

# REPORT: BunnyKillsMommy

Introduction: 这是一款兼养成解谜于一体的剧情向像素游戏，玩家通过控制主角“Bunny”探索小镇“The Great-Great Town”发生的故事，帮助村民斩杀“邪恶势力”，净化水源的故事。

第一天，按照村长的指示打怪赚取金币购买种子，在菜地播种，并回家接食物补充血量；第二天，继续打怪获得金币用来提升自己的属性（血量、攻击力、防御力）并收获植物合成神秘药水；第三天，用前两天打怪时掉落的碎片引出污染水源的“罪魁祸首”，并将它击杀。

Implementation:

**Scene:**

(1) We have **3 different scenes**: City, Wild and Home.

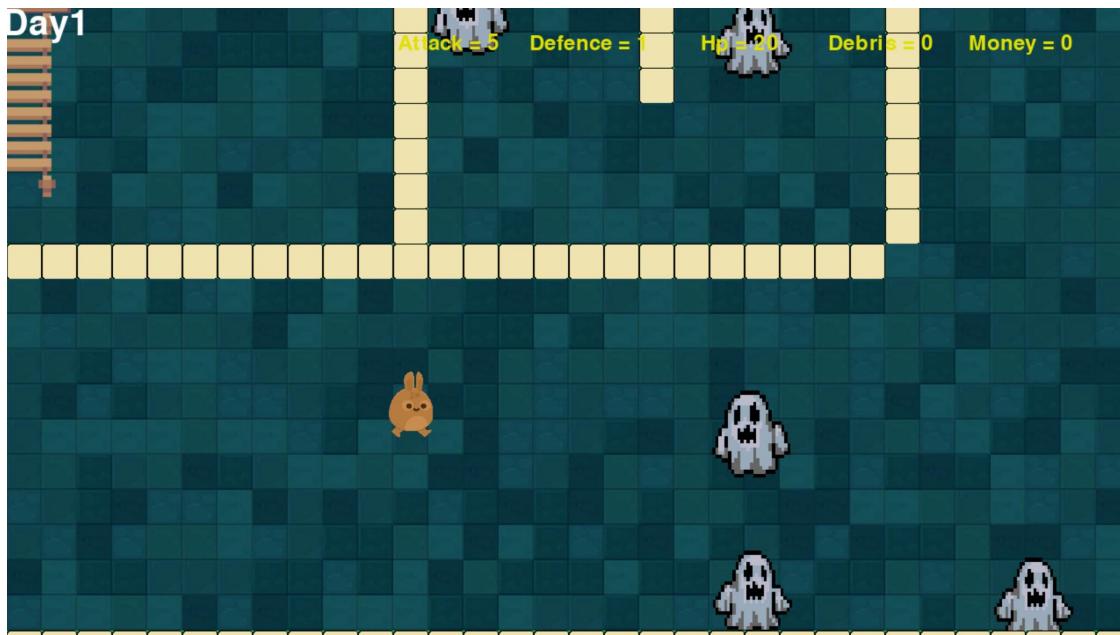
关于切换 Scene 的实现：

1. 在 menu 界面通过 enter 改变 state 为 CityScene
2. 玩家通过与“井”碰撞切换 CityScene 和 WildScene
3. 玩家与“树屋”碰撞进入“家”，在家中走到“门”的位置回到 CityScene
4. 在接金币时 click mouse 以切换 README (Scene)，再次 click 返回 HomeScene 且保留离开时的游戏进度

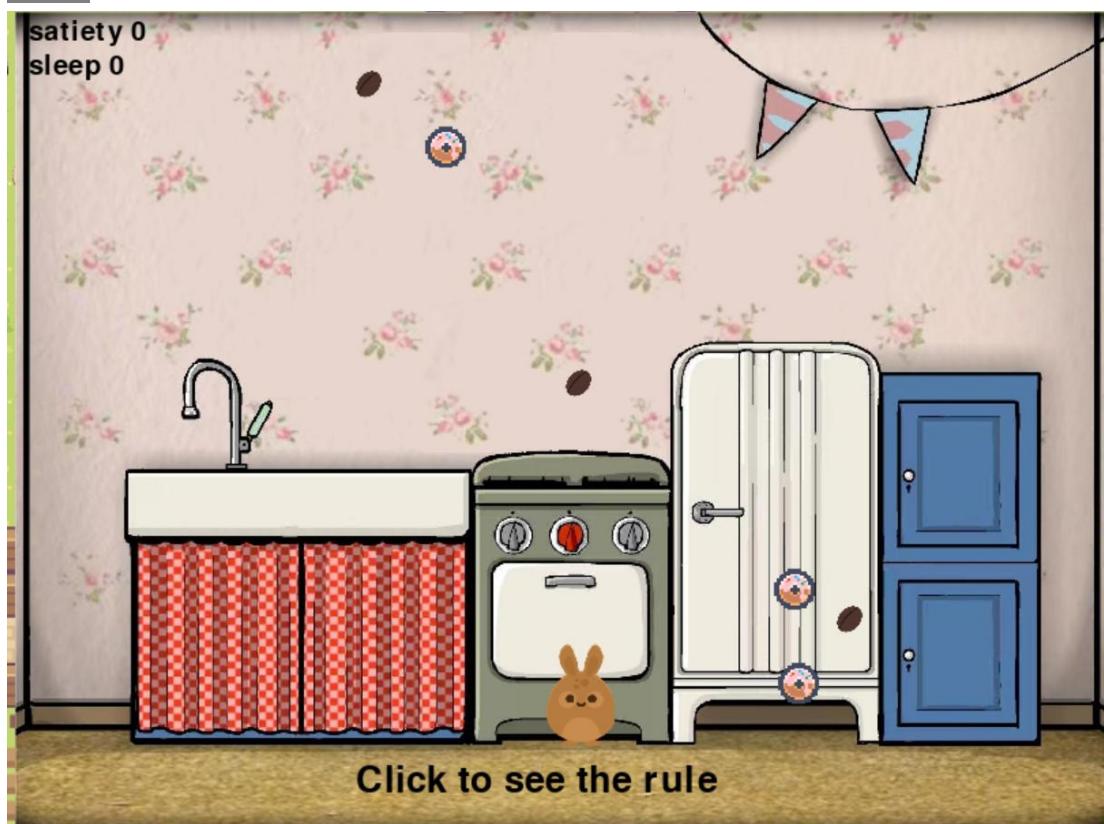
City:



Wild:



Home:

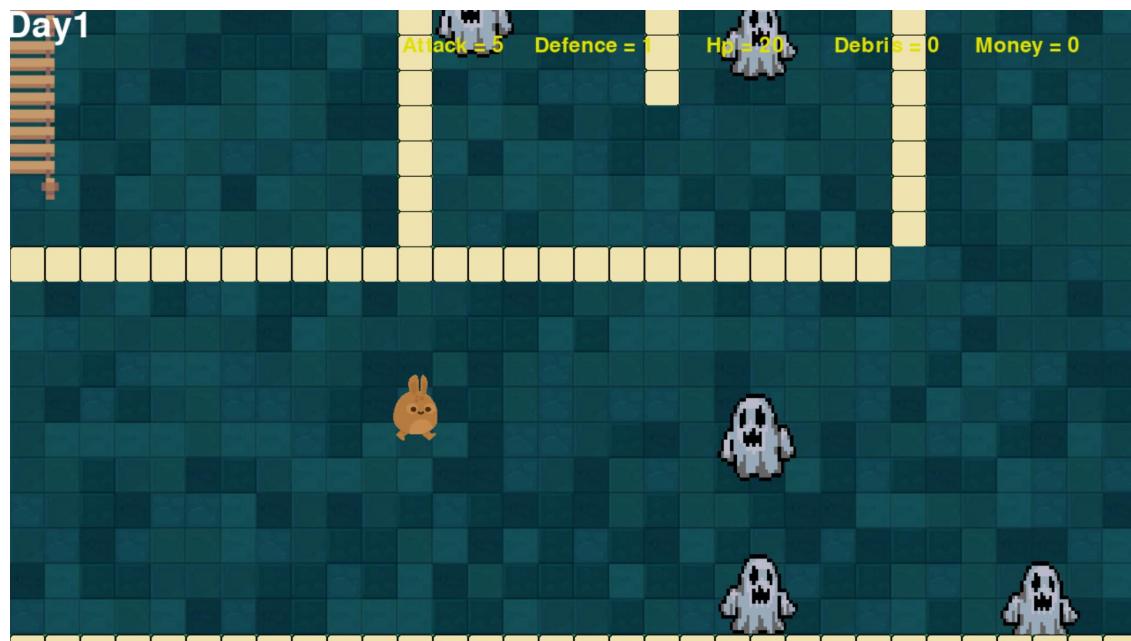


(2) We have a **large scene**——the wild scene——**with the camera following**. (The relevant code is in "wild scene".)

关于 camera 的实现：

首先确定 player 应该可以在屏幕上随意移动而不是被固定在中间，其次确定 player 移动时其他景物（怪物，portal，地图）都都会往反方向移动（包括他们的 rect 以及他们的活动范围），而 player 的移动范围应为以 portal 的中心为中心的一定范围（大于屏幕尺寸），当 player 碰壁时，定义为被墙壁上的蜘蛛丝等束缚住（速度变为 0），只能通过摁 ENTER 返回出发点，而此时其他的景物也应

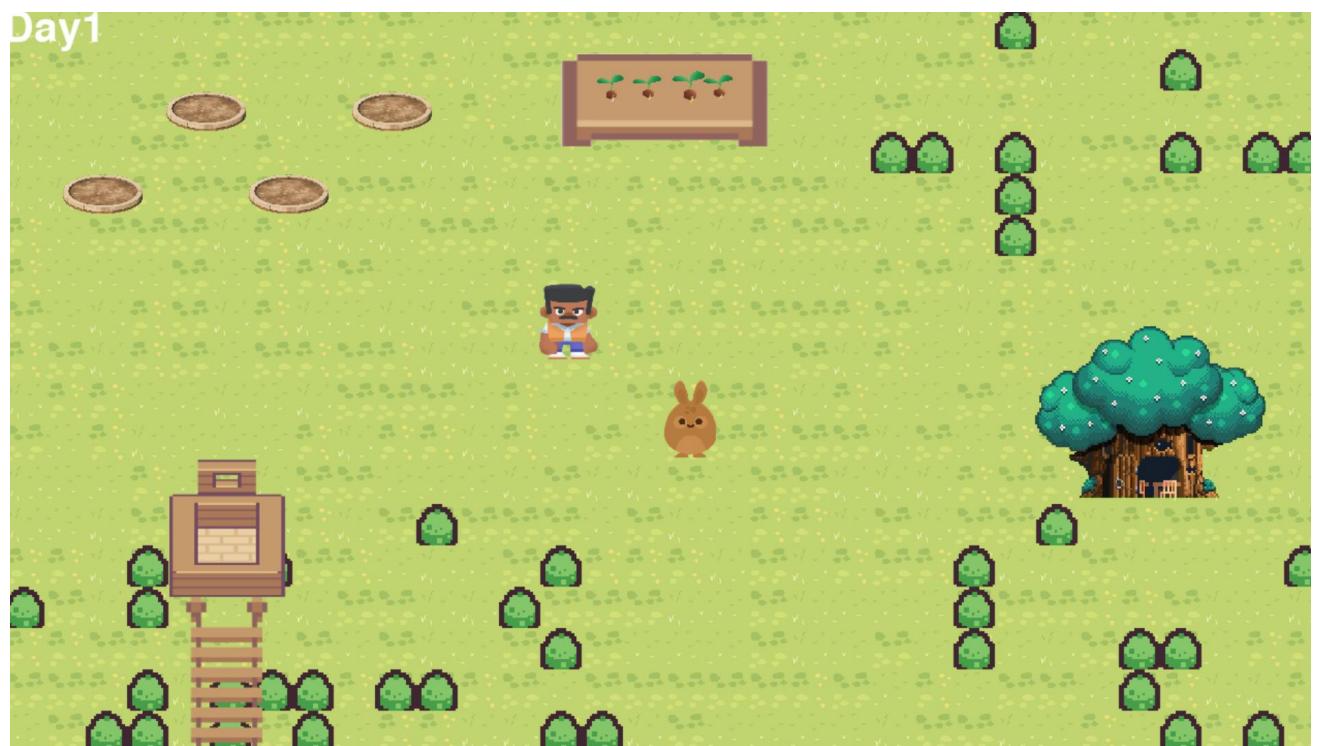
该返回最初的位置 (code 中可以看到怪物位置 (each.x, each.y) 以及活动范围的 rect (centerx, 以及 initialposition) 的刷新)



### (3) Interactive objects: obstacles in city scene , portal and 2 different foods at home.

【Map.py 中的 gen\_obstacles() 用于生成城市里的障碍物，用 get\_obstacles 生成野外的障碍物； main.py 中的 flush scene 和 Scenemanager 中的碰撞检测函数中构建了传送门； NPC.py 中的 Class Coin 用于生成食物。】

大多数碰撞通过使 player.move (-dx, -dy) 实现，少部分碰撞（和怪物的）通过怪物速度变为 0 实现（让他们不会重叠到一起）



## Characters:

- (1) Have a **main character** "Bunny".
- (2) **2 different friendly NPCs**: village Chief and shopping desk. 通过定义 NPC 中的 NPC 类以及 deck 类来实现
- (3) **A simple enemy**: the ghosts underground
- (4) **A special enemy**: A boss that used to be Bunny's mom.
  - 关于 monster (normal) 和 boss (special) 方面：  
特殊怪和一般的产生函数都是 gen\_monster, 根据输入的参数来判断生成哪种怪, 他们的 image 和运动方式, 以及攻击方式还有大小, 血量, 防御, 攻击力都不一样。除了攻击特效之外的都在 boss 和 monster 的类中定义, 而攻击特效则在 battlebox 中体现, 所以给 battlebox 增加 monstertype 的参数, 来区分两种不同的战斗动画, 自动判断怪物类型靠的是 player 的碎片属性。
- (5) **Vegetables and fruit**: 3 different vegetables and fruit. (通过定义 Field 中的 fruit 类来实现)

## Game Mechanics:

- (1) **Core mechanics** (In Main.py): We can obtain gold coins by attacking ghosts underground, using gold coins to improve yourself and buy seeds for planting and harvest to obtain potions. Finally, we can defeat the boss and unlock an unexpected ending.

1.接金币、躲避: 通过 coin 类产生 coin 精灵 (两种 food), 当游戏没有失败或成功时, player 与他们碰撞时会根据他们的总类加不同的属性, 然后必须接到一定范围内, 不可超过那个范围, 一旦接近最大值就需要开始躲避, 首先时确保屏幕中有均匀分布的 6 个 food, 一类 food 有 3 个 (通过设置掉落 rect 范围来确保他们均匀掉落), 若 player 或地板 (其实就是他们的 rect.y>一定数值) 一旦与他们碰撞, 他们就会回到最高点并重置 rect.x。每次点击鼠标就可以看到规则 (切换 README 这个 GAMSTATE 和 Scene)。若吃过多某种食物, 通往外界的 portal 就不会打开, 会要求点击右键重启游戏 (重新 flush home scene, 并且数据清零), 若成功, 则可以去往 cityscene, 并且数据清 0.

2.打斗: 通过 battlebox 进行自动打斗, 每次动画播放一轮就扣掉相应被打的对象的血量, 等一只怪被打完后 (播放完一次完整的 battlebox) 会更新 player 的血量, 并给 player 加钱和碎片, 然后再反映到右上角的属性栏目里

3.种植: Day1 时, 首先玩家在 desk 购买自己想要的种子 (此时 player 的对应属性增减), 再通过与 field 碰撞选择自己想要种植的种类 (player 对应属性增减), blit 种植的动画与种子。Day2 时, 玩家与 field 的碰撞取消, 以便于与果蔬碰撞 (True), 此时 player 的 fruits 属性+1, 继而可在 desk 中凭 fruits 数量购买 potion。

- (2) **Collision system** (In SceneManager.py): We add Collision function to detect collisions between characters and village chief, monsters, houses, shopping desk and vegetable fields.

关于统一碰撞检测的实现:

首先罗列碰撞对像, 其次让 player 与他们碰撞时都实现反弹 (阻止他们一直碰撞, 而反弹的一部分相关属性如 dx, dy 都是 player 的固有属性, 在 palyer 中有写到) 然后再写特殊的碰撞触发的事件 (各种 check 函数, 而其中 blit 出的东西都用类先写好然后 import 即可, 这样其他文件不会出现碰撞), 传送门, 以此实现碰撞与其他部分的隔离

- (3) **Resource system:** Including various seeds, fruits, potions, and other resources related to farming, as well as health, attack, defense, and other resources related to monster hunting, as well as resources related to receiving food such as donuts. 【code 见 shoppingbox 以及 check\_event\_shopping】

## Gameplay:

- (1) **Main menu:** As shown in the figure. Click the Enter key to enter the game.



实现：

通过定义名为 Mainmenu 的 scene 和 gamestate 来实现 (blit 大图+blit 文字)，一旦打开游戏就会定义默认 state 为 Mainmenu

- (2) **BGM:** K.K.House (Main menu and city), The Last Battle (wild), Prelude (Home), Loser (Lose), Thunder (Bunny's Attack), Yell (When Bunny was injured by monster), Fire(Boss's Attack).

Bgm 的实现：

1.背景音乐：通过加在 Main 中的传送相关代码中实现在场景转换时 bgm 的切换；

2. 打斗音效：通过加在 BattleBox 中来实现；3.结算画面音乐：加在相应的结算场景显示的后面

### Creativity:

1. 运用动画：在主角与怪兽打斗时为主角和 boss 分别加上了攻击动画，在种菜时加入了动画，在离开家时加上了开门动画。
2. 运用音效：为打斗动作添加音效，增加趣味性。
3. 创新的机制：在单调的打怪之外，我们引入了种菜收获合成神秘药水的机制和在家接食物补充 Hp 的机制，让游戏的玩法更加多元。
4. 结尾动画：在结尾用动画实现反转并进行结算

关于结算页面的实现：

1.blit 相应结局的 image，其中结局“用药水拯救妈妈”是 blot 一段动画

2. 在结算页面全部 blot 结束后，press ENTER 以退出游戏

5. 多样的结局：针对玩家故事线的发展设置了四种不同的结局：

(1) 在打怪过程中血量降为零：Bunny 被杀死，未能拯救村庄，也未能救妈妈。

(2) 成功击败 boss 但没有合成神秘药水：仅仅拯救了村庄，并未揭露 boss 的真实身份以及村长的丑恶面目。

(3) 成功击败 boss 并使用了神秘药水：Boss 变回了 mom 的样子，Bunny 得知了污染水源的真相，惩罚了村长一家，Bunny 与 mom 团圆。

(4) 在三天内未能击败 boss，甚至未收集够碎片：未能拯救村庄，也没能救妈妈。

6. 迷宫的设计：

迷宫的实现：采用生成障碍物的方法，将障碍物拼接起来形成迷宫地图，并添加碰撞检测。值得注意的是，因为我们的大地图是采用检测是否按住方向键来移动，这使得在遇到障碍物时主角停在原地但地图仍然向前移动甚至让人出框的情况。为了防止它发生，我们为 player 设置了一个变量 coll\_w，当产生碰撞时，coll\_w 发生变化，使 camera 暂时停止运作，等碰撞结束之后在继续正常运行。为了卡好时间我们设置了延时，彻底避免了碰撞时地图的移动。