

1. pygame 安装

Pygame 是跨平台 Python 模块，专为电子游戏设计，包含图像、声音。建立在 SDL 基础上，允许实时电子游戏研发而无需被低级语言（如机器语言和汇编语言）束缚。

Pygame 虽然很少用于商业性的游戏项目,但是对于 Python 学习的做一些简单的 2D 游戏已经够用,其中官网地址:<https://www.pygame.org/> 下有很多使用 pygame 开源的项目可以供学些和运行

安装方式: `pip install pygame`

`import pygame` 检验是否安装成功

2. 植物大战僵尸

2.1. 资源准备

背景图片

豌豆

炮弹

僵尸

2.2. 游戏界面搭建

```
import pygame
# 设置窗口的宽度
WIDTH = 1200
# 设置窗口的高度
HEIGHT = 600
if __name__ == '__main__':
    # 初始化 pygame
    pygame.init()
    # 显示界面 窗口的大小
    screen = pygame.display.set_mode((WIDTH, HEIGHT))
    # 设置背景图片
    background_image = pygame.image.load("./res/background.png")
    # 调整背景图片大小 设置和窗口一样大小
    scale_background_image = pygame.transform.scale(background_image, (WIDTH, HEIGHT))
    # 获取到背景图片的位置和大小 (x,y,width,height) 矩形区域
    scale_background_image_rect = scale_background_image.get_rect()
    while True:
        # 设窗口设置背景图片
        screen.blit(scale_background_image, scale_background_image_rect)
        # 重新显示窗口 显示图片
        pygame.display.update()
```

2.3. 豌豆显示

定义一个豌豆类

```
# 定义一个豌豆对象
class Peas:
    # 初始化方法
    def __init__(self):
        # 加载一个豌豆图片
        self.image = pygame.image.load("./res/peas.gif")
        # 获取到豌豆图片的大小和位置
        self.image_rect = self.image.get_rect()
        # 设置豌豆显示的位置
        self.image_rect.top = 285
        self.image_rect.left = 30
        # 判断豌豆是否向上移动
        self.is_move_up = False
        # 判断豌豆是否向下移动
        self.is_move_down = False

    def display(self):
        """显示豌豆的方法"""
        screen.blit(self.image, self.image_rect)

    def move_up(self):
        """豌豆向上移动"""
        if self.image_rect.top > 30:
            # move_ip 可以改变图片的显示对 x,y 的坐标位置
            self.image_rect.move_ip(0, -10)

    def move_down(self):
        """豌豆向下移动"""
        if self.image_rect.bottom < HEIGHT - 10:
            self.image_rect.move_ip(0, 10)
```

创建一个豌豆对象

```
# 创建一个豌豆对象
peas = Peas()
```

在循环中调用豌豆的显示方法

```
peas.display()
```

2.4. 通过键盘控制豌豆上下移动

定义一个键盘监听的处理函数

```
def key_control():
```

```
# 获取到所有的事件
for event in pygame.event.get():
    # 如果事件类型是退出, 点击关闭按钮
    if event.type == QUIT:
        # 退出 pygame
        pygame.quit()
        # 退出程序
        exit()
    # 事件类型是键盘按下
    if event.type == KEYDOWN:
        # 向下键
        if event.key == K_UP:
            # 设置豌豆的移动方向
            peas.is_move_up = True
            peas.is_move_down = False
        elif event.key == K_DOWN:
            peas.is_move_up = False
            peas.is_move_down = True
    # 事件类型是键盘松开
    if event.type == KEYUP:
        if event.key == K_UP:
            peas.is_move_up = False
        elif event.key == K_DOWN:
            peas.is_move_down = False
```

在显示完图片和豌豆以后调用事件监听函数,并且根据操作状态移动豌豆

```
key_control()
if peas.is_move_up:
    peas.move_up()
if peas.is_move_down:
    peas.move_down()
```

因为在移动的过程速度较快,对于动画的设置使用 pygame.time 中的 Lock

```
# 定义一个时钟对象
clock = pygame.time.Clock()
```

在循环中刷新图片后使用延时

```
clock.tick(60)
```

2.5. 通过键盘控制发射炮弹

创建一个炮弹对象

```
class Bullet:
    # 所有的炮弹信息
    bullet_list = []
    # 创建炮弹间隔
    interval = 0

    def __init__(self, peas):
        # 加载一个豌豆图片
        self.image = pygame.image.load("./res/bullet.gif")
        # 获取到豌豆图片的大小和位置
        self.image_rect = self.image.get_rect()
        # 设置豌豆显示的位置
        self.image_rect.top = peas.image_rect.top
        self.image_rect.left = peas.image_rect.right

    def display(self):
        """显示炮弹的方法"""
        screen.blit(self.image, self.image_rect)

    def move(self):
        """移动炮弹"""
        self.image_rect.move_ip(8, 0)
        if self.image_rect.right > WIDTH - 20:
            Bullet.bullet_list.remove(self)
        else:
            for z in Zombie.zombie_list[:]:
                if self.image_rect.colliderect(z.image_rect):
                    Zombie.zombie_list.remove(z)
                    Bullet.bullet_list.remove(self)
                    break
```

在豌豆中定义发射炮弹的方法:

```
def shout_bullet(self):
    """发射炮弹方法"""
    # 创建一个炮弹
    bullet = Bullet(self)
    Bullet.bullet_list.append(bullet)
```

2.6. 炮弹移动

```
def move(self):
    """移动炮弹"""
    self.image_rect.move_ip(8, 0)
    if self.image_rect.right > WIDTH - 20:
```

```
Bullet.bullet_list.remove(self)
else:
    for z in Zombie.zombie_list[:]:
        if self.image_rect.colliderect(z.image_rect):
            Zombie.zombie_list.remove(z)
            Bullet.bullet_list.remove(self)
            break
```

```
for bullet in Bullet.bullet_list:
    # 显示炮弹
    bullet.display()
    # 移动炮弹
    bullet.move()
```

2.7. 显示僵尸

定义一个僵尸类

```
class Zombie:
    # 所有的僵尸信息
    zombie_list = []
    # 创建炮弹间隔
    interval = 0

    def __init__(self):
        # 加载一个豌豆图片
        self.image = pygame.image.load("./res/zombie.gif")
        self.image = pygame.transform.scale(self.image, (70, 70))
        # 获取到豌豆图片的大小和位置
        self.image_rect = self.image.get_rect()
        # 设置豌豆显示的位置
        self.image_rect.top = random.randint(10, HEIGHT-70)
        self.image_rect.left = WIDTH

    def display(self):
        """显示炮弹的方法"""
        screen.blit(self.image, self.image_rect)

    def move(self):
        """移动僵尸"""
        self.image_rect.move_ip(-2, 0)
        if self.image_rect.left < 0:
            Zombie.zombie_list.remove(self)
        else:
            # 撞到豌豆
```

```
if self.image_rect.colliderect(peas.image_rect):
    pygame.quit()
    exit()

for b in Bullet.bullet_list[:]:
    if self.image_rect.colliderect(b.image_rect):
        Bullet.bullet_list.remove(b)
        Zombie.zombie_list.remove(self)
        break
```

```
# 每隔一段时间创建炮弹
Zombie.interval += 1
if Zombie.interval >= 15:
    Zombie.interval = 0
    Zombie.zombie_list.append(Zombie())
```

2.8. 僵尸移动

```
for zombie in Zombie.zombie_list:
    zombie.display()
    zombie.move()
```