

游戏背景

目标

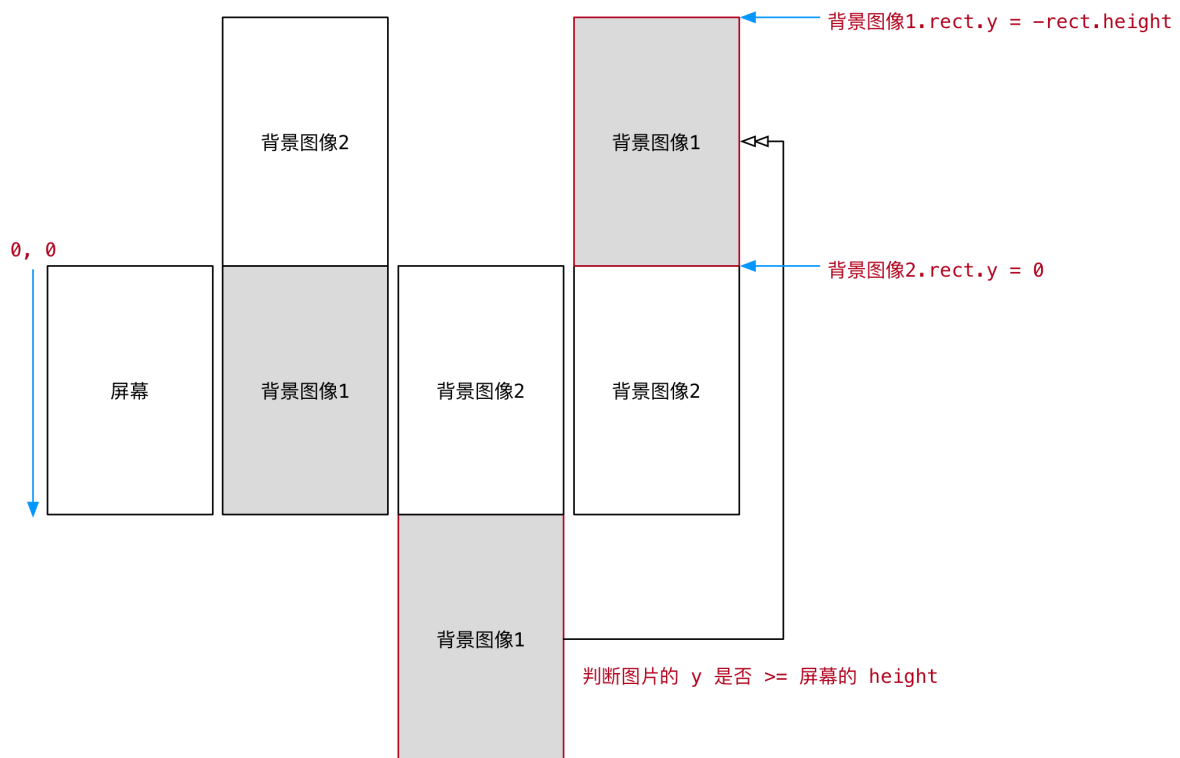
- 背景交替滚动的思路确定
- 显示游戏背景

01. 背景交替滚动的思路确定

运行 备课代码，观察 背景图像的显示效果：

- 游戏启动后，背景图像 会 连续不断地 向下方 移动
- 在 视觉上 产生英雄的飞机不断向上方飞行的 错觉 —— 在很多跑酷类游戏中常用的套路
 - 游戏的背景 不断变化
 - 游戏的主角 位置保持不变

1.1 实现思路分析

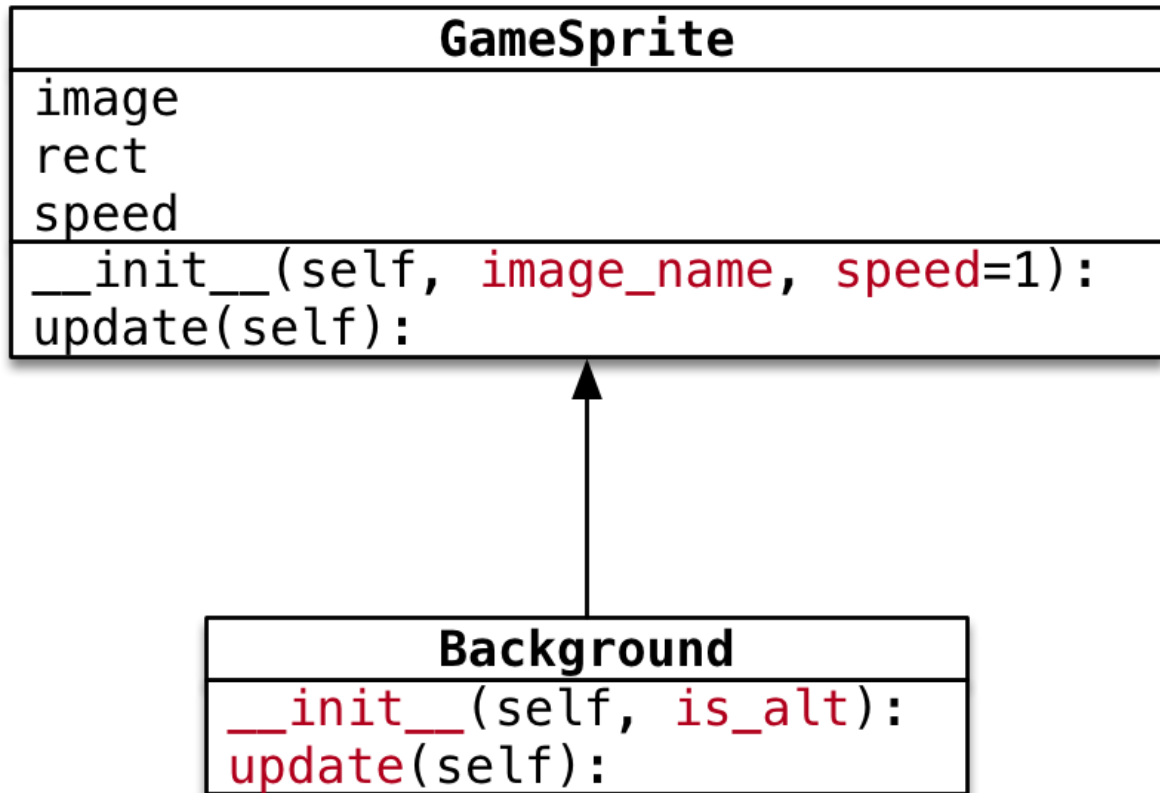


解决办法

1. 创建两张背景图像精灵
 - 第 1 张 完全和屏幕重合
 - 第 2 张在 屏幕的正上方
2. 两张图像 一起向下方运动
 - `self.rect.y += self.speed`
3. 当 任意背景精灵 的 `rect.y >= 屏幕的高度` 说明已经 移动到屏幕下方
4. 将 移动到屏幕下方的这张图像 设置到 屏幕的正上方

- `rect.y = -rect.height`

1.2 设计背景类



- 初始化方法
 - 直接指定 背景图片
 - `is_alt` 判断是否是另一张图像
 - `False` 表示 第一张图像，需要与屏幕重合
 - `True` 表示 另一张图像，在屏幕的正上方
- `update()` 方法
 - 判断 是否移动出屏幕，如果是，将图像设置到 屏幕的正上方，从而实现 交替滚动

继承 如果父类提供的方法，不能满足子类的需求：

- 派生一个子类
- 在子类中针对特有的需求，重写父类方法，并且进行扩展

02. 显示游戏背景

2.1 背景精灵的基本实现

- 在 `plane_sprites` 新建 `Background` 继承自 `GameSprite`

```

1 class Background(GameSprite):
2     """游戏背景精灵"""
3
4     def update(self):
5
6         # 1. 调用父类的方法实现
7         super().update()
8
9         # 2. 判断是否移出屏幕，如果移出屏幕，将图像设置到屏幕的上方
10        if self.rect.y >= SCREEN_RECT.height:
11            self.rect.y = -self.rect.height
12

```

2.2 在 plane_main.py 中显示背景精灵

1. 在 `__create_sprites` 方法中创建 精灵 和 精灵组
2. 在 `__update_sprites` 方法中，让 精灵组 调用 `update()` 和 `draw()` 方法

`__create_sprites` 方法

```

1 def __create_sprites(self):
2
3     # 创建背景精灵和精灵组
4     bg1 = Background("./images/background.png")
5     bg2 = Background("./images/background.png")
6     bg2.rect.y = -bg2.rect.height
7
8     self.back_group = pygame.sprite.Group(bg1, bg2)

```

`__update_sprites` 方法

```

1 def __update_sprites(self):
2
3     self.back_group.update()
4     self.back_group.draw(self.screen)

```

2.3 利用初始化方法，简化背景精灵创建

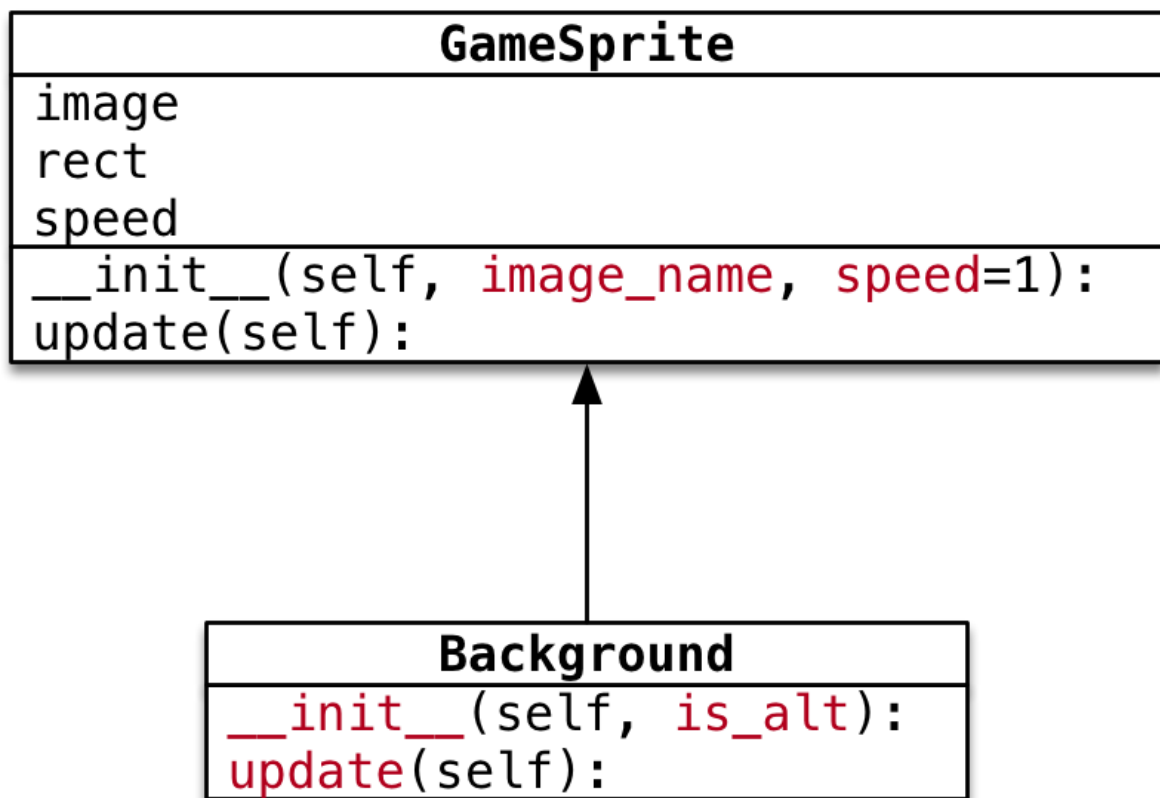
思考 —— 上一小结完成的代码存在什么样的问题？能否简化？

- 在主程序中，创建的两个背景精灵，传入了相同的图像文件路径
- 创建 第二个 背景精灵 时，在主程序中，设置背景精灵的图像位置

思考 —— 精灵 初始位置 的设置，应该 由主程序负责？ 还是 由精灵自己负责？

答案 —— 由精灵自己负责

- 根据面向对象设计原则，应该将对象的职责，封装到类的代码内部
- 尽量简化程序调用一方的代码调用



- 初始化方法

- 直接指定 背景图片
- `is_alt` 判断是否是另一张图像
 - `False` 表示 第一张图像，需要与屏幕重合
 - `True` 表示 另一张图像，在屏幕的正上方

在 `plane_sprites.py` 中实现 `Background` 的 初始化方法

```
1 def __init__(self, is_alt=False):
2
3     image_name = "./images/background.png"
4     super().__init__(image_name)
5
6     # 判断是否交替图片，如果是，将图片设置到屏幕顶部
7     if is_alt:
8         self.rect.y = -self.rect.height
```

- 修改 `plane_main` 的 `__create_sprites` 方法

```
1 # 创建背景精灵和精灵组
2 bg1 = Background()
3 bg2 = Background(True)
4
5 self.back_group = pygame.sprite.Group(bg1, bg2)
```