



Pygame游戏设计

# 第05章 动画精灵

Python编程课

# 动画精灵

- **动画精灵**表示作为一个单位来移动和显示一组像素，它是一种图形对象。我们可以把动画精灵想成一个小图片-----可以在屏幕上移动的图形对象，并且可以与其他图形对象交互。
- 图像 (image)：为动画精灵显示的图片；
- 矩形区 (rect)：包含动画精灵的的矩形区域。
- Pygame的sprite模块提供了一个动画精灵基类，名为Sprite。一般，我们不会直接使用基类，而是基于pygame.sprite.Sprite来创建自己的子类。



- 下面，我们来完成一个例子，基于pygame.sprite.Sprite基类来创建我们的子类，命名为PlaneClass。
- ```
import sys
import pygame
from pygame.locals import *
#创建动画精灵的子类，命名为PlaneClass
class PlaneClass(pygame.sprite.Sprite):
    def __init__(self, img_file, location):
        super().__init__()#初始化动画精灵
        self.image = pygame.image.load(img_file) #加载图像
        self.rect = self.image.get_rect() #获取图像的矩形数据
        self.rect.left, self.rect.top = location #设置图像的初始位置
```



- 实例化PlaneClass。
- pygame.init()
- screen\_size = 480, 700  
screen = pygame.display.set\_mode(screen\_size)  
pygame.display.set\_caption( “AirplaneSprite” )  
background = pygame.image.load( “background.png” ).convert()  
screen.blit(background, (0, 0)) #只加载一次图片，因此先将背景加载好  
img\_airplane = “me1.png”  
airplanes = []  
for i in range(3):  
    location\_img = [i \* 100, 10] #每次循环产生一个不同的位置  
    airplane = PlaneClass(img\_airplane, location\_img) #在这个位置创建一个  
    airplane实例对象  
    airplanes.append(airplane) #把新创建的airplane添加到列表
- #依次将airplane显示在窗口不同的位置  
for each in airplanes:  
    screen.blit(each.image, each.rect)



- 程序退出。
- `running = True`  
`while` `running`:  
    **for** `event` **in** `pygame.event.get()`:  
        **if** `event.type == QUIT`:  
            `pygame.quit()`  
            `sys.exit()`  
    `pygame.display.flip()`



- 现在我们加载了多个动画精灵，怎样让动画精灵动起来呢？我们要给它们增加一个移动的技能，给PlaneClass类添加speed属性和move()方法。

- **class** PlaneClass(pygame.sprite.Sprite):
  - def** \_\_init\_\_(self, img\_file, location, speed): #增加speed参数
    - pygame.sprite.Sprite.\_\_init\_\_(self)
    - self.image = pygame.image.load(img\_file)
    - self.rect = self.image.get\_rect()
    - self.rect.left, self.rect.top = location
    - self.speed = speed #为PlaneClass类创建一个speed属性
- #定义move()方法来移动精灵
  - def** move(self):
    - self.rect = self.rect.move(self.speed)
    - #如果遇到边界，则移动速度反向
    - if** self.rect.left < 0 **or** self.rect.right > width:
      - self.speed[0] = -self.speed[0]
    - if** self.rect.top < 0 **or** self.rect.bottom > height:
      - self.speed[1] = -self.speed[1]



- 动起来...
- **from** random **import** \*
- .....[省略代码]
- screen\_size = width, height = 480, 700  
screen = pygame.display.set\_mode(screen\_size)  
pygame.display.set\_caption("AirplaneSprite")  
background = pygame.image.load("background.png").convert()

img\_airplane = "me1.png"

airplanes = []

speed = [2, 2] #初始化speed

**for** i **in** range(3):

location\_img = [i \* 100, 10]

speed = [choice([-2, 2]), choice([-2, 2])] #随机生成速度

airplane = PlaneClass(img\_airplane, location\_img, speed) #实例化对象

airplanes.append(airplane)



- `running = True`  
`while running:`  
    `for event in pygame.event.get():`  
        `if event.type == QUIT:`  
            `pygame.quit()`  
            `sys.exit()`  
    `pygame.time.delay(10)`  
    `screen.blit(background, (0, 0))`  
    `for each in airplanes:`  
        `each.move()` #调用move()方法, 移动精灵  
        `screen.blit(each.image, each.rect)`  
    `pygame.display.flip()`





尝试在同一个程序中  
加载不同的精灵。



- .....[省略代码]
- #定义所有精灵的文件列表  
imgs = [ “me1.png” , “enemy1.png” , “enemy2.png” ,  
“enemy3\_n1.png” ]  
airplanes = []  
speed = [2, 2] #初始化speed  
for i in range(4):  
    location\_img = [i \* 100, 10]  
    speed = [choice([-2, 2]), choice([-2, 2])] #随机生成速度  
    airplane = PlaneClass(imgs[i], location\_img, speed) #加载所有的飞机对象  
    airplanes.append(airplane)

