



Pygame游戏设计

# 第02章 surface对象、 rect对象

Python编程课

# Surface对象

- **surface**对象就是pygame中用来表示图像的对象，简单的说，它表示的就是图像。
- 上节课，我们用到了两个surface对象，我们分别把它们命名为screen和heroPlane。
- screen表示的是游戏窗口的背景图像，heroPlane是小飞机的图像。



# 创建surface对象

- 怎样创建surface对象？有两种方法：
- 第一种方法是，指定尺寸，创建一个空的surface对象。screen就是我们创建的一个固定尺寸、空的surface 对象；
- 第二种方法是，使用pygame.image.load()方法，加载一个实际的图像，括号中传入图像文件名，该方法会返回一个surface对象。heroPlane就是用这种方式创建的一个surface对象。

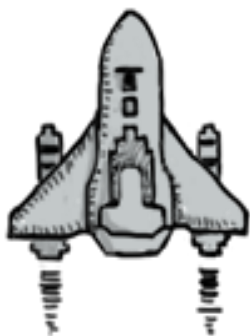


- 将一个图像绘制到另一个图像上是怎样实现的呢?
- 上节课我们通过调用blit()函数，将heroPlane图像绘制到了screen图像上，源代码如下：  
`screen.blit(heroPlane, position)`
- 难道真的是画上去的吗?



- 首先，我们先了解一下图像是什么样的？
- 图像是由**像素**组成的，当我们将一个图像放大时，会观察到图像是由很多个方块（像素点）组成的，如下：

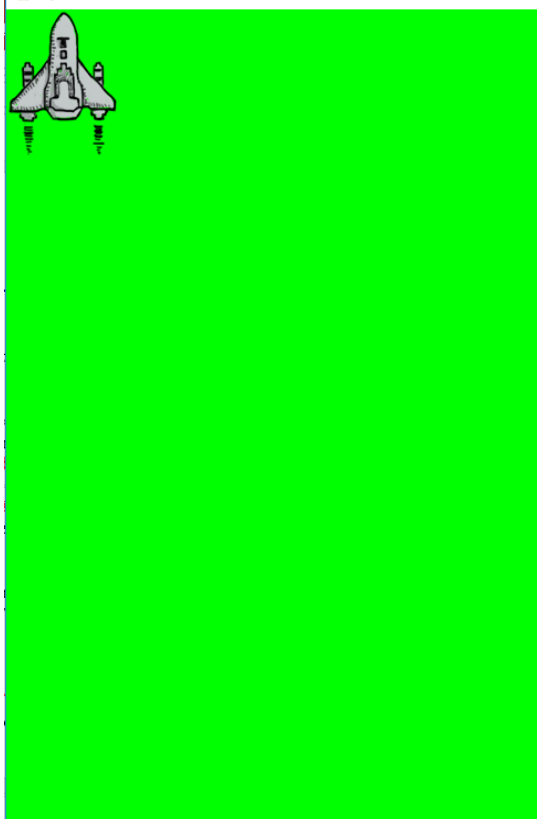
小飞机正常图



小飞机放大后的左翼图



- pygame将一个图像绘制到另一个图像上面，实际上是通过修改底图对应位置的像素点的颜色实现的，从而达到覆盖的效果。
- 我们上节课做的程序中，在screen图像上绘制小飞机图像，就是将小飞机所占区域像素点的绿色修改为了小飞机对应的像素颜色。



- 那么，**移动**一个图像的实现原理是什么呢？
- 获取图像移动后的新的位置；
- 填充游戏窗口背景，遮盖原来的图像；
- 在新的位置绘制图像；
- 更新游戏界面。
- **上节课的程序中，对应以上步骤的代码如下：**
- `position = position.move(speed)`  
`screen.fill(bg)`  
`screen.blit(heroPlane, position)`  
`pygame.display.flip()`



- **rect**对象

- **rect**对象是用来存储矩形坐标的pygame对象。

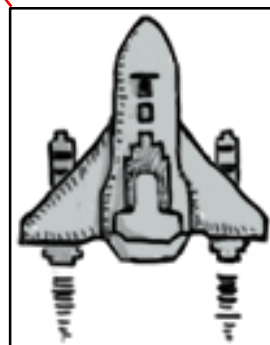
- rect对象表示的是一个矩形区域，如下面的小飞机图像。

`<rect(0, 0, 102, 126)>`

图像左上角的坐标点。

图像的大小width\*height。

刚加载时，左上角坐标为 (0,0)



width  
= 102个像素点

height = 126个像素点





- 一般来说在制定一个区域的时候，矩形是必须的，比如在屏幕的一部分画东西。在pygame中矩形对象极为常用，它的指定方法可以用一个四元素的元组，或者两个二元素的元组，前两个数为左上坐标，后两位为图像的width(宽)\*height(高)。
  - `Rect(left, top, width, height) -> Rect`
  - `Rect((left, top), (width, height)) -> Rect`
  - `Rect(object) -> Rect`
- 一旦有了Rect对象，我们就可以对其做很多操作，比如调整位置和大小，判断一个点是否在其中等等。



- 在上一节课的程序中，我们使用surface对象的一个方法 `get_rect()`，创建了一个 `rect` 对象，并命名为 `position`，用来获取 `heroPlane` 的坐标及图像大小。
- #创建 `heroPlane` 的 `rect` 对象，并命名 为`position`
- `position = heroPlane.get_rect()`
- 然后对该 `rect` 对象进行移动的操作。
- #调用 `rect` 对象的 `move()` 方法，调整位置，并返回新的 `rect` 对象，赋值给`position`
- `position = position.move(speed)`



- 上节课我们的小飞机飞走了（飞出了窗口.....），现在我们要把它困在窗口中，当小飞机遇到窗口边界时，就反向飞。想一想，要实现这个功能，需要哪些步骤？
- 小飞机（名字叫heroPlane的对象）要会飞（移动），这个我们上节课已经学过了；
- 小飞机要能知道自己飞到窗口边界了；
- 小飞机要会转向；



- 我们在上节课程序的基础上添加程序，上节课的程序已经可以让小飞机飞了。
- .....
- `speed = [1, 1]`    #设定小飞机移动的速度
- .....
- .....
- `position = position.move(speed)`    #更新小飞机的位置，`position`是一个`rect`对象
- `screen.fill(bg)`    #填充纯色背景，覆盖之前的图像  
`screen.blit(heroPlane, position)`    #将更新后的小飞机绘制在背景图上  
`pygame.display.flip()`    #更新整个画面，双缓存机制，前面先将背景和小飞机更新但不显示，使用`pygame.display.flip()` 同时更新所有画面



- .....
- `screen_size = width, height = 480, 700`
- `speed = [1, 1]` #设定小飞机移动的速度
- .....
- .....
- `position = position.move(speed)` #更新小飞机的位置, `position`是一个`rect`对象
- #如果小飞机的左侧坐标小于0或者右侧坐标大于窗口的宽度`width`, 则说明水平方向要飞出窗口了
- **if** `position.left < 0 or position.right > width:`  
     # `flip(Surface, xbool, ybool) -> Surface`, 水平或垂直翻转图像, 返回值赋值给`heroPlane`
- `heroPlane = pygame.transform.flip(heroPlane, True, False)`
- # 翻转后, 水平方向的移动速度反向  
     `speed[0] = -speed[0]`

垂直方向该怎么做?



- #如果小飞机的顶部坐标小于0或者底部坐标大于窗口的高度height, 则说明垂直方向要飞出窗口了

```
if position.top < 0 or position.bottom > height:
```

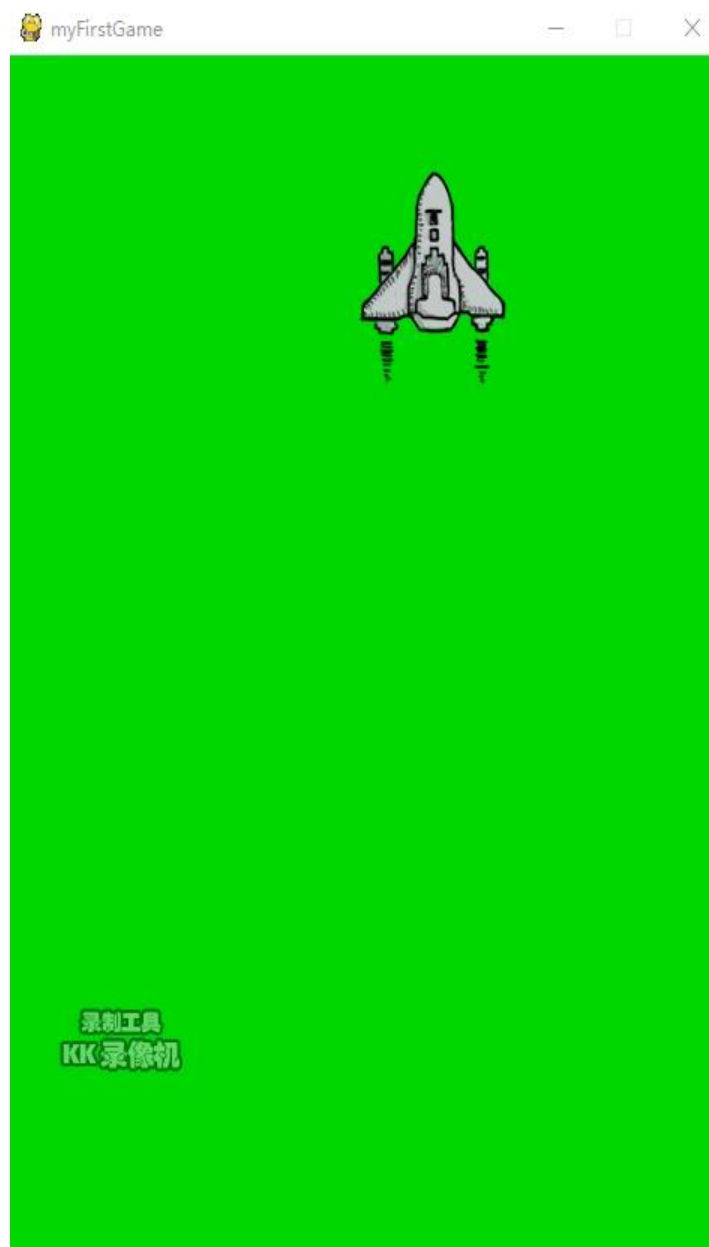
```
    # 垂直方向翻转
```

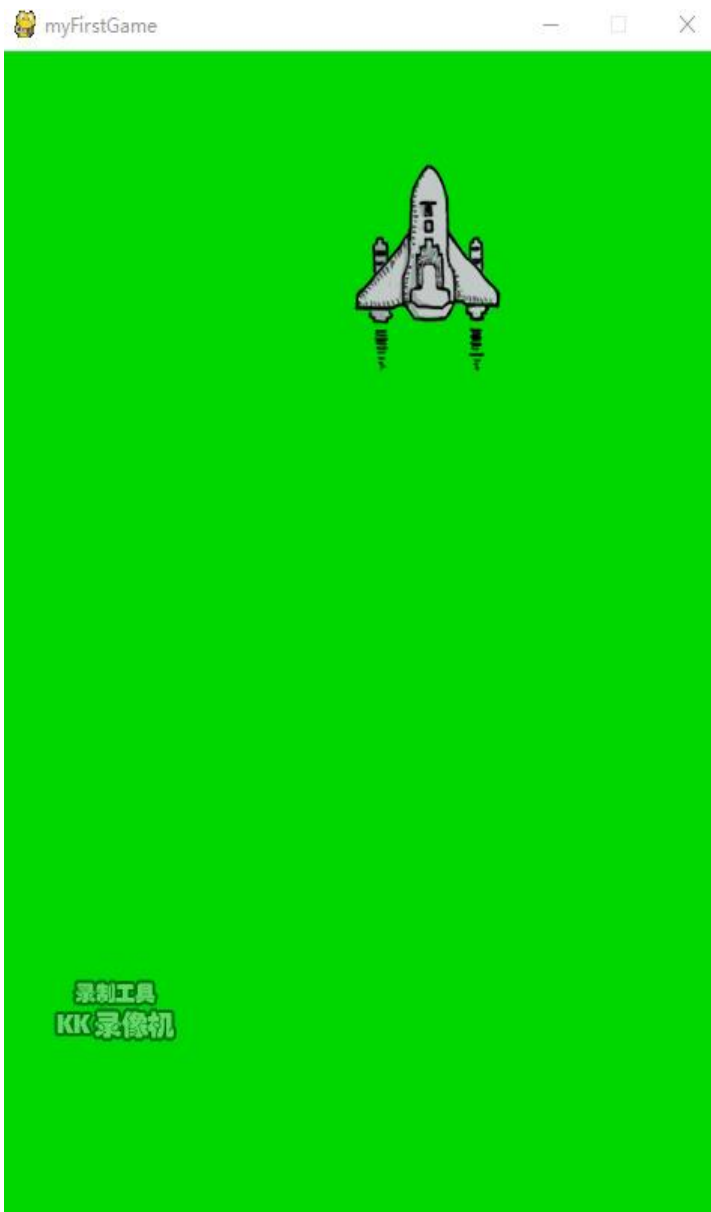
```
    heroPlane = pygame.transform.flip(heroPlane, True, False)
```

```
    speed[1] = -speed[1]    #垂直方向不翻转, 保持飞机头向上, 垂直方向  
    的移动速度反向
```



- 小飞机飞不出去啦~~~





是不是跑的太快了?  
要不要加点延迟.....





- **time**模块在pygame中用来管理时间。
- 可以实现延时（或者说让程序暂停一段时间），比如
- `pygame.time.wait()`和`pygame.time.delay()`;
- 限制程序的运行速度，比如`pygame.time.Clock()`对象中的`tick()`方法，`tick(40)`表示程序运行速度不会超过每秒40帧（次）；



- `pygame.time.wait()`
- 将暂停给定的毫秒数（括号中的参数应传入毫秒数）。该函数（方法）会将游戏进程休眠，可以与其他程序共享处理器，占用很少的处理器资源，但是准确度不够高。
- 下面，我们让小飞机每延时20ms移动一次。
- .....[省略代码]
- `screen.fill(bg)`  
`screen.blit(heroPlane, position)`  
`pygame.display.flip()`  
`pygame.time.wait(20)`    #延时20ms再执行下面测程序

还能移动的再慢一点吗？



- `pygame.time.delay()`
- 将暂停给定的毫秒数（括号中的参数应传入毫秒数）。该函数（方法）不会让游戏进程休眠，会占用处理器资源，但是时间比`wait()`更准确。
- 下面，我们让小飞机每延时20ms移动一次。
- .....[省略代码]
- `screen.fill(bg)`  
`screen.blit(heroPlane, position)`  
`pygame.display.flip()`  
`pygame.time.delay(20)`    #延时20ms再执行下面测程序

还能移动的再慢一点  
吗？



- `pygame.time.Clock()`
- 创建一个可用于跟踪一段时间的Clock对象，该Clock对象还提供了几个函数（方法）来控制游戏的帧率。
- 比如`tick()`方法可以设置游戏的帧率，也只会占用很少的处理器资源。
- 1帧表示的是一幅图像，帧率指的是1秒钟可以切换多少幅图像。pygame支持40~200帧/秒的帧率。
- .....[省略代码]
- `clock = pygame.time.Clock()` #创建一个Clock对象，并命名为clock
- .....[省略代码]
- `screen.fill(bg)`  
`screen.blit(heroPlane, position)`  
`pygame.display.flip()`  
`clock.tick(50)` #程序以每秒不超过50帧的速度运行。

还能移动的再慢一点吗？

