import pygame

#游戏代码分两个部分，循环上面的叫游戏的初始化，循环下面的叫游戏的循环，也就是游戏开始

#游戏初始化分几个部分，1、设置游戏窗口--->绘制图像初始位置--->3、设置游戏时钟

'''

hero\_rect = pygame.Rect(100,500,120,125)

#Rect是一个类，用于描述矩形区域，括号里面的参数是有顺序的，（第一个参数代表横坐标x(100)，第二个参数代表竖坐标y(500)，第三个参数代表图形的宽width(120)，第四个参数代表图形的高height(125)）

print("英雄的原点%d %d" % (hero\_rect.x, hero\_rect.y)) #代表着英雄(图片)的位置，我理解的是：英雄的坐标位置（100.500）

print("英雄的尺寸%d %d" % (hero\_rect.width, hero\_rect.height)) #代表英雄(图片)的大小,width代表宽度，height代表高度

print("%d %d" %hero\_rect.size) #size语法可以直接代表英雄的尺寸，也就是宽width和高height

pygame.init() #初始化

#创建游戏的主窗口,使用set\_mode函数，如500 \* 277，游戏窗口最好和游戏背景图像统一分辨率

#set\_mode有三个参数可以填(resolution(0.0), flags=0, depth=0)第一个参数指的是分辨率，也就是窗口的大小。第二个参数指的是屏幕附加选项，例如是否全屏等等

screen = pygame.display.set\_mode((500, 277))#这里示范第一个参数resolution()，也就是分辨率宽500\*高277

#1、使用pygame.image.load()加载图像的数据，括号里面的参数写图片位置

bg = pygame.image.load('D:/IT/人生苦短，我学python/pygame/游戏素材/背景.jpg')

#2、使用游戏屏幕对象，调用blit方法将图像绘制到指定位置，blit需要传递两个参数（图像，位置）

screen.blit(bg, (0, 0)) #bg是加载图像，（0，0）是位置，一般都是写0，0，代表在左上角位置开始，当你位置写别的数值，图片将和制作的屏幕分辨率有差异

#3、调用pygame.display.update()方法更新整个屏幕显示，如果不使用这句语法，那上面调用的背景图片将无法显示出来

pygame.display.update()

#加载英雄图像，英雄的图片必须要用透明的，透明图片只有png格式的，透明图片是在显示的时候，除了主要的图像，其他不会显示任何内容

hero = pygame.image.load('D:/IT/人生苦短，我学python/pygame/游戏素材/主角.png')

#初始英雄位置

#要想将英雄初始化在底部，下面的y值高度，就要将屏幕分辨率高度减去英雄图像的分辨率高度，得出的结果就是在底部位置

screen.blit(hero, (250, 215)) #像这样，屏幕分辨率高度277-英雄分辨率高度62=215，这句话可以理解成，在（变量名screen）屏幕里面插入（变量名hero）英雄图片，然后设置位置

pygame.display.update()

'''

#游戏中通常会添加一个循环，这样就可以保持窗口不会自动退出

#游戏中的循环叫游戏循环

while True:

pass

'''

#创建时钟对象

clock = pygame.time.Clock()

#1、定义rect记录飞机的初始位置

#Rect是一个类，用于描述矩形区域，括号里面的参数是有顺序的，（第一个参数代表横坐标x(255)，第二个参数代表竖坐标y(177)，第三个参数代表图形的宽width(50)，第四个参数代表图形的高height(62)）

#英雄初始之后的位置，可以理解成游戏开始移动的位置，（不过一般都可以和初始位置相同）

#要想将英雄初始化在底部，下面的y值高度，就要将屏幕分辨率高度减去英雄图像的分辨率高度，得出的结果就是在底部位置

hero\_rect = pygame.Rect(250, 215, 50, 62) #像这样，屏幕分辨率高度277-英雄分辨率高度62=215

#游戏循环，当代码执行到游戏循环，就意味着游戏的正式开始！

#游戏循环开始几个部分，1、设置刷新帧率--->2、检测用户交互--->3、更新所有图像位置--->4、更新屏幕显示

#游戏设置刷新帧率一般为60，帧率越少，播放就越慢

while True:

#可以指定循环体内部的代码执行的帧率

clock.tick(60)

#这段代码是固定的，只要是用了pygame都会使用这段代码，代表关闭按钮

#事件监听，当用户点击了关闭按钮，将退出整个程序

#event.get()方法，捕获事件，监听用户的操作，无论用户弄了什么，只要是在屏幕里面的，哪怕移动鼠标

for event in pygame.event.get():

#判断用户是否点击了关闭按钮

if event.type == pygame.QUIT: #QUIT代表右上角的关闭按钮，当用户点击了关闭按钮，则执行下面语句

print('退出游戏...')

pygame.quit()

#直接退出系统,终止整个程序

exit()

#event.get()方法，捕获事件，监听用户的操作，无论用户弄了什么，只要是在屏幕里面的，哪怕移动鼠标

event\_list = pygame.event.get()#捕获事件，监听用户的操作

if len(event\_list) > 0:

print(event\_list)

#2、修改飞机的位置

hero\_rect.y -= 1 #每次向y轴方向移动1，也就是-1

#判断飞机的位置

'''

#当飞机的Y值等于0的时候，则执行下面这语句

if hero\_rect.y <= 0: #当飞机的Y值等于0的时候

hero\_rect.y = 277 #当飞机的Y值等于0的时候，将重新从屏幕底部出发

'''

#如果想写美观一点，实现飞机完全走出屏幕再从底部重来，而不像上面这样，飞机的头部执行到屏幕上方Y值为0 的时候重来

if hero\_rect.y <= -62: #这个-62的参数，是把飞机图像的分辨率高度改为负数就好了

hero\_rect.y = 277

#3、调用blit方法绘制图像

screen.blit(bg, (0,0)) #加载背景图片

screen.blit(hero, hero\_rect)

#4、调用update方法更新图像显示

pygame.display.update()

pygame.quit()

print('------pygame------')