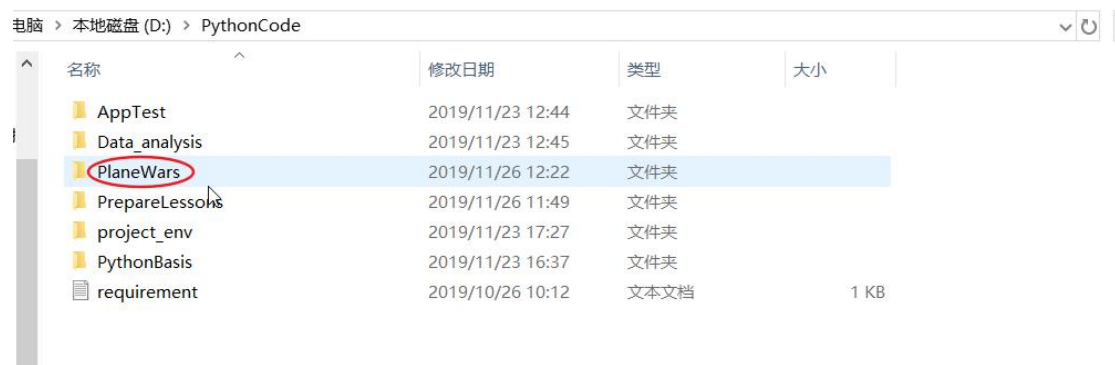
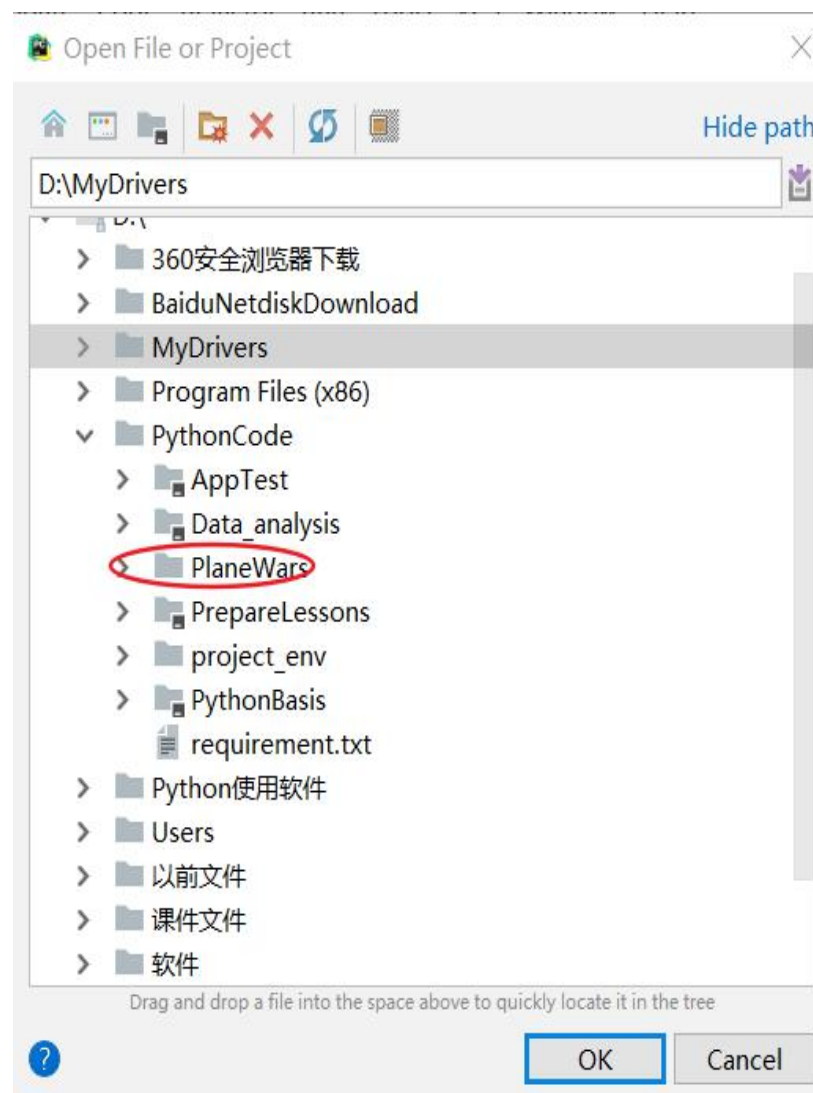
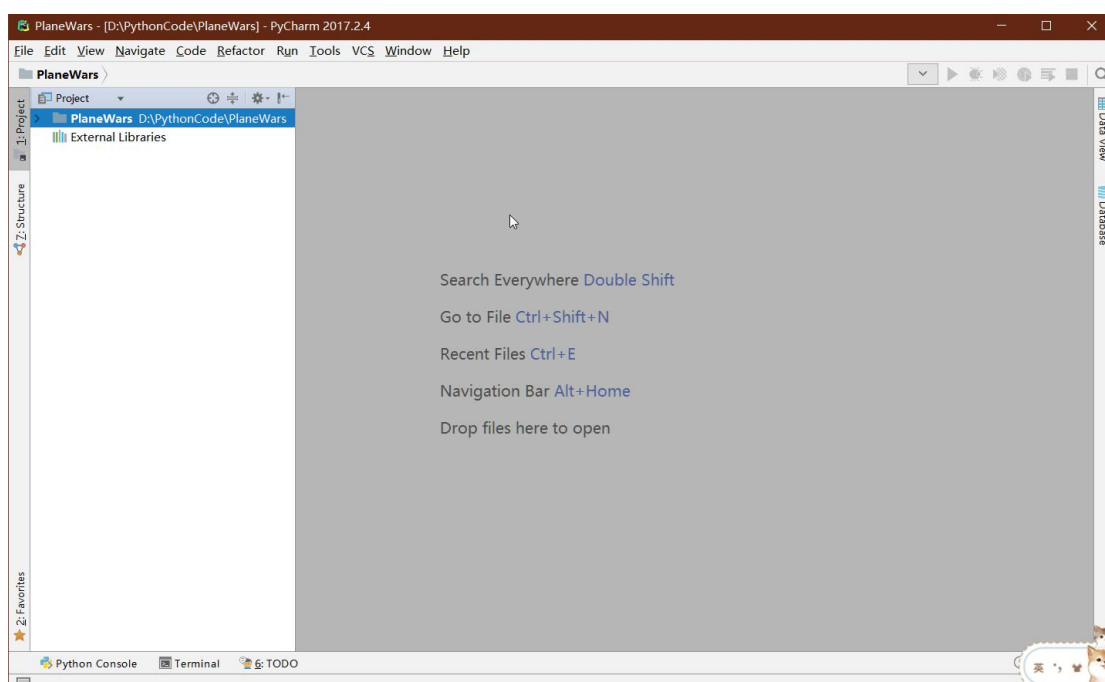


首先我们新建本地新建一个文件夹 **PlaneWars**,如下图



在 **pycharm** 中打开这个文件夹，以后飞机大战的项目就在这文件夹中进行。





## 任务 1.2 pygame 模块下载及入门

### 任务描述

本任务主要实现以下 2 个内容：

- (1) 安装 pygame
- (2) 尝试使用 pygame

### 知识学习

#### Pygame 历史

Pygame 是一个利用 SDL 库的写就的游戏库，SDL 呢，全名 Simple DirectMedia Layer，是一位叫做 Sam Lantinga 的大牛写的。SDL 是用 C 写的，不过它也可以使用 C++ 进行开发，当然还有很多其它的语言，Pygame 就是 Python 中使用它的一个库。Pygame 已经存在很多时间了，许多优秀的程序员加入其中，把 Pygame 做得越来越好。

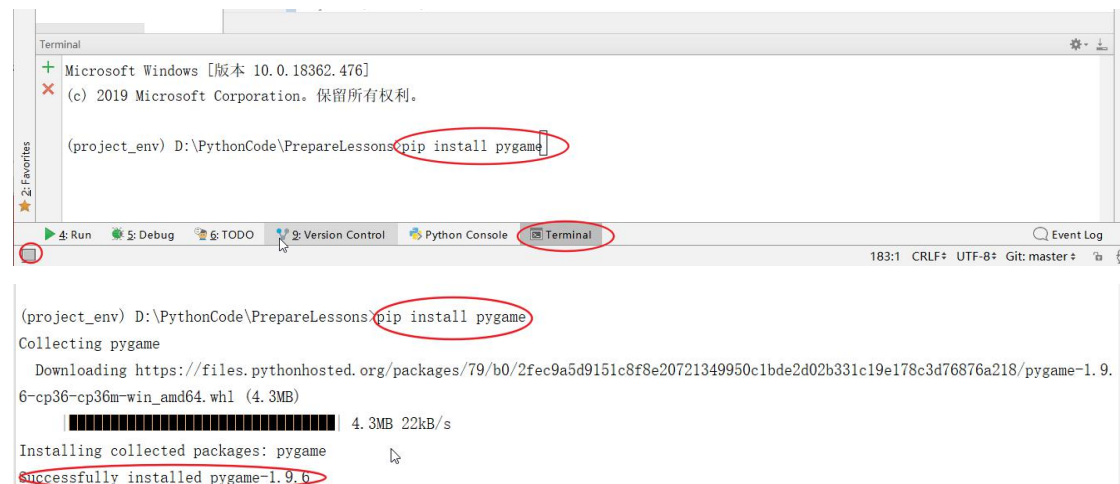
## 任务实施

### 1. 安装 pygame 模块

其安装命令如下：

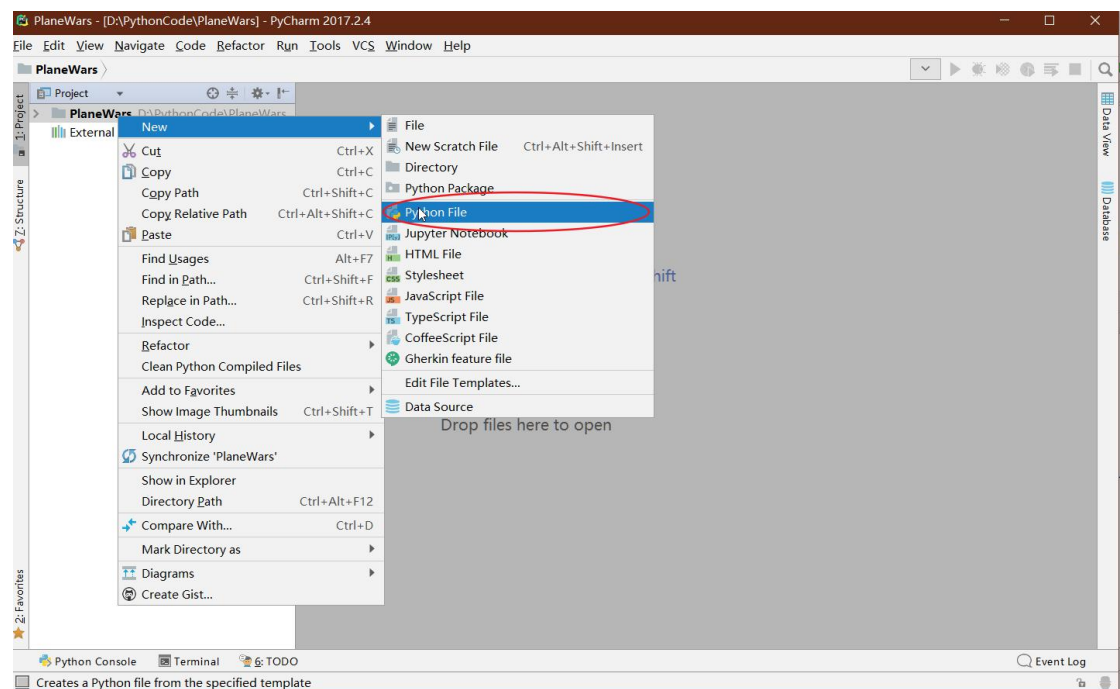
`pip install pygame`

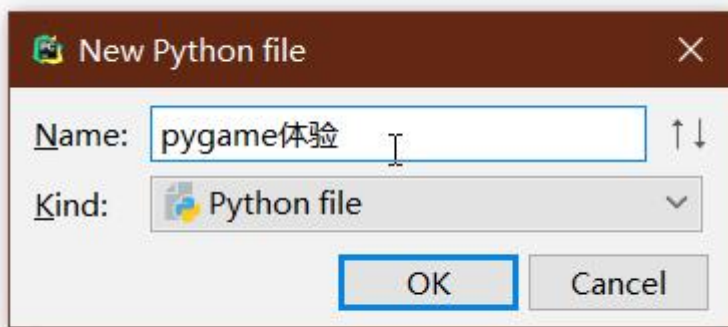
我们直接在 pycharm 中打开 命令行 下载，如下图



### 2. 尝试使用 pygame 模块

- 首先新建一个 py 文件 pygame 体验.py





- 然后编辑如下代码

```
import pygame
```

```
pygame.init() # 初始化
```

```
print("游戏代码....")
```

```
pygame.quit() # 卸载模块, 释放资源
```

运行结果:

```
D:\PythonCode\project_env\project_env\Scripts\python.exe D:/PythonCode/PlaneWars/pygame体验.py
pygame 1.9.6
Hello from the pygame community. https://www.pygame.org/contribute.html
游戏代码....|
```

## 任务 1.3 使用 rect 描述位置

### 任务描述

本任务主要是学会使用 rect 描述位置信息

### 知识学习

Rect 是用于存储矩形坐标的 pygame 对象, rect 对象有一些虚拟属性, 比如 top, left, bottom, right 这些是用来固定矩形的位置的, 还有 size, width, height, 这些是描述矩形大小, 宽高分别是多大。

center 为矩形的中心点, 其实就是关于横纵坐标的二元组, 因此又有 centerx, centery 两个属性。此外, 还有 x, y。

Rect 构造方法:

```
rect = pygame.Rect( left , top, width, height )
```

## 任务实施

下面我们再创建一个 plane\_rect.py 文件，这个是我们飞机大战游戏英雄机的位置描述。

创建好.py 文件之后，我们进行下面编程，代码如下：

```
import pygame

pygame.init()
hero_rect = pygame.Rect(100, 200, 125, 300)
# 100: 表示距离 x 轴原点的位置
# 200: 表示距离 y 轴原点的位置
# 125: 矩形的宽度
# 300: 矩形的长度
print("英雄机矩形的 x={},y={}".format(hero_rect.x, hero_rect.y))
print("英雄机矩形的宽 width = {},高 height = {}".format(hero_rect.width, hero_rect.height))
print("英雄机矩形的中心 centerx={}".format(hero_rect.centerx))
print("英雄机的低部 bottom={}".format(hero_rect.bottom))
print("英雄机的左部 left = {}".format(hero_rect.left))
print("英雄机的右部 right = {}".format(hero_rect.right))
print("英雄机大小 {}".format(hero_rect.size))
pygame.quit()
```

运行之后的效果如下图所示

```
pygame 1.9.6
Hello from the pygame community. https://www.pygame.org/contribute.html
英雄机矩形的x=100, y=200
英雄机矩形的宽width = 125, 高height = 300
英雄机矩形的中心 centerx=162
英雄机的低部 bottom=500
英雄机的左部 left = 100
英雄机的右部right = 225
英雄机大小
```

I

## 技术应用篇

### 单元二：pygame 应用及游戏开发

#### 学习目标

学生自己写一个飞机大战的游戏

#### 任务 2.1 创建游戏窗口

##### 任务描述

本任务主要是有下面 2 个任务：

- (1) 创建游戏窗口并实现可视化
- (2) 使游戏循环

##### 知识学习

```
set_mode(resolution=(0,0), flags=0, depth=0)
```

- resolution: 窗体的宽和高
- flags: 附件选项的集合

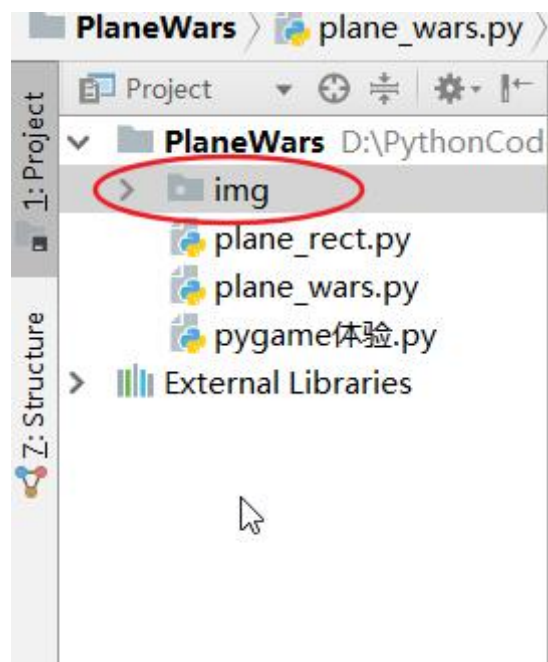
选项	含义
pygame.FULLSCREEN	全屏显示
pygame.DOUBLEBUF	双缓冲模式（推荐和 HWSURFACE 或 OPENGGL 一起使用）
pygame.HWSURFACE	硬件加速，只有在 FULLSCREEN 下可以使用
pygame.OPENGGL	创建一个 OPENGGL 渲染的显示
pygame.RESIZABLE	创建一个可调整尺寸的窗口
pygame.NOFRAME	创建一个没有边框和控制按钮的窗口

- **depth:** 使用的颜色深度

## 任务实施

我们在创建一个 plane\_wars.py 文件，这个文件是飞机大战项目文件，以后的代码都写在这么文件里。

文件创建好之后，需要做的是要将游戏的图片文件拷贝到项目文件下，图片文件在百度网盘里 <https://pan.baidu.com/s/1xkzl0c-Sun3RA6bqusmLHw>，或者直接到班级群里下载，下载好之后在放到项目的目录下



下面来实现创建游戏窗口和游戏循环

```
import pygame
```

```
pygame.init()
# 1.创建游戏窗口
pygame.display.set_mode((480,650)) # 元组中 480 表示宽度， 650 表示高度

# 游戏循环
while True:
    pass
pygame.quit()
```

执行结果如下图所示





## 任务 2.2 修改游戏名称和图标

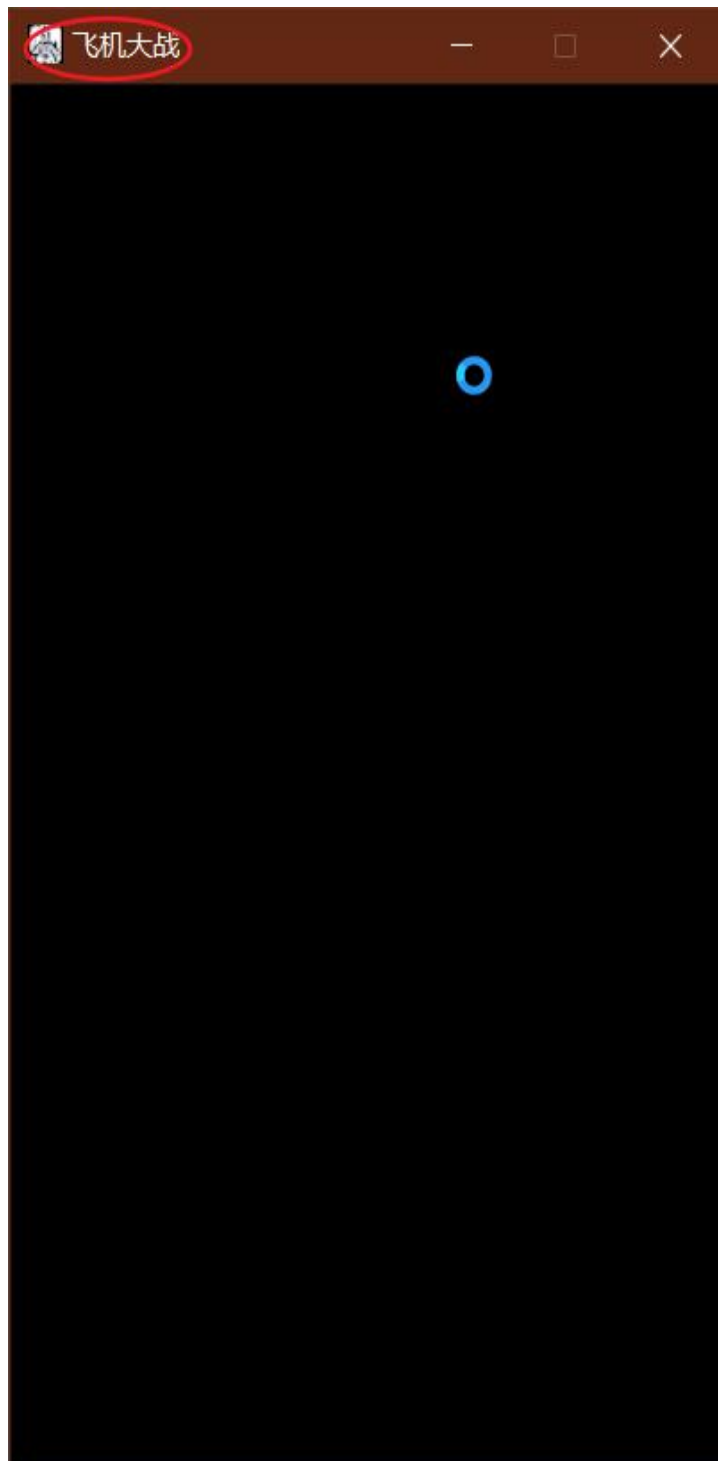
### 任务描述

本任务主要修改游戏的名称和图标，使其变成飞机大战游戏专有。

### 任务实施

```
# 修改游戏名称
pygame.display.set_caption("飞机大战")
# 修改游戏图标
icon = pygame.image.load("img/icon72x72.png")
pygame.display.set_icon(icon)
```

运行之后的效果图如下：



## 任务 2.3 绘制背景精灵和英雄飞机精灵

### 任务描述

本任务有 2 个：

- (1) 绘制游戏背景图片
- (2) 绘制英雄飞机图片

### 任务实施

#### 1. 绘制背景精灵

# 1.加载背景图片

```
bg_img = pygame.image.load("img/background.png")
```

# 2.将背景图片加载到窗口中，（0,0）表示背景图片放到原点位置

```
screen.blit(bg_img,(0,0))
```

#### 2. 绘制英雄飞机精灵

绘制英雄飞机精灵，我们要实现飞机的动态效果，必须是两张图片，我们 if...else 语句实现其动态效果

- 首先加载英雄精灵的图片

```
hero_img1 = pygame.image.load("img/hero1.png")
```

```
hero_img2 = pygame.image.load("img/hero2.png")
```

- 然后用.....if...else....绘制英雄精灵图片

```
heroIndexShif = 0 #作为切换图片索引
```

```
hero_rect = pygame.rect.Rect(190,526,66,80) # 定义英雄机 rect
```

# 将英雄的飞机绘制到窗口上

```
if heroIndexShift == 0:
```

```
    screen.blit(hero_img1,(hero_rect.x,hero_rect.y))
```

```
    heroIndexShift +=1
```

```
else:
```

```
    screen.blit(hero_img2,(hero_rect.x,hero_rect.y))
```

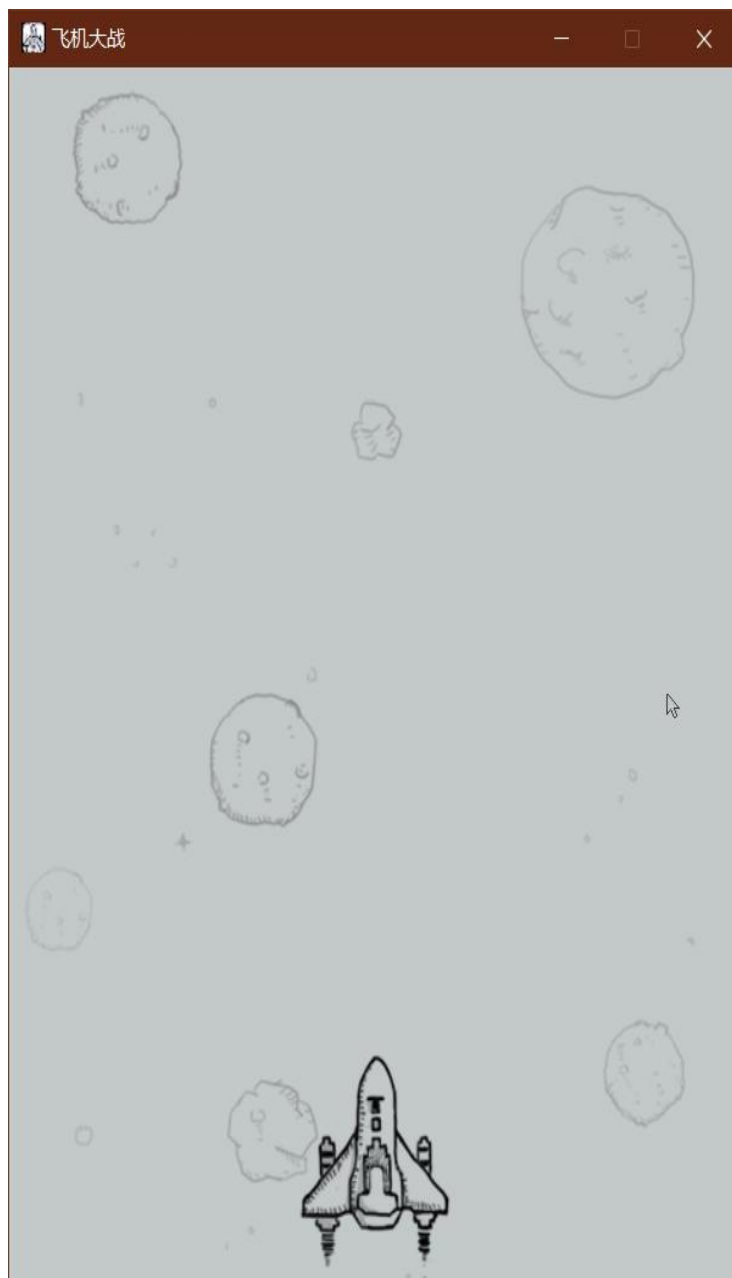
```
    heroIndexShift = 0
```

## 代码位置图:

```
import pygame

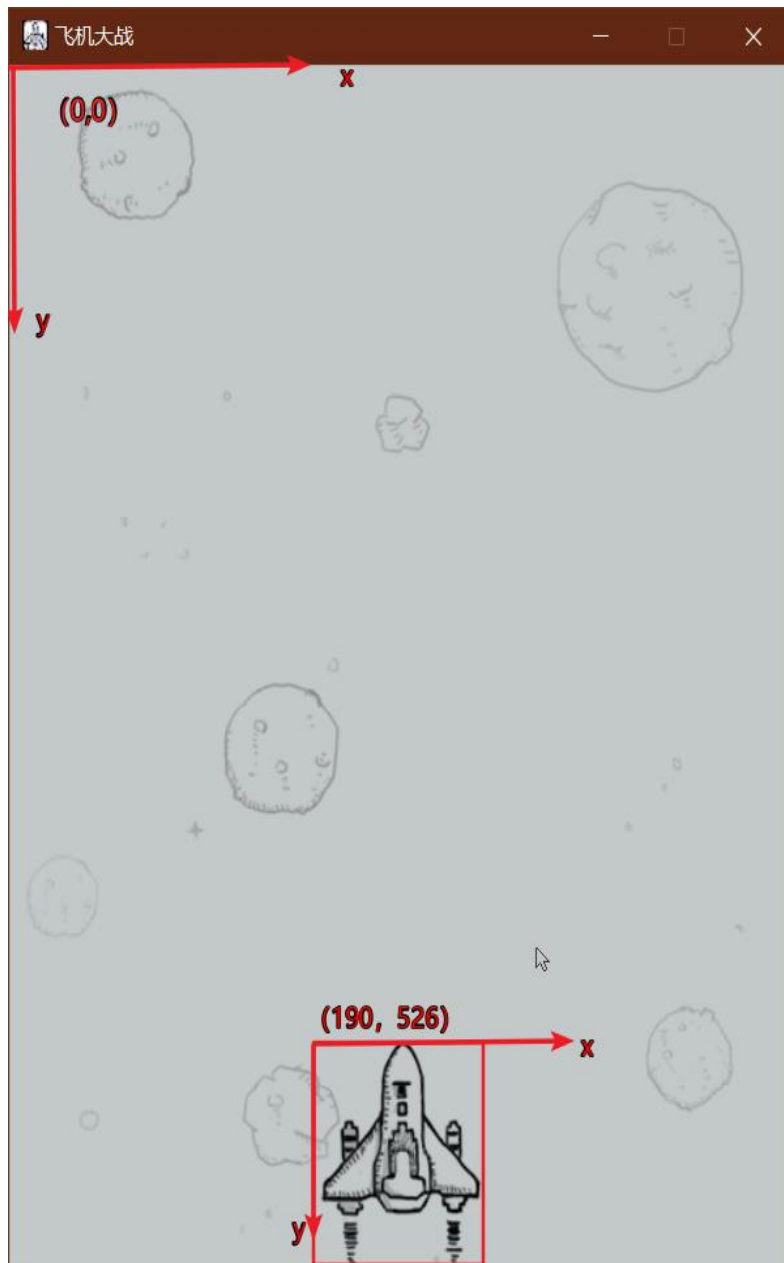
pygame.init()# 游戏初始化
# 使用变量screen 接收返回值,代表整个窗口对象
screen = pygame.display.set_mode((480,650)) # 元组中320表示宽度, 568表示高度
# 修改游戏名称
pygame.display.set_caption("飞机大战")
# 修改游戏图标
icon = pygame.image.load("img/icon72x72.png")
pygame.display.set_icon(icon)
# 加载背景图片
bg_img = pygame.image.load("img/background.png")
# 加载英雄飞机
hero_img1 = pygame.image.load("img/hero1.png")
hero_img2 = pygame.image.load("img/hero2.png")
heroIndexShift = 0
# 定义英雄飞机的rect
hero_rect = pygame.rect.Rect(190,526,66,80)
# 游戏循环
while True:
    # 将背景图片加载到窗口中, (0,0)表示背景图片放到原点位置
    screen.blit(bg_img,(0, 0))
    # 将英雄的飞机绘制到窗口上
    if heroIndexShift == 0:
        screen.blit(hero_img1,(hero_rect.x,hero_rect.y))
        heroIndexShift +=1
    else:
        screen.blit(hero_img2,(hero_rect.x,hero_rect.y))
        heroIndexShift = 0
```

## 运行代码后的效果图



其中 飞机位置的确定确认是这样的，如下图：

窗口的大小都是 480\*650，飞机大小是 100\*124，由于飞机初始化的时候在窗口下方的正下方，所以英雄机的坐标为（190,526），如图所示：

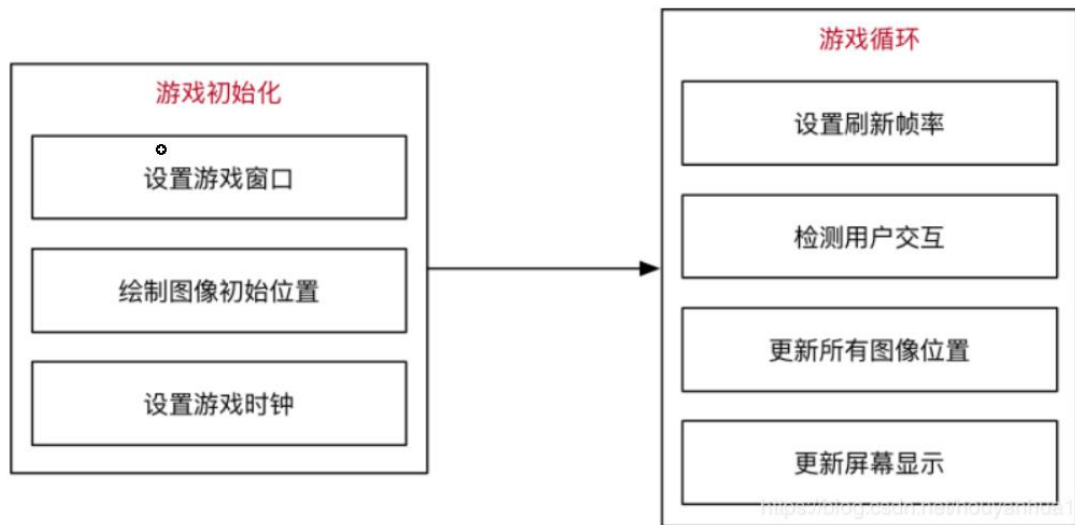


## 任务 2.4 游戏时钟设置

### 任务描述

本任务主要是为了设置游戏的刷新次数

## 任务知识



## 任务实施

# 创建游戏时钟

```
clock = pygame.time.Clock()
```

```
clock.tick(60) # 60 表示每秒钟刷新 60 次
```

代码位置图:

```

import pygame

pygame.init()# 游戏初始化
# 使用变量screen 接收返回值，代表整个窗口对象
screen = pygame.display.set_mode((480,650)) # 元组中320表示宽度， 568表示高度
# 修改游戏名称
pygame.display.set_caption("飞机大战")
# 修改游戏图标
icon = pygame.image.load("img/icon72x72.png")
pygame.display.set_icon(icon)
# 加载背景图片
bg_img = pygame.image.load("img/background.png")
# 加载英雄飞机
hero_img1 = pygame.image.load("img/hero1.png")
hero_img2 = pygame.image.load("img/hero2.png")
heroIndexShift = 0
# 定义英雄飞机的rect
hero_rect = pygame.rect.Rect(190,526,66,80)
# 创建游戏时钟
clock = pygame.time.Clock()
# 游戏循环
while True:
    clock.tick(60) # 60表示每秒钟刷新60次
    # 将背景图片加载到窗口中，（0,0）表示背景图片放到原点位置
    screen.blit(bg_img, (0, 0))
    # 将英雄的飞机绘制到窗口上
    if heroIndexShift == 0:
        screen.blit(hero_img1,(hero_rect.x,hero_rect.y))
        heroIndexShift +=1
    else:
        screen.blit(hero_img2,(hero_rect.x,hero_rect.y))
        heroIndexShift = 0

```

## 任务 2.5 更改英雄飞机精灵坐标位置

### 任务描述

本任务为了英雄精灵实现上下自动移动

### 任务实施

```

# 修改英雄飞机的 y 轴值
hero_rect.y -= 1
# 让飞机从底部飞进
if hero_rect.bottom <=0:

```



```
hero_rect.y = 650
```

代码位置图:

```
# 定义英雄飞机的rect
hero_rect = pygame.rect.Rect(190,526,66,80)
# 创建游戏时钟
clock = pygame.time.Clock()
# 游戏循环 ->意味着游戏正式开始
while True:
    clock.tick(60) # 60表示每秒钟刷新60次
    # 将背景图片加载到窗口中, (0,0)表示背景图片放到原点位置
    screen.blit(bg_img, (0, 0))
    # 将英雄的飞机绘制到窗口上
    if heroIndexShift == 0:
        screen.blit(hero_img1, (hero_rect.x, hero_rect.y))
        heroIndexShift += 1
    else:
        screen.blit(hero_img2, (hero_rect.x, hero_rect.y))
        heroIndexShift = 0
    # 修改英雄飞机的y轴值
    hero_rect.y -= 1
    # 让飞机从底部飞进
    if hero_rect.bottom <= 0:
        hero_rect.y = 650
```

I

## 任务 2.6 事件监听和控制英雄精灵移动

### 任务描述

本任务主要有 2 个:

- (1) 事件监听
- (2) 控制英雄精灵移动

事件监听主要是键盘事件和鼠标事件, 之前游戏不能正常关闭, 是因为没有事件监听。

### 任务实施

1. 首先获取所有事件

# 获取所有的事件

```
event_list = pygame.event.get()
```

## 2. 实现正常退出游戏

# 2. 捕获窗口退出事件

```
for event in event_list:
```

```
    if event.type == pygame.QUIT: # 加上这个模块就不卡了
```

```
        print("游戏结束了.....")
```

```
        pygame.quit() # 卸载模块
```

```
        exit(0) # 终止 Python 程序, exit(0)表示正常退出 exit(1)表示异常退出
```

## 3. 控制英雄精灵移动

# 控制英雄精灵移动

```
if event.type == pygame.KEYDOWN:
```

```
    if event.key == pygame.K_LEFT: # 向左移动
```

```
        heroPlaneX = heroPlaneX - 5 if heroPlaneX >= 5 else 0
```

```
    elif event.key == pygame.K_RIGHT: # 向右移动
```

```
        heroPlaneX = heroPlaneX + 5 if heroPlaneX <= 375 else 380
```

```
    elif event.key == pygame.K_DOWN: # 向下移动
```

```
        heroPlaneY = heroPlaneY + 5 if heroPlaneY <= 521 else 526
```

```
    elif event.key == pygame.K_UP: # 向上移动
```

```
        heroPlaneY = heroPlaneY - 5 if heroPlaneY >= 5 else 0
```

## 4. 按键动作重复执行

```
pygame.key.set_repeat(20,30)
```

- set\_repeat(20,30)中 20,30 都是毫秒级的, 20 代表按下键盘 20ms 后反应, 30 代表 当在 30ms 内还没有抬起键盘, 就会再给一个 20ms 按键反应

**代码位置图:**

```

# 英雄精灵的x,y轴坐标
heroPlaneX = hero_rect.x
heroPlaneY = hero_rect.y
pygame.key.set_repeat(20,30)

while True: # 游戏循环 ->意味着游戏正式开始
    clock.tick(60) # 60表示每秒钟刷新60次
    # 将背景图片加载到窗口中, (0,0) 表示背景图片放到原点位置
    screen.blit(bg_img, (0, 0))
    # 修改英雄飞机的y轴值
    hero_rect.y -= 1
    # 让飞机从底部飞进
    if hero_rect.bottom <= 0:
        hero_rect.y = 650
    # 将英雄的飞机绘制到窗口上
    if heroIndexShift == 0:
        screen.blit(hero_img1, (heroPlaneX, heroPlaneY))
        heroIndexShift += 1
    else:
        screen.blit(hero_img2, (heroPlaneX, heroPlaneY))
        heroIndexShift = 0

    # 1. 获取所有的事件
    event_list = pygame.event.get()
    # 2. 捕获窗口退出事件
    for event in event_list:
        if event.type == pygame.QUIT: # 加上这个模块就不卡了
            print("游戏结束了.....")
            pygame.quit() # 卸载模块
            exit(0) # 终止Python程序, exit(0)表示正常退出 exit(1)表示异常退出
        # 控制英雄精灵移动
        if event.type == pygame.KEYDOWN:
            if event.key == pygame.K_LEFT: # 向左移动
                heroPlaneX = heroPlaneX - 5 if heroPlaneX >= 5 else 0
            elif event.key == pygame.K_RIGHT: # 向右移动
                heroPlaneX = heroPlaneX + 5 if heroPlaneX <= 375 else 380
            elif event.key == pygame.K_DOWN: # 向下移动
                heroPlaneY = heroPlaneY + 5 if heroPlaneY <= 521 else 526
            elif event.key == pygame.K_UP: # 向上移动
                heroPlaneY = heroPlaneY - 5 if heroPlaneY >= 5 else 0

    pygame.display.update()

```

记得修改, 因为不修改在循环里一直会被初始化, 无法控制精灵移动

## 任务 2.7 绘制敌机精灵和控制敌机精灵移动

### 任务描述

#### (1) 绘制敌机

## (2) 控制敌机精灵移动

### 任务实施

#### 1. 绘制敌机精灵

# 加载敌机精灵

```
enemy_img = pygame.image.load("img/enemy1.png")
```

# 定义敌机精灵的 rect

```
enemy_rect = pygame.rect.Rect(206,0,69,89) # x= 480//2-69//2
```

# 英雄精灵的 x,y 轴坐标

```
enemyPlaneX = enemy_rect.x
```

```
enemyPlaneY = enemy_rect.y
```

# 绘制敌机精灵

```
screen.blit(enemy_img,(enemyPlaneX,enemyPlaneY))
```

#### 2. 控制敌机精灵移动

# 控制敌机精灵移动

```
if direct == "左":
```

```
    enemyPlaneX -= 5
```

```
    if enemyPlaneX <= 0:
```

```
        direct = "右"
```

```
elif direct == "右":
```

```
    enemyPlaneX += 5
```

```
    if enemyPlaneX >= 480-69:
```

```
        direct = "左"
```

**代码位置图：**

```

import pygame

pygame.init()# 游戏初始化
# 使用变量screen 接收返回值，代表整个窗口对象
screen = pygame.display.set_mode((480,650)) # 元组中320表示宽度， 568表示高度
# 修改游戏名称
pygame.display.set_caption("飞机大战")
# 修改游戏图标
icon = pygame.image.load("img/icon72x72.png")
pygame.display.set_icon(icon)
# 加载背景图片
bg_img = pygame.image.load("img/background.png")
# 加载英雄飞机
hero_img1 = pygame.image.load("img/hero1.png")
hero_img2 = pygame.image.load("img/hero2.png")
# 加载敌机精灵
enemy_img = pygame.image.load("img/enemy1.png")
heroIndexShift = 0
# 定义英雄飞机的rect
hero_rect = pygame.rect.Rect(190,526,100,124)
# 定义敌机精灵的rect
enemy_rect = pygame.rect.Rect(206,0,69,89) # x= 480//2-69//2
# 敌机精灵的x,y轴坐标
enemyPlaneX = enemy_rect.x
enemyPlaneY = enemy_rect.y
direct = '左' # 定义敌机初始移动方向
# 创建游戏时钟
clock = pygame.time.Clock()
# 英雄精灵的x,y轴坐标
heroPlaneX = hero_rect.x
heroPlaneY = hero_rect.y
pygame.key.set_repeat(20,30)
while True:# 游戏循环 ->意味着游戏正式开始

```

```

# 获取所有的事件
event_list = pygame.event.get()
# 捕获窗口退出事件
for event in event_list:
    if event.type == pygame.QUIT:# 加上这个模块就不卡了
        print("游戏结束了.....")
        pygame.quit() # 卸载模块
        exit(0) # 终止Python程序, exit(0)表示正常退出    exit(1)表示异常退出
# 控制英雄精灵移动
if event.type == pygame.KEYDOWN:
    if event.key == pygame.K_LEFT:# 向左移动
        heroPlaneX = heroPlaneX - 5 if heroPlaneX >= 5 else 0
    elif event.key == pygame.K_RIGHT:# 向右移动
        heroPlaneX = heroPlaneX + 5 if heroPlaneX <= 375 else 380
    elif event.key == pygame.K_DOWN: # 向下移动
        heroPlaneY = heroPlaneY + 5 if heroPlaneY <= 521 else 526
    elif event.key == pygame.K_UP: # 向上移动
        heroPlaneY = heroPlaneY - 5 if heroPlaneY >= 5 else 0

# 绘制敌机精灵
screen.blit(enemy_img, (enemyPlaneX, enemyPlaneY))
# 控制敌机精灵移动
if direct == "左":
    enemyPlaneX -= 5
    if enemyPlaneX <= 0:
        direct = "右"
elif direct == "右":
    enemyPlaneX += 5
    if enemyPlaneX >= 480-69:
        direct = "左"

pygame.display.update()

```

## 任务 2.8 绘制精灵子弹

### 任务描述

本任务主要绘制两种子弹:

- (1) 英雄机子弹
- (2) 敌机子弹

### 任务实施

1. 绘制精灵子弹的时候我们用类方法

```

class HeroBullet():
    """
    英雄精灵子弹的类
    """

    def __init__(self, x, y, screen):
        """
        :param x: x 坐标
        :param y: y 坐标
        :param screen: 窗口对象
        """
        self.x = x
        self.y = y
        self.screen = screen
        self.pic = pygame.image.load("img/bullet.png")

    def draw(self):
        """用来画子弹"""
        self.screen.blit(self.pic, (self.x, self.y))
        self.move()

    def move(self):
        self.y -= 5

class EnemyBullet():
    """敌机精灵子弹类"""

    def __init__(self, x, y, screen):
        self.x = x
        self.y = y
        self.screen = screen
        self.pic = pygame.image.load("img/bullet1.png")

    def draw(self):
        """用来画子弹"""
        self.screen.blit(self.pic, (self.x, self.y))
        self.move()

```

```
def move(self):  
    self.y -= 5
```

## 2. 画出英雄战机的子弹

```
    elif event.key == pygame.K_SPACE: # 英雄机控制发射子弹  
        # 实例化子弹对象  
        hero_bullet = HeroBullet(heroPlaneX + 50 - 11, heroPlaneY - 22, screen) # 把子弹  
        添加到列表里  
        HeroBiulist.append(hero_bullet)  
# 画出英雄战机的子弹每一个子弹  
for bullet in HeroBiulist:  
    bullet.draw()  
    # 让子弹到最上边的时候消失  
    HeroBiulist.remove(bullet) if bullet.y < 0 else ""
```

**代码位置图：**



```

# 控制英雄精灵移动
if event.type == pygame.KEYDOWN:
    if event.key == pygame.K_LEFT: # 向左移动
        heroPlaneX = heroPlaneX - 5 if heroPlaneX >= 5 else 0
    elif event.key == pygame.K_RIGHT: # 向右移动
        heroPlaneX = heroPlaneX + 5 if heroPlaneX <= 375 else 380
    elif event.key == pygame.K_DOWN: # 向下移动
        heroPlaneY = heroPlaneY + 5 if heroPlaneY <= 521 else 526
    elif event.key == pygame.K_UP: # 向上移动
        heroPlaneY = heroPlaneY - 5 if heroPlaneY >= 5 else 0
    elif event.key == pygame.K_SPACE: # 英雄机控制发射子弹
        hero_bullet = HeroBullet(heroPlaneX + 50 - 11, heroPlaneY - 22,
screen)
        HeroBiulist.append(hero_bullet)

# 绘制敌机精灵
screen.blit(enemy_img, (enemyPlaneX, enemyPlaneY))
# 控制敌机精灵移动
if direct == "左":
    enemyPlaneX -= 5
    if enemyPlaneX <= 0:
        direct = "右"
elif direct == "右":
    enemyPlaneX += 5
    if enemyPlaneX >= 480 - 69:
        direct = "左"

# 画出英雄战机的子弹每一个子弹
for bullet in HeroBiulist:
    bullet.draw() # 绘制子弹
    # 让子弹到最上边的时候消失
    HeroBiulist.remove(bullet) if bullet.y < 0 else ""

```

### 3. 画出敌机子弹

```

# 画出敌机子弹
# 产生随机数
x = random.randint(0,100)
if x == 5 or x == 78:
    # 实例化一个子弹
    enemy_bullet = EnemyBullet(enemyPlaneX+69//2-9//2,enemyPlaneY+89,screen)
    # 产生的每一个子弹放到一个列表里
    EnemyBiulist.append(enemy_bullet)
for bullet in EnemyBiulist:
    bullet.draw() # 绘制子弹
    # 让子弹到最下面的时候消失
    EnemyBiulist.remove(bullet) if bullet.y >650-89 -21//2 else ""

```

```
pygame.display.update()
```

## 代码位置图:

```
# 绘制敌机精灵
screen.blit(enemy_img, (enemyPlaneX, enemyPlaneY))
# 控制敌机精灵移动
if direct == "左":
    enemyPlaneX -= 5
    if enemyPlaneX <= 0:
        direct = "右"
elif direct == "右":
    enemyPlaneX += 5
    if enemyPlaneX >= 480 - 69:
        direct = "左"

# 画出英雄战机的子弹每一个子弹
for bullet in HeroBiulist:
    bullet.draw() # 绘制子弹
    # 让子弹到最上边的时候消失
    HeroBiulist.remove(bullet) if bullet.y < 0 else ""

# 画出敌机子弹
# 产生随机数
x = random.randint(0,100)
if x == 5 or x == 78:
    # 实例化一个子弹
    enemy_bullet = EnemyBullet(enemyPlaneX+69//2-9//2, enemyPlaneY+89, screen)
    # 产生的每一个子弹放到一个列表里
    EnemyBiulist.append(enemy_bullet)
for bullet in EnemyBiulist:
    bullet.draw() # 绘制子弹
    # 让子弹到最下面的时候消失
    EnemyBiulist.remove(bullet) if bullet.y > 650-89 -21//2 else ""
pygame.display.update()
```

## 任务 2.9 绘制精灵爆炸

### 任务描述

本任务主要有两个:

- (1) 绘制敌机爆炸

## (2) 绘制战机爆炸

### 任务知识

这一小节我们主要研究飞机与子弹碰撞控制条件，即碰撞检测

我们子弹和飞机都维护了一个矩形，`rect(x, y, width, height)`，其中 `x` 和 `y` 是矩形在窗口上的坐标，碰撞检测我们只需要循环遍历每一颗已经发射的子弹，子弹和每一个敌人飞机的矩形是否相交，如果相交，说明子弹击中了敌机，我们对应做行为就可以了。

这里我们用到 `colliderect()` 函数 用法如下：

```
rect1.colliderect(rect2)
```

### 任务实施

#### 1. 绘制敌机爆炸图片

```
is_bomb = False # 爆炸条件
enemy_bomb_index = 0 # 爆炸图片索引

# 控制敌机爆炸的条件
for bullet in HeroBiulist:
    # 定义子弹的 rect
    hero_bullet_rect = pygame.rect.Rect(bullet.x, bullet.y, 10, 10)
    flag = pygame.Rect.colliderect(enemy_rect, hero_bullet_rect) # 检测敌机和子弹的矩形
    是否相交
    if flag:
        print("敌机爆炸了.....")
        enemy_bomb = True # 爆炸条件为真
        HeroBiulist.remove(bullet)

enemy_bomb_list = ["img/enemy1_down1.png", "img/enemy1_down2.png",
"img/enemy1_down3.png", "img/enemy1_down4.png"]
# 绘制敌机爆炸的图片
if enemy_bomb == False:
    # 如果没有检测到爆炸，就绘制没有爆炸敌机的图片
```

```

screen.blit(enemy_img, (enemyPlaneX, enemyPlaneY))
else:# 绘制敌机爆炸
    if enemy_bomb_index == len(enemy_bomb_list):# 当敌机爆炸图片的下标和图片总数
        相同时, 说明爆炸图片已经绘制结束
        time.sleep(0.2)
        exit(0)# 结束程序
    enemy_bomb_img = pygame.image.load(enemy_bomb_list[enemy_bomb_index])#
    加载敌机爆炸图片
    screen.blit(enemy_bomb_img, (enemyPlaneX, enemyPlaneY))# 绘制敌机爆炸图片
    enemy_bomb_index +=1
    time.sleep(0.2)

```

### 代码位置图:

```

pygame.key.set_repeat(20, 30) # 重复按键操作
# 存放英雄机子弹的列表
HeroBiulist = []
# 存放敌机子弹列表
EnemyBiulist = []
enemy_is_bomb = False # 敌机爆炸条件
enemy_bomb_index = 0 # 敌机爆炸图片索引
while True: # 游戏循环 ->意味着游戏正式开始
    clock.tick(60) # 60表示每秒钟刷新60次
    # 将背景图片加载到窗口中, (0,0)表示背景图片放到原点位置

```

```

# 控制英雄精灵移动
if event.type == pygame.KEYDOWN:
    if event.key == pygame.K_LEFT: # 向左移动
        heroPlaneX = heroPlaneX - 5 if heroPlaneX >= 5 else 0
    elif event.key == pygame.K_RIGHT: # 向右移动
        heroPlaneX = heroPlaneX + 5 if heroPlaneX <= 375 else 380
    elif event.key == pygame.K_DOWN: # 向下移动
        heroPlaneY = heroPlaneY + 5 if heroPlaneY <= 521 else 526
    elif event.key == pygame.K_UP: # 向上移动
        heroPlaneY = heroPlaneY - 5 if heroPlaneY >= 5 else 0

# 绘制敌机精灵
screen.blit(enemy_img, (enemyPlaneX, enemyPlaneY))
# 控制敌机精灵移动
if direct == "左":
    enemyPlaneX -= 5
    if enemyPlaneX <= 0:
        direct = "右"
elif direct == "右":
    enemyPlaneX += 5
    if enemyPlaneX >= 480-69:
        direct = "左"

```

删除上面绘制敌机精灵的代码

```

elif direct == "右":
    enemyPlaneX += 5
    if enemyPlaneX >= 480 - 69:
        direct = "左"

# 画出英雄战机的子弹每一个子弹
for bullet in HeroBiulist:
    bullet.draw() # 绘制子弹
    # 让子弹到最上边的时候消失
    HeroBiulist.remove(bullet) if bullet.y < 0 else ""
    hero_bullet_rect = pygame.rect.Rect(bullet.x, bullet.y, 10, 10) # 定义英雄子弹的
rect
    flag = hero_bullet_rect.colliderect(enemy_rect) # 检测敌机和子弹的矩形是否相交
    if flag:
        print("敌机爆炸了.....")
        is_bomb = True # 爆炸条件为真
        HeroBiulist.remove(bullet)

# 绘制敌机爆炸的图片
if is_bomb == False:
    # 如果没有检测到爆炸,就绘制没有爆炸敌机的图片
    screen.blit(enemy_img, (enemyPlaneX, enemyPlaneY))
    # screen.blit(enemy_img, (enemy_rect.x, enemy_rect.y))
else: # 绘制敌机爆炸
    if enemy_bomb_index == len(enemy_bomb_list): # 当敌机爆炸图片的下标和图片总数相同
        # 说明爆炸图片已经绘制结束
        time.sleep(0.2)
        exit(0) # 结束程序
    enemy_bomb_img = pygame.image.load(enemy_bomb_list[enemy_bomb_index]) # 加载敌
    机爆炸图片
    screen.blit(enemy_bomb_img, (enemyPlaneX, enemyPlaneY)) # 绘制敌机爆炸图片
    # screen.blit(enemy_bomb_img, (enemy_rect.x, enemy_rect.y)) # 绘制敌机爆炸图片
    enemy_bomb_index += 1
    time.sleep(0.2)

```

## 2.绘制英雄机爆炸图片

hero\_is\_bomb = False # 英雄机爆炸条件

hero\_bomb\_index = 0 # 英雄机爆炸图片索引

for bullet in EnemyBiulist:

enemy\_bullet\_rect = pygame.rect.Rect(bullet.x, bullet.y, 9, 21) # 定义敌机子弹 rect

flag = enemy\_bullet\_rect.colliderect(hero\_rect) # 英雄机爆炸的条件

if flag:

print("英雄爆炸了.....")

hero\_is\_bomb = True # 爆炸条件为真

EnemyBiulist.remove(bullet)# 当战机爆炸的时候, 移除子弹



```

if hero_is_bomb == False:
    # 将英雄的飞机绘制到窗口上
    if heroIndexShift == 0:
        screen.blit(hero_img1, (heroPlaneX, heroPlaneY))
        heroIndexShift += 1
    else:
        screen.blit(hero_img2, (heroPlaneX, heroPlaneY))
        heroIndexShift = 0
else:
    if hero_bomb_index == len(hero_bomb_list):# 当爆炸图片加载结束后
        time.sleep(0.3)
        exit()
    # 加载英雄机爆炸图片
    hero_bomb_img = pygame.image.load(hero_bomb_list[hero_bomb_index])
    # 绘制英雄机爆炸的图片
    screen.blit(hero_bomb_img,(heroPlaneX,heroPlaneY))
    hero_bomb_index += 1
    time.sleep(0.2)

```

### 代码位置图:

```

# 存放英雄机子弹的列表
HeroBiulist = []
# 存放敌机子弹列表
EnemyBiulist = []
enemy_is_bomb = False # 敌机爆炸条件
enemy_bomb_index = 0 # 敌机爆炸图片索引
hero_is_bomb = False # 英雄机爆炸条件
hero_bomb_index = 0 # 英雄机爆炸图片索引
while True: # 游戏循环 ->意味着游戏正式开始
    clock.tick(60) # 60表示每秒钟刷新60次
    # 将背景图片加载到窗口中, (0,0)表示背景图片放到原点位置
    screen.blit(bg_img, (0, 0))

```

```

enemy_bomb_index = 0 # 爆炸图片索引
while True: # 游戏循环 ->意味着游戏正式开始
    clock.tick(60) # 60表示每秒钟刷新60次
    # 将背景图片加载到窗口中, (0,0)表示背景图片放到原点位置
    screen.blit(bg_img, (0, 0))
    # 修改英雄飞机的y轴值
    hero_rect.y -= 1
    # 让英雄精灵飞机从底部飞进
    if hero_rect.bottom <= 0:
        hero_rect.y = 650

    # 将英雄的飞机绘制到窗口上
    if heroIndexShift == 0:
        screen.blit(hero_img1, (heroPlaneX, heroPlaneY))
        heroIndexShift += 1
    else:
        screen.blit(hero_img2, (heroPlaneX, heroPlaneY))
        heroIndexShift = 0

    # 获取所有的事件
    event_list = pygame.event.get()
    # 捕获窗口退出事件
    for event in event_list:
        if event.type == pygame.QUIT: # 加上这个模块就不卡了
            print("游戏结束了.....")
            pygame.quit() # 卸载模块

```

删除这段代码  
因为和下面功能重复



```

    enemy_bullet = EnemyBullet(enemyPlaneX + 69 // 2 - 9 // 2, enemyPlaneY + 89,
screen)
    # 产生的每一个子弹放到一个列表里
    EnemyBiulist.append(enemy_bullet)
    for bullet in EnemyBiulist:
        bullet.draw() # 绘制子弹
        EnemyBiulist.remove(bullet) if bullet.y > 650 - 89 - 21 // 2 else "" # 让子弹
到最下面的时候消失

    enemy_bullet_rect = pygame.rect.Rect(bullet.x, bullet.y, 9, 21) # 定义敌机子弹
rect
    flag = enemy_bullet_rect.colliderect(hero_rect) # 英雄机爆炸的条件
    if flag:
        print("英雄爆炸了.....")
        hero_is_bomb = True # 爆炸条件为真
        EnemyBiulist.remove(bullet)# 当战机爆炸的时候，移除子弹

    if hero_is_bomb == False:
        # 将英雄的飞机绘制到窗口上
        if heroIndexShift == 0:
            screen.blit(hero_img1, (heroPlaneX, heroPlaneY))
            heroIndexShift += 1
        else:
            screen.blit(hero_img2, (heroPlaneX, heroPlaneY))
            heroIndexShift = 0
    else:
        if hero_bomb_index == len(hero_bomb_list):# 当爆炸图片加载结束后
            time.sleep(0.3)
            exit()
        # 加载英雄机爆炸图片
        hero_bomb_img = pygame.image.load(hero_bomb_list[hero_bomb_index])
        # 绘制英雄机爆炸的图片
        screen.blit(hero_bomb_img, (heroPlaneX, heroPlaneY))
        hero_bomb_index += 1
        time.sleep(0.2)

```

到这里，飞机大战游戏基本上完成了。