[【python】10分钟教你用Python做个打飞机小游戏超详细教程](https://www.cnblogs.com/dengfaheng/p/9260954.html)

我知道你们一定想先看效果如何



00 目录

* 整体框架
* 开始之前-精灵类Sprite
* 子弹类class Bullet
* 玩家飞机类class Player
* 敌机类class Enemy
* 游戏主体循环以及帧率设置
* 让子弹飞
* 刷出敌机 打怪
* 把飞机敌机子弹都画出来
* 处理键盘事件
* 分数显示 和 GameOver
* 最终代码

01 前言

这次还是用python的pygame库来做的游戏。关于这个库的内容，读者可以上网了解一下。本文只讲解用到的知识。代码参考自网上，自己也做了一点代码简化。尽量把最核心的方面用最简单的方式呈现给大家，让大家尽快掌握这个游戏的框架。至于那些华丽的功能，大家在弄懂了核心知识以后，再去添加也是非常easy的。

02 整体框架

这个游戏设计用到了面向对象的编程思想。  
游戏主体划分为三个主要的类：

* 子弹类class Bullet
* 玩家类class Player
* 敌机类class Enemy

在屏幕上可见的也就是这三个东西了。自己的飞机、敌人的飞机、子弹。因此整个游戏的核心就是：

* 把这三个东西的图像呈现在屏幕上。
* 判断和处理子弹撞击敌机和敌机撞击玩家这两种情况。

下面我们会展开为大家一一讲解。

03 开始之前-精灵类Sprite

在下面的代码中，你们会大量见到这个pygame.sprite模块。这里就给大家介绍一下。“sprite”，中文翻译“精灵”，在游戏动画一般是指一个独立运动的画面元素，在pygame中，就可以是一个带有图像（Surface）和大小位置（Rect）的对象。 简单来说是一个会动图片。它的两个成员变量

* self.image=要显示图片的Surface
* self.rect = 显示Surface的区域

对于self.rect，常用的设置rect的方法：self.rect = self.image.get\_rect()。然后设定self.rect.topleft=(0,0)来设定左上角的位置，从而设定这个精灵在屏幕上的显示位置。精灵特别适合用在OO语言中，比如Python。

pygame.sprite.Sprite是pygame精灵的基类，一般来说，你总是需要写一个自己的精灵类继承一下它然后加入自己的代码。

关于此类的其他函数，咱们用到的时候会详细跟大家说的。

04 子弹类class Bullet

先来看代码吧。

1# 子弹类  
 2class Bullet(pygame.sprite.Sprite):  
 3    def \_\_init\_\_(self, bullet\_img, init\_pos):  
 4        pygame.sprite.Sprite.\_\_init\_\_(self)  
 5        self.image = bullet\_img  
 6        self.rect = self.image.get\_rect()  
 7        self.rect.midbottom = init\_pos  
 8        self.speed = 10  
 9  
10    def move(self):  
11        self.rect.top -= self.speed

子弹类继承于pygame.sprite.Sprite， 成员主要是子弹的图片对象和子弹刷出来的位置，当然，还有移动速度。一个方法就是移动，从发出位置直线往屏幕上方移动。

05 玩家飞机类class Player

老样子。先看代码

1# 玩家飞机类  
 2class Player(pygame.sprite.Sprite):  
 3    def \_\_init\_\_(self, plane\_img, player\_rect, init\_pos):  
 4        pygame.sprite.Sprite.\_\_init\_\_(self)  
 5        self.image = []                                 # 用来存储玩家飞机图片的列表  
 6        for i in range(len(player\_rect)):  
 7            self.image.append(plane\_img.subsurface(player\_rect[i]).convert\_alpha())  
 8        self.rect = player\_rect[0]                      # 初始化图片所在的矩形  
 9        self.rect.topleft = init\_pos                    # 初始化矩形的左上角坐标  
10        self.speed = 8                                  # 初始化玩家飞机速度，这里是一个确定的值  
11        self.bullets = pygame.sprite.Group()            # 玩家飞机所发射的子弹的集合  
12        self.is\_hit = False                             # 玩家是否被击中  
13  
14    # 发射子弹  
15    def shoot(self, bullet\_img):  
16        bullet = Bullet(bullet\_img, self.rect.midtop)  
17        self.bullets.add(bullet)  
18  
19    # 向上移动，需要判断边界  
20    def moveUp(self):  
21        if self.rect.top <= 0:  
22            self.rect.top = 0  
23        else:  
24            self.rect.top -= self.speed  
25  
26    # 向下移动，需要判断边界  
27    def moveDown(self):  
28        if self.rect.top >= SCREEN\_HEIGHT - self.rect.height:  
29            self.rect.top = SCREEN\_HEIGHT - self.rect.height  
30        else:  
31            self.rect.top += self.speed  
32  
33    # 向左移动，需要判断边界  
34    def moveLeft(self):  
35        if self.rect.left <= 0:  
36            self.rect.left = 0  
37        else:  
38            self.rect.left -= self.speed  
39  
40    # 向右移动，需要判断边界          
41    def moveRight(self):  
42        if self.rect.left >= SCREEN\_WIDTH - self.rect.width:  
43            self.rect.left = SCREEN\_WIDTH - self.rect.width  
44        else:  
45            self.rect.left += self.speed

老样子，成员变量主要还是那几个。图像对象以及矩形参数和刷出位置，当然还会有移动速度和子弹集合（用来保存飞机射出的子弹）。方法的话就是上下左右移动了，不过需要做好边界判断。这个直接看代码就能理解了。

06 敌机类class Enemy

好吧，先上代码伺候。

1# 敌机类  
 2class Enemy(pygame.sprite.Sprite):  
 3    def \_\_init\_\_(self, enemy\_img, enemy\_down\_imgs, init\_pos):  
 4       pygame.sprite.Sprite.\_\_init\_\_(self)  
 5       self.image = enemy\_img   #正常的图像  
 6       self.rect = self.image.get\_rect()  
 7       self.rect.topleft = init\_pos  
 8       self.down\_imgs = enemy\_down\_imgs # 爆炸的图像  
 9       self.speed = 2  
10  
11    # 敌机移动，边界判断及删除在游戏主循环里处理  
12    def move(self):  
13        self.rect.top += self.speed

需要注意的时候，该类保存了两个图像对象，一个是正常情况下的敌机图像。一个是爆炸的敌机图像。以便在撞击时能把撞击效果显示出来。一个方法就是和子弹差不多的移动了，不过它是从屏幕上方往底下移动的而已。然后刷出位置的话，后面我们会用一个随机函数生成的。

07 游戏主体循环以及帧率设置

游戏主体的话，我们直接开一个死循环来不断刷新显示上面介绍的三个对象。代码设计如下：

1# 游戏循环帧率设置  
 2clock = pygame.time.Clock()  
 3  
 4# 判断游戏循环退出的参数  
 5running = True  
 6  
 7# 游戏主循环  
 8while running:  
 9    # 控制游戏最大帧率为 60  
10    clock.tick(60)  
11  
12    ……游戏运行部分

关于pygame.time.Clock()，贪吃蛇那篇已经介绍过了。就是用来控制游戏帧率的。只要我们的玩家飞机没有被敌机撞到，即属于存活状态时。running将一直为真。

08 让子弹飞

在running循环里面，我们要做的是不断自动刷出子弹。当然，子弹是从玩家飞机上射出来的。

* 首先是发射子弹

1# 生成子弹，需要控制发射频率  
2# 首先判断玩家飞机没有被击中  
3# 循环15次发射一个子弹  
4if not player.is\_hit:  
5    if shoot\_frequency % 15 == 0:  
6        player.shoot(bullet\_img)  
7    shoot\_frequency += 1  
8    if shoot\_frequency >= 15:  
9        shoot\_frequency = 0

shoot\_frequency变量的作用就是控制子弹发射的频率，它控制在running每循环15次发射一个子弹。

* 接着是子弹移动

1for bullet in player.bullets:  
2# 以固定速度移动子弹  
3bullet.move()  
4# 移动出屏幕后删除子弹  
5if bullet.rect.bottom < 0:  
6    player.bullets.remove(bullet)

子弹移动的话，running每循环一次，就move一下。不过要注意当子弹移动出屏幕后删除。不然可能会爆电脑内存。

09 刷出敌机 打怪

和子弹类似的，在running循环里，随机刷出敌机。

* 先是刷怪

1# 生成敌机，需要控制生成频率  
2# 循环50次生成一架敌机  
3if enemy\_frequency % 50 == 0:  
4    enemy1\_pos = [random.randint(0, SCREEN\_WIDTH - enemy1\_rect.width), 0]  
5    enemy1 = Enemy(enemy1\_img, enemy1\_down\_imgs, enemy1\_pos)  
6    enemies1.add(enemy1)  
7enemy\_frequency += 1  
8if enemy\_frequency >= 100:  
9    enemy\_frequency = 0

enemy\_frequency变量的作用同样是控制刷怪的频率。running每循环50次就刷一个怪出来，位置是randint函数随机生成的。

* 接着让怪移动

1 for enemy in enemies1:  
2    #2. 移动敌机  
3    enemy.move()  
4    #4. 移动出屏幕后删除敌人  
5    if enemy.rect.top < 0:  
6        enemies1.remove(enemy)

移动的话也很简单，每running循环一次就move一次就行。但是还是注意。敌机移出屏幕后要删除，避免爆内存啊。

* 然后是碰撞检测

1#3. 敌机与玩家飞机碰撞效果处理  
2if pygame.sprite.collide\_circle(enemy, player):  
3    enemies\_down.add(enemy)  
4    enemies1.remove(enemy)  
5    player.is\_hit = True  
6    break

这里介绍一下pygame.sprite.collide\_circle，这个函数的作用是判断两个精灵对象有没有碰撞。如果敌机和玩家飞机装上了，那很明显GameOver了。直接把running循环给break就行了。

10 把飞机敌机子弹都画出来

前面说了这么多，最终我们还是要把这三个主要的对象画到屏幕上显示出来，然后通过每一次running循环更新它们的状态（正常？撞击？爆炸？）。

1#敌机被子弹击中效果处理  
 2#将被击中的敌机对象添加到击毁敌机 Group 中  
 3enemies1\_down = pygame.sprite.groupcollide(enemies1, player.bullets, 1, 1)  
 4for enemy\_down in enemies1\_down:  
 5    enemies\_down.add(enemy\_down)  
 6  
 7# 绘制背景  
 8screen.fill(0)  
 9screen.blit(background, (0, 0))  
10  
11# 绘制玩家飞机  
12if not player.is\_hit:  
13    screen.blit(player.image[0], player.rect) #将正常飞机画出来  
14else:  
15    # 玩家飞机被击中后的效果处理  
16    screen.blit(player.image[1], player.rect) #将爆炸的飞机画出来  
17    running = False  
18  
19# 敌机被子弹击中效果显示  
20for enemy\_down in enemies\_down:  
21    enemies\_down.remove(enemy\_down)  
22    score += 1  
23    screen.blit(enemy\_down.down\_imgs, enemy\_down.rect) #将爆炸的敌机画出来  
24  
25  
26    # 显示子弹  
27    player.bullets.draw(screen)  
28    # 显示敌机  
29    enemies1.draw(screen)

注意的是，玩家飞机和敌机都有两种状态，一种是正常状态，另外一种是爆炸状态。在画之前要判断清楚再下手。然后再介绍一下pygame.sprite.groupcollide函数，这个函数是判断两个精灵组里面的精灵有没有相互碰撞的。它会把A组的精灵逐个和B组的精灵进行比较判断。

11 处理键盘事件

键盘事件的处理是十分重要的，我们通过键盘移动飞机，更新飞机的位置。最终再画出来。代码如下

1# 处理游戏退出  
 2for event in pygame.event.get():  
 3    if event.type == pygame.QUIT:  
 4        pygame.quit()  
 5        exit()  
 6  
 7# 获取键盘事件（上下左右按键）  
 8key\_pressed = pygame.key.get\_pressed()  
 9  
10# 处理键盘事件（移动飞机的位置）  
11if key\_pressed[K\_w] or key\_pressed[K\_UP]:  
12    player.moveUp()  
13if key\_pressed[K\_s] or key\_pressed[K\_DOWN]:  
14    player.moveDown()  
15if key\_pressed[K\_a] or key\_pressed[K\_LEFT]:  
16    player.moveLeft()  
17if key\_pressed[K\_d] or key\_pressed[K\_RIGHT]:  
18    player.moveRight()

12 分数显示 和 GameOver

对于分数显示，其实很简单，用一个font对象，在render渲染到屏幕上就可以了。

1# 绘制得分  
2score\_font = pygame.font.Font(None, 36)  
3score\_text = score\_font.render('score: '+str(score), True, (128, 128, 128))  
4text\_rect = score\_text.get\_rect()  
5text\_rect.topleft = [10, 10]  
6screen.blit(score\_text, text\_rect)

不过，需要注意的是，最后我们还要将总得分在游戏结束的时候写出来。然后游戏结束的时候，我们还要把GameOver那张图片也blit出来。

1# 游戏 Game Over 后显示最终得分  
2font = pygame.font.Font(None, 64)  
3text = font.render('Final Score: '+ str(score), True, (255, 0, 0))  
4text\_rect = text.get\_rect()  
5text\_rect.centerx = screen.get\_rect().centerx  
6text\_rect.centery = screen.get\_rect().centery + 24  
7screen.blit(game\_over, (0, 0))  
8screen.blit(text, text\_rect)

13 最终代码

讲了这么多，相信大家都明白了。最后再贴一个完整的代码和游戏所需的资源吧。不过，这是比较基本的一个打飞机。具体大家还可以根据自己想要的内容，给它画龙点睛。比如爆炸效果，加个背景音乐。设置联网对战等等等等。靠大家自己发挥啦。

欲获取代码和相关资源，请关注我们的微信公众号【程序猿声】，在后台回复：pyplane。即可获取。



1#-\*- coding: utf-8 -\*-  
 2import pygame  
 3from sys import exit  
 4from pygame.locals import \*  
 5import random  
 6  
 7# 设置游戏屏幕大小  
 8SCREEN\_WIDTH = 480  
 9SCREEN\_HEIGHT = 800  
 10  
 11# 子弹类  
 12class Bullet(pygame.sprite.Sprite):  
 13    def \_\_init\_\_(self, bullet\_img, init\_pos):  
 14        pygame.sprite.Sprite.\_\_init\_\_(self)  
 15        self.image = bullet\_img  
 16        self.rect = self.image.get\_rect()  
 17        self.rect.midbottom = init\_pos  
 18        self.speed = 10  
 19  
 20    def move(self):  
 21        self.rect.top -= self.speed  
 22  
 23# 玩家飞机类  
 24class Player(pygame.sprite.Sprite):  
 25    def \_\_init\_\_(self, plane\_img, player\_rect, init\_pos):  
 26        pygame.sprite.Sprite.\_\_init\_\_(self)  
 27        self.image = []                                 # 用来存储玩家飞机图片的列表  
 28        for i in range(len(player\_rect)):  
 29            self.image.append(plane\_img.subsurface(player\_rect[i]).convert\_alpha())  
 30        self.rect = player\_rect[0]                      # 初始化图片所在的矩形  
 31        self.rect.topleft = init\_pos                    # 初始化矩形的左上角坐标  
 32        self.speed = 8                                  # 初始化玩家飞机速度，这里是一个确定的值  
 33        self.bullets = pygame.sprite.Group()            # 玩家飞机所发射的子弹的集合  
 34        self.is\_hit = False                             # 玩家是否被击中  
 35  
 36    # 发射子弹  
 37    def shoot(self, bullet\_img):  
 38        bullet = Bullet(bullet\_img, self.rect.midtop)  
 39        self.bullets.add(bullet)  
 40  
 41    # 向上移动，需要判断边界  
 42    def moveUp(self):  
 43        if self.rect.top <= 0:  
 44            self.rect.top = 0  
 45        else:  
 46            self.rect.top -= self.speed  
 47  
 48    # 向下移动，需要判断边界  
 49    def moveDown(self):  
 50        if self.rect.top >= SCREEN\_HEIGHT - self.rect.height:  
 51            self.rect.top = SCREEN\_HEIGHT - self.rect.height  
 52        else:  
 53            self.rect.top += self.speed  
 54  
 55    # 向左移动，需要判断边界  
 56    def moveLeft(self):  
 57        if self.rect.left <= 0:  
 58            self.rect.left = 0  
 59        else:  
 60            self.rect.left -= self.speed  
 61  
 62    # 向右移动，需要判断边界          
 63    def moveRight(self):  
 64        if self.rect.left >= SCREEN\_WIDTH - self.rect.width:  
 65            self.rect.left = SCREEN\_WIDTH - self.rect.width  
 66        else:  
 67            self.rect.left += self.speed  
 68  
 69# 敌机类  
 70class Enemy(pygame.sprite.Sprite):  
 71    def \_\_init\_\_(self, enemy\_img, enemy\_down\_imgs, init\_pos):  
 72       pygame.sprite.Sprite.\_\_init\_\_(self)  
 73       self.image = enemy\_img  
 74       self.rect = self.image.get\_rect()  
 75       self.rect.topleft = init\_pos  
 76       self.down\_imgs = enemy\_down\_imgs  
 77       self.speed = 2  
 78  
 79    # 敌机移动，边界判断及删除在游戏主循环里处理  
 80    def move(self):  
 81        self.rect.top += self.speed  
 82  
 83# 初始化 pygame  
 84pygame.init()  
 85  
 86# 设置游戏界面大小、背景图片及标题  
 87# 游戏界面像素大小  
 88screen = pygame.display.set\_mode((SCREEN\_WIDTH, SCREEN\_HEIGHT))  
 89  
 90# 游戏界面标题  
 91pygame.display.set\_caption('Python打飞机大战')  
 92  
 93# 背景图  
 94background = pygame.image.load('resources/image/background.png').convert()  
 95  
 96# Game Over 的背景图  
 97game\_over = pygame.image.load('resources/image/gameover.png')  
 98  
 99# 飞机及子弹图片集合  
100plane\_img = pygame.image.load('resources/image/shoot.png')  
101  
102# 设置玩家飞机不同状态的图片列表，多张图片展示为动画效果  
103player\_rect = []  
104player\_rect.append(pygame.Rect(0, 99, 102, 126))        # 玩家飞机图片  
105player\_rect.append(pygame.Rect(165, 234, 102, 126))     # 玩家爆炸图片  
106  
107player\_pos = [200, 600]  
108player = Player(plane\_img, player\_rect, player\_pos)  
109  
110# 子弹图片  
111bullet\_rect = pygame.Rect(1004, 987, 9, 21)  
112bullet\_img = plane\_img.subsurface(bullet\_rect)  
113  
114# 敌机不同状态的图片列表，包括正常敌机，爆炸的敌机图片  
115enemy1\_rect = pygame.Rect(534, 612, 57, 43)  
116enemy1\_img = plane\_img.subsurface(enemy1\_rect)  
117enemy1\_down\_imgs = plane\_img.subsurface(pygame.Rect(267, 347, 57, 43))  
118  
119  
120#存储敌机，管理多个对象  
121enemies1 = pygame.sprite.Group()  
122  
123# 存储被击毁的飞机  
124enemies\_down = pygame.sprite.Group()  
125  
126# 初始化射击及敌机移动频率  
127shoot\_frequency = 0  
128enemy\_frequency = 0  
129  
130# 初始化分数  
131score = 0  
132  
133# 游戏循环帧率设置  
134clock = pygame.time.Clock()  
135  
136# 判断游戏循环退出的参数  
137running = True  
138  
139# 游戏主循环  
140while running:  
141    # 控制游戏最大帧率为 60  
142    clock.tick(60)  
143  
144    # 生成子弹，需要控制发射频率  
145    # 首先判断玩家飞机没有被击中  
146    # 循环15次发射一个子弹  
147    if not player.is\_hit:  
148        if shoot\_frequency % 15 == 0:  
149            player.shoot(bullet\_img)  
150        shoot\_frequency += 1  
151        if shoot\_frequency >= 15:  
152            shoot\_frequency = 0  
153  
154    # 生成敌机，需要控制生成频率  
155    # 循环50次生成一架敌机  
156    if enemy\_frequency % 50 == 0:  
157        enemy1\_pos = [random.randint(0, SCREEN\_WIDTH - enemy1\_rect.width), 0]  
158        enemy1 = Enemy(enemy1\_img, enemy1\_down\_imgs, enemy1\_pos)  
159        enemies1.add(enemy1)  
160    enemy\_frequency += 1  
161    if enemy\_frequency >= 100:  
162        enemy\_frequency = 0  
163  
164    for bullet in player.bullets:  
165        # 以固定速度移动子弹  
166        bullet.move()  
167        # 移动出屏幕后删除子弹  
168        if bullet.rect.bottom < 0:  
169            player.bullets.remove(bullet)     
170  
171    for enemy in enemies1:  
172        #2. 移动敌机  
173        enemy.move()  
174        #3. 敌机与玩家飞机碰撞效果处理  
175        if pygame.sprite.collide\_circle(enemy, player):  
176            enemies\_down.add(enemy)  
177            enemies1.remove(enemy)  
178            player.is\_hit = True  
179            break  
180        #4. 移动出屏幕后删除敌人  
181        if enemy.rect.top < 0:  
182            enemies1.remove(enemy)  
183  
184    #敌机被子弹击中效果处理  
185    #将被击中的敌机对象添加到击毁敌机 Group 中  
186    enemies1\_down = pygame.sprite.groupcollide(enemies1, player.bullets, 1, 1)  
187    for enemy\_down in enemies1\_down:  
188        enemies\_down.add(enemy\_down)  
189  
190    # 绘制背景  
191    screen.fill(0)  
192    screen.blit(background, (0, 0))  
193  
194    # 绘制玩家飞机  
195    if not player.is\_hit:  
196        screen.blit(player.image[0], player.rect) #将正常飞机画出来  
197    else:  
198        # 玩家飞机被击中后的效果处理  
199        screen.blit(player.image[1], player.rect) #将爆炸的飞机画出来  
200        running = False  
201  
202    # 敌机被子弹击中效果显示  
203    for enemy\_down in enemies\_down:  
204        enemies\_down.remove(enemy\_down)  
205        score += 1  
206        screen.blit(enemy\_down.down\_imgs, enemy\_down.rect) #将爆炸的敌机画出来  
207  
208  
209    # 显示子弹  
210    player.bullets.draw(screen)  
211    # 显示敌机  
212    enemies1.draw(screen)  
213  
214    # 绘制得分  
215    score\_font = pygame.font.Font(None, 36)  
216    score\_text = score\_font.render('score: '+str(score), True, (128, 128, 128))  
217    text\_rect = score\_text.get\_rect()  
218    text\_rect.topleft = [10, 10]  
219    screen.blit(score\_text, text\_rect)  
220  
221    # 更新屏幕  
222    pygame.display.update()  
223  
224    # 处理游戏退出  
225    for event in pygame.event.get():  
226        if event.type == pygame.QUIT:  
227            pygame.quit()  
228            exit()  
229  
230    # 获取键盘事件（上下左右按键）  
231    key\_pressed = pygame.key.get\_pressed()  
232  
233    # 处理键盘事件（移动飞机的位置）  
234    if key\_pressed[K\_w] or key\_pressed[K\_UP]:  
235        player.moveUp()  
236    if key\_pressed[K\_s] or key\_pressed[K\_DOWN]:  
237        player.moveDown()  
238    if key\_pressed[K\_a] or key\_pressed[K\_LEFT]:  
239        player.moveLeft()  
240    if key\_pressed[K\_d] or key\_pressed[K\_RIGHT]:  
241        player.moveRight()  
242  
243# 游戏 Game Over 后显示最终得分  
244font = pygame.font.Font(None, 64)  
245text = font.render('Final Score: '+ str(score), True, (255, 0, 0))  
246text\_rect = text.get\_rect()  
247text\_rect.centerx = screen.get\_rect().centerx  
248text\_rect.centery = screen.get\_rect().centery + 24  
249screen.blit(game\_over, (0, 0))  
250screen.blit(text, text\_rect)  
251  
252# 显示得分并处理游戏退出  
253while 1:  
254    for event in pygame.event.get():  
255        if event.type == pygame.QUIT:  
256            pygame.quit()  
257            exit()  
258    pygame.display.update()