

SI 100B DIE 项目报告

◆ 综述

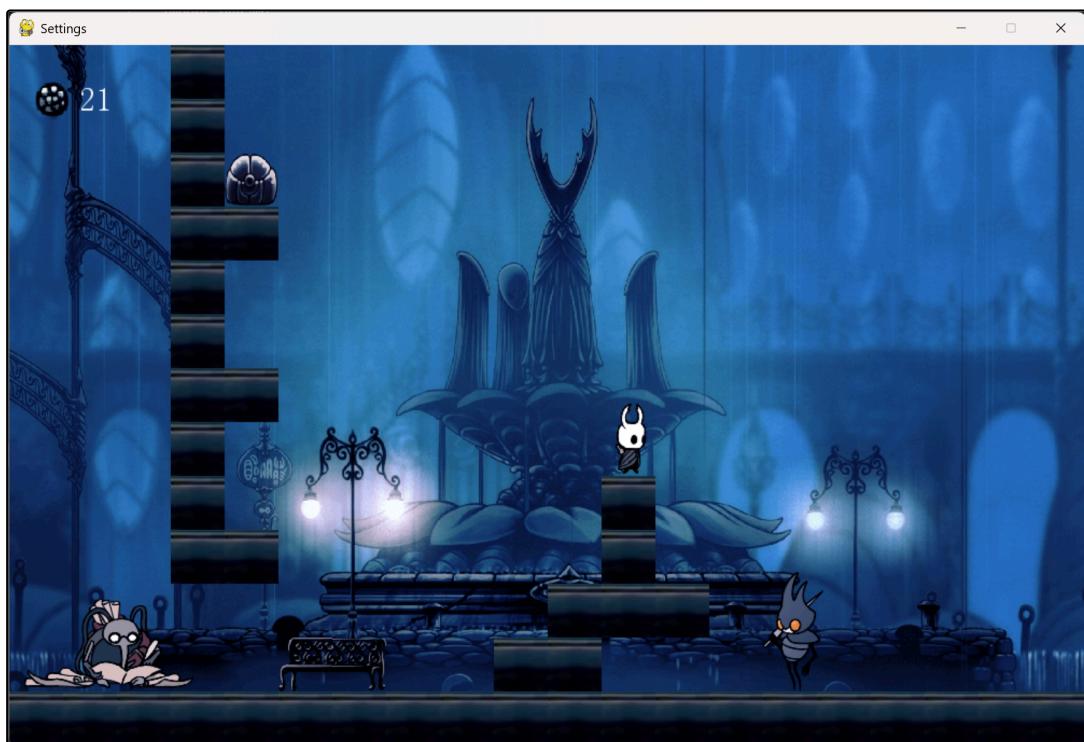
本项目为 SI100B 课程期末项目，项目文件已上传至 <https://github.com/Arahc/Si100-project-game-3Ts>。

1. 本游戏为对游戏《空洞骑士》的致敬（？）。玩家将扮演小骑士，在探索与战斗中成长，并最终击败 boss 辐光。基本的玩法为横板跳跃跑酷 + 卡牌战斗，带有少量随机要素。
2. 地图总共有4章，标号为 `map0` 到 `map3`，玩家出生在 `map1`，可以通过长椅在地图间进行切换。`map0` 为肉鸽地图，每次随机重新生成，用于刷钱刷怪。
3. 最终 boss 为辐光，位于 `map3`，击败其则游戏结束。
4. 游戏中一共有 4 种不同的怪物：泪城守卫、持盾守卫、黄蜂女和辐光。另外，由于我们的战斗系统**支持多波次战斗**，所以还有泪城守卫+持盾守卫的组合。其中持盾守卫和辐光具有特殊机制。每种怪物的特色都不同，难度也不同。

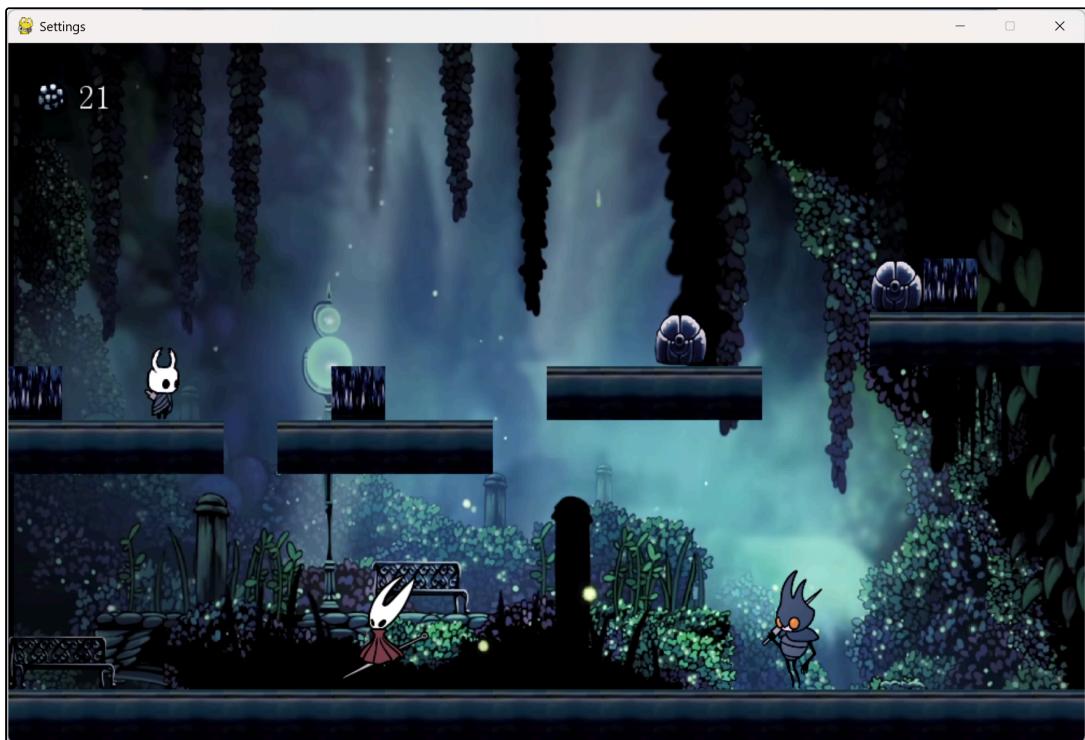
◆ Baseline 的实现

场景

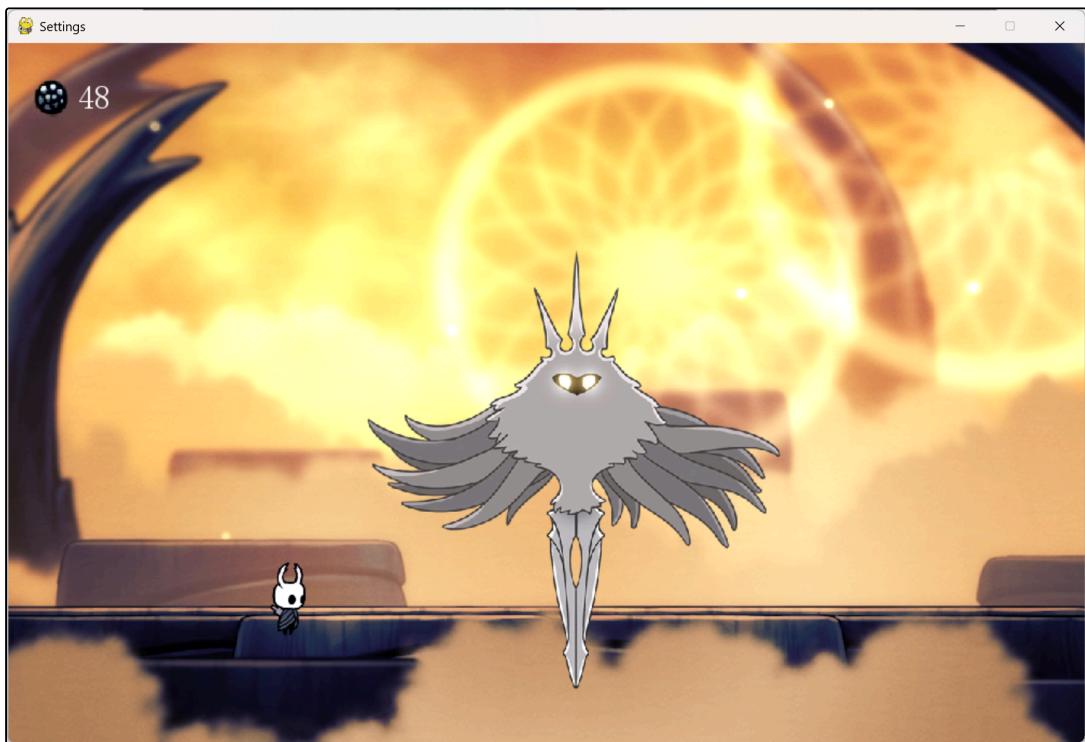
1. **多种不同的场景：**地图中共有三种背景不同的场景（`map1` 与 `map2` 沿用统一背景，但是内容不同）。在地图中按 `Esc` 可以回到菜单。地图展示如下：



map 1



map 0



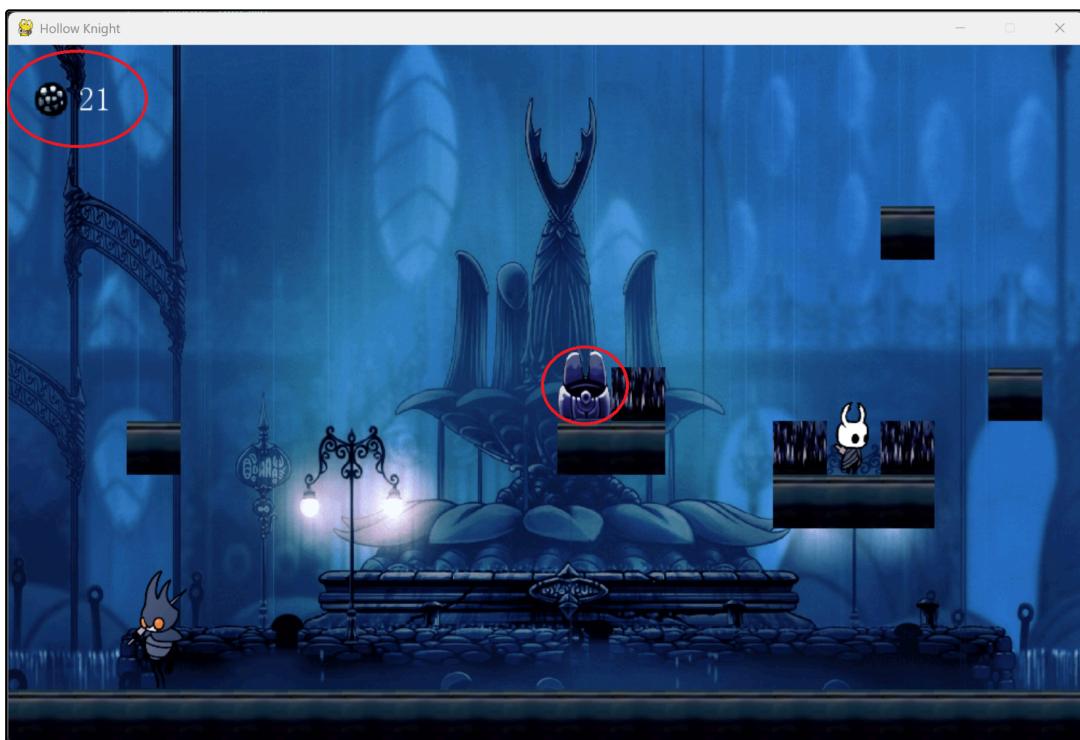
map 3

2.有镜头跟随的大场景：我们游戏中的每个场景都为**卷轴式长地图**，并且有着跟随视角，镜头跟随在 `MapPage.py` 中实现。当判断到玩家距离地图边界较近时，程序会给所有的物体（包括方块、NPC等）施加一个偏移量，来达到人物不动，物体移动的效果。

3.可交互物品：我们的地图中有障碍物 箱子 地刺三种要素。

- 障碍物：阻碍角色行动
- 箱子：按 `E` 打开后可以获得吉欧（游戏内货币），用于在商人柯尼法处购买物品。同时，打开前与打开后的箱子具有不同的外观，且在主线地图中箱子只能被开启一次（之后会一直维持被打开的外观）
- 地刺：小骑士接触到地刺则会受伤，会被传送到出生点重新开始挑战。

下图展示了这三类障碍物的作用，注意到该图中箱子被打开，左上角的货币数量随之增加（初始数量为 0）：



角色

1. 主要角色：小骑士

- 通过 `A` 和 `D` 控制左右移动，`Space` 跳跃，`L` 冲刺，按 `E` 以与 **所有可交互物品或人物** 进行交互。左右方向的移动通过检测按键按下与抬起这两种 event 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。而跳跃与冲刺则是检测按键的触发。

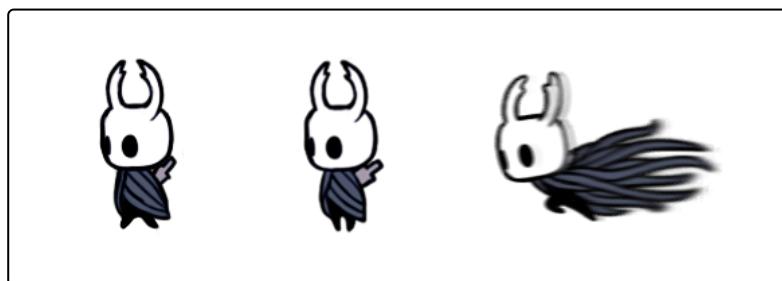
在这里有必要介绍我们的实现机制：我们实现了 `touchDown()` 函数来判断小骑士是否“脚踏实地”，如果是，那么当检测到玩家按下空格时，小骑士开始跳跃。在跳跃过程中我们采用了**真实的重力加速度系统**，也就是：小骑士的起跳具有初速度，且会受重力加速度的影响，当纵向速度为 0 时就进入下坠状态，直到再次“脚踏实地”为止。

此外，我们还实现了冲刺功能：冲刺功能在“脚踏实地”的情况下可以一直使用。但如果冲出平台，或者在空中，则只有一次使用机会，需要再次落地才能补充。为了尊重原游戏，并且增强平台跳跃的游戏性，**冲刺途中不能切换方向、不能中途停止，且不受重力影响**。

- 精致的移动动画：在 `A` 和 `D` 被按下时，小骑士会相应改变行走方向，渲染的图片也因此切换。

同时，小骑士的走路有两种姿态，在持续按住 `A` 或 `D` 时，小骑士的**走路姿态会来回切换**（注意腿部区别）。

此外，由于我们的游戏含有跑酷（平台跳跃）元素，我们给小骑士**添加了冲刺动画**，这些姿态见下。上述所有小骑士的移动和动作都在 `MapPage.py`（地图界面）里实现



2. 友善 NPC：

- 游戏中的友善 NPC 共有两位，分别是蛾族先知 (Seer) 和商人柯尼法(Cornifer)。在与他们有模型接触的情况下按 `E` 可以进行交流（在后面会深入讲述）。接触柯尼法并按 `B` 可以进入商店界面（图片价格仅供参考），进行购买。这些 npc 在 `Npc.py` 中被初始化，并在 `MapPage.py` 中被部署到地图中。

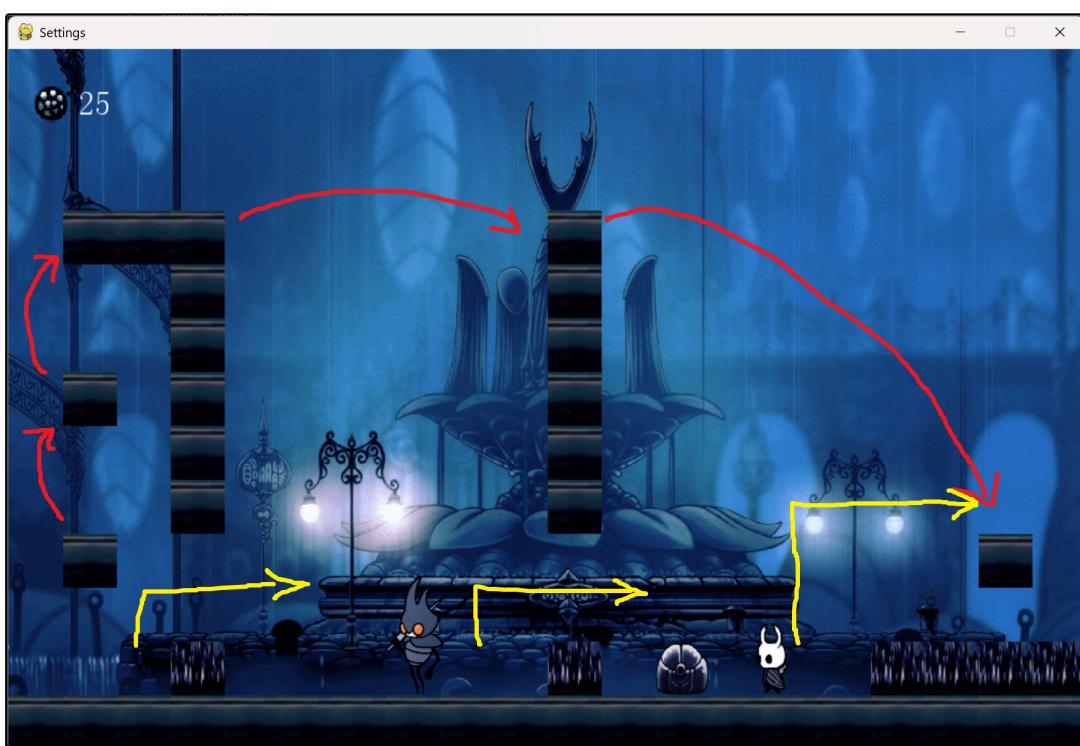


- 普通敌人：游戏中的普通敌人共有三种：泪城守卫、持盾守卫、黄蜂女。另外，由于我们的战斗系统**支持多波次战斗**，所以还有泪城守卫+持盾守卫的组合。普通敌人中持盾守卫具有特殊机制。这些怪物在 json 中被部署，并且在 `MapPage.py` 中被部署到地图中。你可以在 `/save/` 中看到每个关卡的配置以及我方单位的配置。**配置文件包含了全部与战斗有关的信息**（生命、精力、使用的卡和卡的信息……）。修改这些文件相当于修改当局游戏的战斗信息。修改 `/save/origin` 下的文件相当于永久修改信息。
- 特殊敌人 (boss)：本游戏的 boss 为辐光，具有特殊机制。
 - 特殊的动画效果。辐光有和普通敌人不一样的入场动画。
 - **防御-反击**机制：当辐光使用防御卡并且防御成功时，将会在下一回合使用强力卡。这是非常具有威胁性的机制，为了防止初见杀严重，我们在普通敌人“持盾守卫”上加入了同样的机制。
 - **可打断治疗**机制：辐光会使用一种卡治疗自己。若这张卡被进行了逻辑门操作，将会使治疗失效。

游戏机制

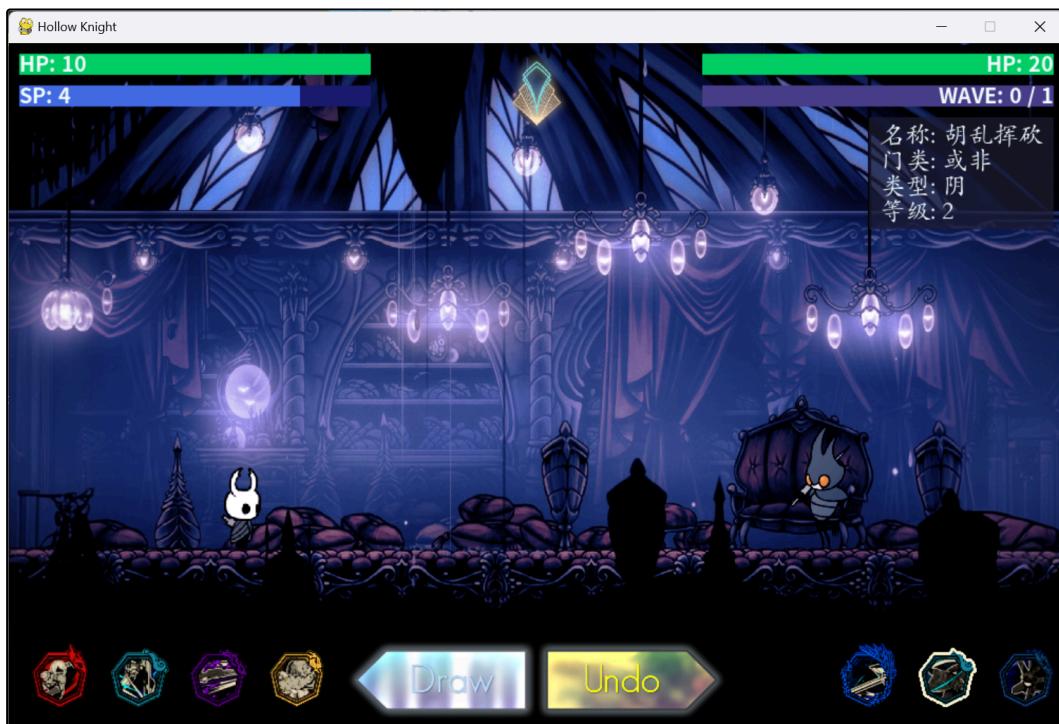
本作拥有丰富的游戏机制。

1. **地图上的核心机制：**平台跳跃机制。本游戏的 `map1` 和 `map2` 由制作者**手工制作**，体现了一片匠心，毫无随机AI生成成分。在保证了优良的游戏性的同时，**有些地方甚至有多种解法**，且最直接的解法未必是最容易操作的，比如：

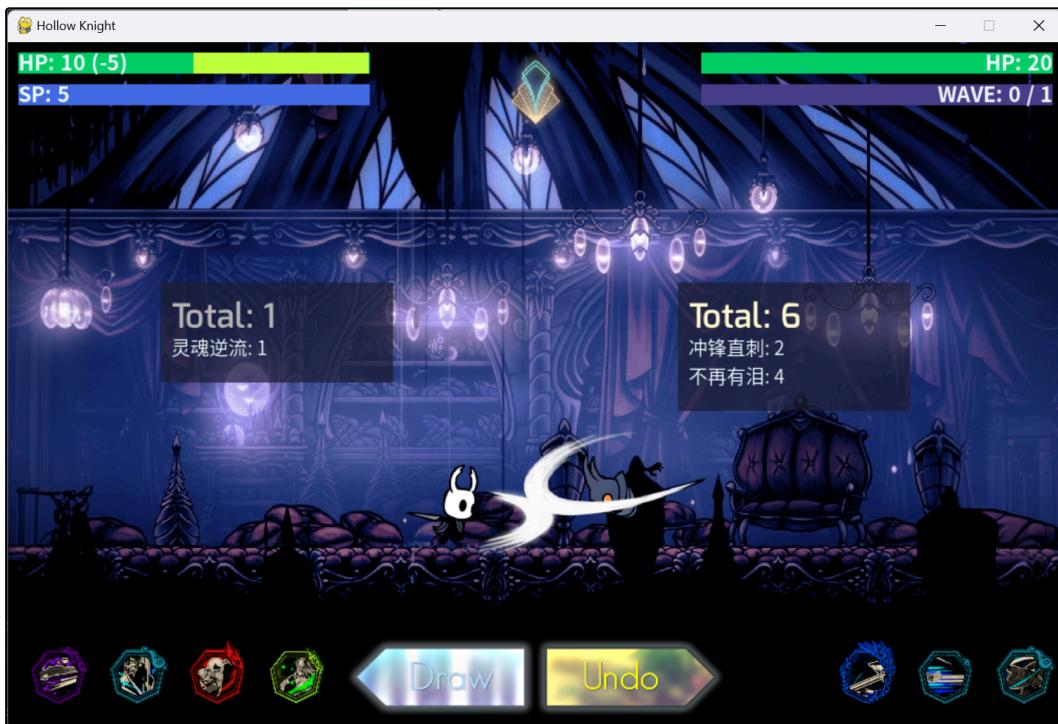


图中两种颜色的路径都可以通过此处，但是红色线看似迂回，实则容易：先用三段左右摇摆跳上顶部，之后接两次简单的跳跃+冲刺即可到达右侧方块。而黄色线需要两次看准时机从刺上跳过的同时，在第三个跳跃上会遇到非常大的难题：人物的最高跳跃极限高度刚好是 3 格，而右侧的方格最高高度也为 3 格，这也就意味着玩家需要精准地在人物跳到最高点的时候按下冲刺。虽然制作者多次亲测后只要稍加尝试找到手感依然可行，但确实不如红色线跳法简单。

2. 战斗上的核心机制：卡牌战斗机制。本游戏的卡牌战斗规则原创，并且非常符合 Si100B 人的口味。我们使用七种颜色来对应七种逻辑门（not, and, or, nand, nor, xor, xnor）。并给卡牌附上了阴阳两种属性。卡牌的具体信息可以通過鼠标移到到卡上查看：



出牌后，点击上方按钮或按空格键结束回合开始拼点。拼点时，根据双方阳属性卡的等级和的比较、阴属性卡的等级和的比较，计算伤害（阴、阳分别计算）例如下面的情况：



图示为阳属性卡的拼点，你可以看到双方参与拼点计算的卡牌信息，以及各自卡牌的点数总和。左上角实时现实了我方本回合将会受到的伤害（伤害将在阴阳拼点都结束后统一结算）。

使用卡牌和使用 Draw 按钮抽卡都会消耗精力 (SP)。精力在每回合开始时自然恢复一定数值。

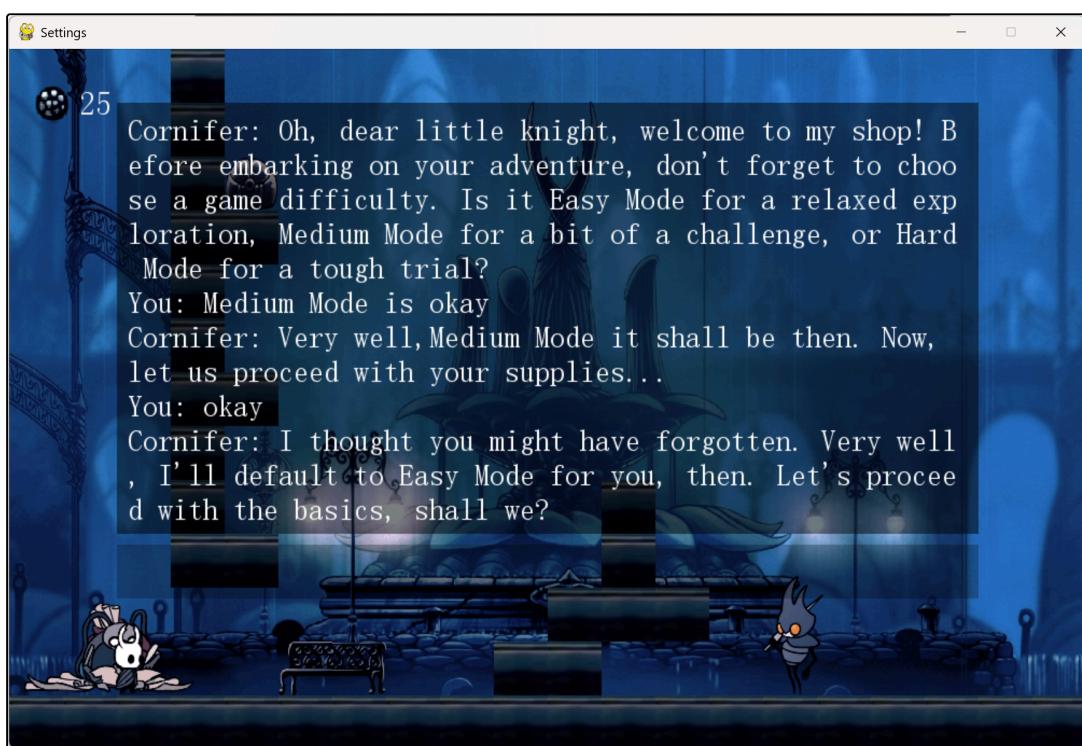
3. **碰撞系统**: 作为平台跳跃游戏，我们自然有**良好的碰撞系统**。不管是人物的**上下左右**哪一面碰到了实体方块，都会无法继续朝该方向移动。如果在空中，碰撞会使得上升速度变为 0，并且开始下坠。这部分主要在 `MapPage.py` 里实现。

4. **资源机制**: 本游戏的资源机制主要体现在吉欧（游戏内货币）上。玩家初始没有吉欧，可以通过两种方式获得吉欧。

- 打败敌人获得吉欧：打败不同难度的敌人可以获得不同的吉欧。由于这部分比较简短，在 `GameManager.py` 对 `battle` 结果的响应中直接处理完成。
- 也可以通过开启宝箱获得吉欧，这部分同样很简短，在 `MapPage.py` 的 `handle` 函数中直接处理完成。
- **随机地图 map0**: 本游戏设置有随机地图 `map0`，玩家每次进入其中，其中地形都会发生变化。其中分布有宝箱和敌人，可以通过反复退出进入来刷取资源。
- **商店价格递增**: 为了控制数值膨胀，我们对商店的购买价格进行了递增处理（每次*1.2）。

LLM Agent 的应用

1. **对话系统**: 在与两位友善 NPC 有模型接触的情况下按 `E` 可以弹出对话框进行对话。两位 NPC 由 AI 驱动。该对话框实现了自动换行、上下翻动（按键盘方向键的上下即可）、保存历史纪录等功能。按 `Esc` 可以退出对话。先知作为最终 boss 前的 NPC，会**提醒玩家前方是最终 boss 辐光**（通过 prompt 实现）。prompt 和对话框等内容在 `ChatBox.py` 中可以看到详细实现。



2. 决策系统：与柯尼法对话，可以**调整游戏的难度**（只要输入中带有 Easy, Medium, Hard 等字眼即可触发调整难度，调整成功时，运行的终端会输出相应提示）。上图中玩家调成了中等难度，同时也可以看出对话框实现了上下滚动和自动换行功能。

GamePlay

1. **制作精美的菜单：**我们有制作精美的菜单，包含了进入游戏、设置、帮助、退出游戏的**多个功能按键**。鼠标点击即可进入游戏，无需繁琐的键盘操作。同时，还**贴心地增加了调节音量功能**，在“设置”中单击音量条则可调整音量，默认音量 0.5。这部分的内容在 `Menu.py`、`SettingPage.py` 和 `HelpPage.py` 里实现。
2. **精心挑选、花样繁多的 bgm：**我们有风格统一、数量繁多的 bgm！从耳熟能详的空洞骑士主界面 bgm 到辐光 boss 战神圣激昂的音乐，我们对音乐的要求从不妥协。bgm 的切换以及音量的调节在 `Utility.py` 里的 `BgmPlayer` 类中实现。

Code

1. **可读性：**我们的代码有大量的注释，且代码风格进行了统一化处理。
2. **代码设计：**在代码结构上，游戏由 `GameManager.py` 通过每帧对场景的渲染和对事件队列中事件的处理进行控制，通过对 `listeners` 队列进行操作来切换场景和界面。战斗部分单独由 `battleController.py` 进行控制。同时我们撰写了简单的 `classes.md`，也就是类的大纲。

创意

我们对前文出现过的创意进行汇总：

1. 平台跳跃+卡牌战斗，丰富的机制与可玩性。
2. 战斗上我们有大量的动画。
3. 地图中采用了**真实的重力加速度系统**，并且给小骑士**添加了冲刺动画**。
4. 添加 `map0` 作为肉鸽地图，每次随机重新生成，用于刷钱刷怪。
5. **手工制作的地图**，保证了优良的游戏性的同时，**鼓励多种解法**。
6. 在菜单中**增加了调节音量功能**。
7. **商店价格递增：**为了控制数值膨胀，我们对商店的购买价格进行了递增处理。（1.2 蠕变指数）