和左右子节点的信息，保证了快速的迭代和回溯。



2.3.2 Collision System

我们的游戏配备了复杂的碰撞系统，包括玩家与场景、道具与场景、道具与敌人、玩家与敌人、玩家与 NPC、玩家与道具之间的碰撞。一部分碰撞通过 pygame 内置的方法实现，另一部分则通过自定义的 mask_groupcollide 方法实现像素级碰撞。

一些碰撞发生时，会产生动画和音效，例如子弹在击中墙体和敌人时，都会有动画和音效。

一些主动的碰撞依赖玩家的操作。例如玩家靠近 NPC 并按 q 键，可以进入 LLM-Chatbox 界面与 NPC 对话，玩家靠近抽奖机并按 q 键，可以进行抽奖。

2.3.3 资源系统

游戏中的资源主要有血量、金币、攻击力和炸弹等。资源可以获取的消耗，奖励房间中资源的获取和消耗方式已经在上文提到。另外，在战斗房间中，玩家可以通过击杀敌人获取金币，击杀每个敌人可获取的上限为该敌人的初始血量。

玩家的资源在游戏界面的左上角实时更新。

2.4 LLM Agent System

2.4.1 Dialogue And Decision System

LLM Agent System 通过调用预制的 API 接口实现。玩家进入对话框后可以打字与 NPC 进行交流，对话框会显示当前对话回合的文字，而所有轮次的对话信息都存储在聊天日志中，且玩家当前的血量和资源数据会实时传给 LLM，LLM 即可根据历史信息与玩家状态做出合适的回答和决策。

Trainer 会检查玩家的血量，如果血量不满，则将奖励设计为给玩家增加血量，否则强化玩家的攻击方式，如将子弹强化为三倍体模式，以 buff 的形式发送回 GameManager。

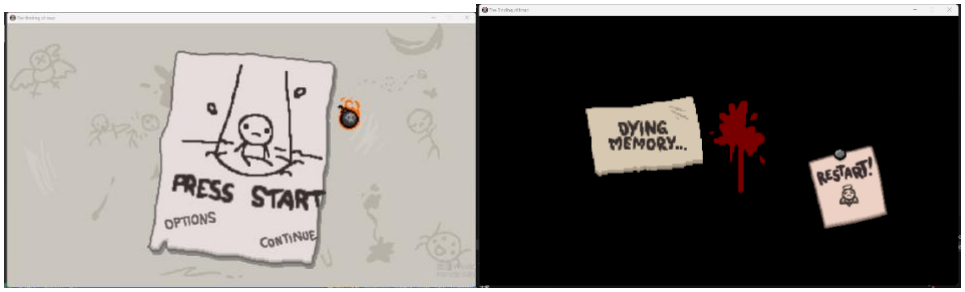
Merchant 则会综合分析玩家的血量、金币、攻击力、攻速等数据，决定与玩家的交易内容。

两位 NPC 拥有鲜明的人设，并会根据人设给出非常有趣的发言。

2.5 GamePlay

2.5.1 Menu

游戏设置有开始和结束菜单，开始菜单是游戏的第一个场景。这个场景中的各个实体都拥有自己的动画。（如中心的主题画和右上角的炸弹），当鼠标进入文字区域时，文字会被放大，单击 PRESS START 可进入游戏。



游戏胜利或失败会进入两种结束菜单，在结束菜单中，你也可以选择重新开始游戏。

2.5.2 多周目系统

Victory Lap、N18、五细胞...Rougelike 怎能没有多周目？Statics.py 中存储了当前

的 Hardship_Coefficient，敌人的血量、数量、攻击力都会据难度系数上升，每次击败 Boss 通关时会提升难度系数，且仅在玩家死亡（游戏失败）时重置。

2.5.3 BGM

游戏的不同事件会触发不同音效，音效由 pygame.mixer 实现。在开始菜单，游戏会播放主题旋律，开始游戏会切歌，Isaac 的移动、射击，门的开启等事件都会触发不同的音效。

值得一提的是，我们原本准备了 docker image 以省去环境配置，但由于 mixer 依赖宿主机的实体扬声器，导致配置 image 反而比直接 install libs 慢且复杂，故而舍弃了 docker。

2.6 Code

仓库 main 分支中存储了游戏文件。

Src 和 data 目录中为游戏的资源文件，原本 Scripts 目录中为.py 文件，我们将不同功能的代码分别存放在 GameManagers,Characters,Scenes,TmpTools,UI 子文件夹中，功能与文件名相对应，（可惜由于要求直接输入 main.py 运行，不得不删除了 Scripts 目录）

其中，GameManager 为游戏的主要总控部分，运行程序 main.py 和静态数据文件 Statics.py 直接位于根目录下。

程序采用面向对象编程的编程方式，各功能高度 self-contained，文件 hierarchy 结构清晰，各类间依赖关系明确，GameManager 拥有所有 GameObject 的实例，其余 GameObjects 按所属逻辑关系拥有其他类的实例（如 Player 拥有 Tear 的实例）。程序中也多处使用 static variables、static methods 和 properties 等进行逻辑清晰的交互。

3 创意

3.1 Animations

我们在不同场景都实现了生动的动画效果和音效，包括开始界面，子弹击中敌人或障碍物、怪物死亡、场景切换等等，希望这些效果能够提升玩家的游玩体验。

我们认为 “打磨的像一个真正的游戏”的关键环节即是游戏动画。

我们的动画制作尽可能的保证精细/还原，且经过多次的更新迭代，包括但不限于：

1. 人物头和身体的动画分别制作，四个方向动画均不同
2. 玩家和 boss 的子弹都会有十几帧的碰撞动画
3. 小怪也实现了四向移动+冲刺动画，boss 攻击的动画经过多次调试，
4. 死亡随机留下 6 种血迹
5. 为了避免图片多次缩放导致的像素损坏，我们采取了独特的处理办法
6. 文本（fonts）采用游戏原生的字体

3.2 UI 界面

游戏会实时更新玩家的血量、金币、攻击力等属性，并呈现在左上角。另外，为防止玩家找不到 boss,游戏右上角对于 boss room 的位置进行了 hint。

最有趣的地方在于：由于玩家不清楚房间的编号逻辑，需要玩家运用自己的数学逻辑能力推断出最快前往 Boss Room 的路径，从而以更快的速度通关游戏。

3.3 隐藏效果

当 Isaac 用炸弹炸伤自己时，会触发分解（飞头（bushi））的隐藏效果,Isaac 的头颅会进行无规则飘动。