# -\*- coding:utf-8 -\*-

import pygame

import sys

from pygame.locals import \*

from pygame.font import \*

import time

import random

class Hero(object):

#玩家 英雄类

def \_\_init\_\_(self, screen\_temp):

self.x = 210

self.y = 700

self.life = 21

# self.life = 100

self.image = pygame.image.load("./feiji/hero1.png")

self.screen = screen\_temp

self.bullet\_list = []#用来存储子弹对象的引用

#爆炸效果用的如下属性

self.hit = False #表示是否要爆炸

self.bomb\_list = [] #用来存储爆炸时需要的图片

self.\_\_create\_images() #调用这个方法向bomb\_list中添加图片

self.image\_num = 0 #用来记录while True的次数,当次数达到一定值时才显示一张爆炸的图,然后清空,,当这个次数再次达到时,再显示下一个爆炸效果的图片

self.image\_index = 0#用来记录当前要显示的爆炸效果的图片的序号

def \_\_create\_images(self):

#添加爆炸图片

self.bomb\_list.append(pygame.image.load("./feiji/hero\_blowup\_n1.png"))

self.bomb\_list.append(pygame.image.load("./feiji/hero\_blowup\_n2.png"))

self.bomb\_list.append(pygame.image.load("./feiji/hero\_blowup\_n3.png"))

self.bomb\_list.append(pygame.image.load("./feiji/hero\_blowup\_n4.png"))

def display(self):

#显示玩家的飞机

#如果被击中,就显示爆炸效果,否则显示普通的飞机效果

if self.hit == True:

self.screen.blit(self.bomb\_list[self.image\_index], (self.x, self.y))#(self.x, self.y)是指当前英雄的位置

#blit方法 （一个对象，左上角位置）

self.image\_num += 1

print(self.image\_num)

if self.image\_num == 7:

self.image\_num = 0

self.image\_index += 1

print(self.image\_index) #这里子弹打住英雄时没有被清除掉，所以打一下，就死了

if self.image\_index > 3:

time.sleep(1)

exit()#调用exit让游戏退出

#self.image\_index = 0

else:

if self.x< 0: #控制英雄，不让它跑出界面

self.x = 0

elif self.x > 382:

self.x = 382

if self.y < 0:

self.y = 0

elif self.y > 750:

self.y = 750

self.screen.blit(self.image,(self.x, self.y))#z这里是只要没有被打中，就一直是刚开始的样子

#不管玩家飞机是否被击中,都要显示发射出去的子弹

for bullet in self.bullet\_list:

bullet.display()

bullet.move()

def move(self, move\_x,move\_y):

self.x += move\_x

self.y += move\_y

def fire(self):

#通过创建一个子弹对象,完成发射子弹

bullet = Bullet(self.screen, self.x, self.y)#创建一个子弹对象

self.bullet\_list.append(bullet)

def bomb(self):

self.hit = True

def judge(self):

global life

if life <= 0:

self.bomb()

class Bullet(object):

#玩家子弹类

def \_\_init\_\_(self, screen\_temp, x\_temp, y\_temp):

self.x = x\_temp + 40

self.y = y\_temp - 20

self.image = pygame.image.load("./feiji/bullet.png")

self.screen = screen\_temp

def display(self):

self.screen.blit(self.image, (self.x, self.y))

def move(self):

self.y -= 10

class Bullet\_Enemy(object):

#敌机子弹类

def \_\_init\_\_(self, screen\_temp, x\_temp, y\_temp):

self.x = x\_temp + 25

self.y = y\_temp + 30

self.image = pygame.image.load("./feiji/bullet1.png")

self.screen = screen\_temp

def display(self):

self.screen.blit(self.image,(self.x,self.y))

def move(self, hero):

self.y += 10

global life

if (hero.y <= self.y and self.y <= hero.y + 40) and (hero.x <= self.x and self.x <= hero.x + 100):

#if self.y in range(hero.y, hero.y + 40) and self.x in range(hero.x, hero.x + 40):

life -= 10

#self.bullet\_list.remove()

print("---judge\_enemy---")

return True

if life<=0:

hero.bomb()

return False

class Bullet\_Boss(object):

#boss子弹类1

def \_\_init\_\_(self, screen\_temp, x\_temp, y\_temp):

self.x = x\_temp + 80

self.y = y\_temp + 230

self.image = pygame.image.load("./feiji/bullet2.png")

self.screen = screen\_temp

def display(self):

self.screen.blit(self.image, (self.x, self.y))

def move(self, hero):

self.y += 6

self.x += 2

global life

if (hero.y <= self.y and self.y <= hero.y + 40) and (hero.x <= self.x and self.x <= hero.x + 100):

#if self.y in range(hero.y, hero.y + 40) and self.x in range(hero.x, hero.x + 40):

life -= 20

#self.bullet\_list.remove()

print("---judge\_boss---")

return True

if life<=0:

hero.bomb()

return False

class Bullet\_Boss1(object):

#boss子弹类2

def \_\_init\_\_(self, screen\_temp, x\_temp, y\_temp):

self.x = x\_temp + 80

self.y = y\_temp + 230

self.image = pygame.image.load("./feiji/bullet2.png")

self.screen = screen\_temp

def display(self):

self.screen.blit(self.image, (self.x, self.y))

def move(self, hero):

self.y += 6

self.x -= 2

global life

if (hero.y <= self.y and self.y <= hero.y + 40) and (hero.x <= self.x and self.x <= hero.x + 100):

#if self.y in range(hero.y, hero.y + 40) and self.x in range(hero.x, hero.x + 40):

life -= 20

#self.bullet\_list.remove()

print("---judge\_boss---")

return True

if life<=0:

hero.bomb()

return False

class Bullet\_Boss2(object):

#boss子弹类3

def \_\_init\_\_(self, screen\_temp, x\_temp, y\_temp):

self.x = x\_temp + 80

self.y = y\_temp + 230

self.image = pygame.image.load("./feiji/bullet2.png")

self.screen = screen\_temp

def display(self):

self.screen.blit(self.image, (self.x, self.y))

def move(self, hero):

self.y += 6

global life

if (hero.y <= self.y and self.y <= hero.y + 40) and (hero.x <= self.x and self.x <= hero.x + 100):

#if self.y in range(hero.y, hero.y + 40) and self.x in range(hero.x, hero.x + 40):

life -= 20

#self.bullet\_list.remove()

print("---judge\_boss---")

return True

if life<=0:

hero.bomb()

return False

class Base(object):

#基类 类似于抽象类

def \_\_init\_\_(self, screen\_temp, x, y, image\_name):

self.x = x

self.y = y

self.screen = screen\_temp

self.image = pygame.image.load(image\_name)

self.alive = True

def display(self):

if self.alive == True:

self.screen.blit(self.image, (self.x, self.y))

def move(self):

self.y += 5

class bomb\_bullet(Base):

#炸弹类

def \_\_init\_\_(self, screen\_temp):

Base.\_\_init\_\_(self, screen\_temp, random.randint(45, 400), 0, "./feiji/bomb.png")

def judge(self, hero):

if (hero.y <= self.y and self.y <= hero.y + 40) and (hero.x <= self.x and self.x <= hero.x + 100):

self.alive = False

hero.bomb()

if self.y >= 850:

#self.alive = False

self.y = 0

self.x = random.randint(45, 400)

#print("bomb.y = %d"%self.y)

class supply(Base):

#补给类

def \_\_init\_\_(self, screen\_temp):

Base.\_\_init\_\_(self, screen\_temp, random.randint(45, 400), -300, "./feiji/bomb-1.gif")

def judge(self, hero):

global life

if (hero.y <= self.y and self.y <= hero.y + 40) and (hero.x <= self.x and self.x <= hero.x + 100):

self.alive = False

life += 10

if self.y >= 1500:

self.y = 0

self.x = random.randint(45, 400)

self.alive = True

class clear\_bullet(Base):

def \_\_init\_\_(self, screen\_temp):

Base.\_\_init\_\_(self, screen\_temp, random.randint(45, 400), 0, "./feiji/bomb-2.gif")

self.alive = False

def judge(self, hero, enemies):

global q

q += 1

#self.move()

if q == 20:

#self.move()

self.alive = True

q = 0

if (hero.y <= self.y and self.y <= hero.y + 40) and (hero.x <= self.x and self.x <= hero.x + 100):

self.alive = False

for enemy in enemies:

enemy.hit == True

class EnemyPlane(object):

#敌机类

def \_\_init\_\_(self, screen\_temp):

self.x = random.randint(15, 480)

self.y = 0

self.image = pygame.image.load("./feiji/enemy0.png")

self.screen = screen\_temp

self.bullet\_list = []#用来存储子弹对象的引用

#self.direction = "right"#用来设置这个飞机默认的移动方向

self.hit = False

self.bomb\_list = []

self.\_\_create\_images()

self.image\_num = 0

self.image\_index = 0

#利用产生的随机数，随机确定飞机初始移动方向

self.k = random.randint(1, 20)

if self.k <= 10:

self.direction = "right"

elif self.k > 10:

self.direction = "left"

def display(self, hero):

#显示敌人的飞机

if not self.hit:

self.screen.blit(self.image, (self.x,self.y))

else:

self.screen.blit(self.bomb\_list[self.image\_index], (self.x,self.y))

self.image\_num += 1

if self.image\_num == 3 and self.image\_index < 3:

self.image\_num = 0

self.image\_index += 1

#print(self.image\_index)

# if self.image\_index > 2:

# time.sleep(0.1)

for bullet in self.bullet\_list:

bullet.display()

if(bullet.move(hero)):

self.bullet\_list.remove(bullet)

def move(self):

#利用随机数来控制飞机移动距离，以及移动范围

d1 = random.uniform(1,3)

d2 = random.uniform(0.2,3)

p1 = random.uniform(50,100)

p2 = random.uniform(-200,0)

if self.direction == "right":

self.x += d1

elif self.direction == "left":

self.x -= d1

if self.x > 480 - p1:

#480 - 50

self.direction="left"

elif self.x < p2:

self.direction = "right"

self.y += d2

def bomb(self):

self.hit = True

def \_\_create\_images(self):

self.bomb\_list.append(pygame.image.load("./feiji/enemy0\_down1.png"))

self.bomb\_list.append(pygame.image.load("./feiji/enemy0\_down2.png"))

self.bomb\_list.append(pygame.image.load("./feiji/enemy0\_down3.png"))

self.bomb\_list.append(pygame.image.load("./feiji/enemy0\_down4.png"))

def fire(self):

#利用随机数来控制敌机的开火，1/80的概率

s = random.randint(0,800)

bullet1 = Bullet\_Enemy(self.screen, self.x, self.y)

if s < 10:

self.bullet\_list.append(bullet1)

class EnemyPlanes(EnemyPlane):

#敌机群类 继承自EnemyPlane类

def \_\_init\_\_(self, screen\_temp):

EnemyPlane.\_\_init\_\_(self, screen\_temp)

self.num = 0

self.enemy\_list = [] #用列表存储产生的多架敌机

self.screen = screen\_temp

def add\_enemy(self, num):

#产生多架敌机的函数

self.num = num

for i in range(num):

enemy = EnemyPlane(self.screen)

self.enemy\_list.append(enemy)

def display(self, hero):

for i in range(self.num):

self.enemy\_list[i].display(hero)

def move(self):

for i in range(self.num):

self.enemy\_list[i].move()

def fire(self):

#s = random.randint(0,1000)

for i in range(self.num):

self.enemy\_list[i].fire()

class Boss(EnemyPlane):

#boss敌机类 继承自EnemyPlane类

def \_\_init\_\_(self,screen\_temp):

EnemyPlane.\_\_init\_\_(self,screen\_temp)

self.x = 150

self.y = 0

self.bomb\_list = []

self.\_\_create\_images()

self.image = pygame.image.load("./feiji/enemy2.png")

self.screen = screen\_temp

self.bullet\_list = []

def \_\_create\_images(self):

#self.bomb\_list.append(pygame.image.load("./feiji/enemy2.png"))

self.bomb\_list.append(pygame.image.load("./feiji/enemy2.png"))

self.bomb\_list.append(pygame.image.load("./feiji/enemy2\_down1.png"))

self.bomb\_list.append(pygame.image.load("./feiji/enemy2\_down2.png"))

self.bomb\_list.append(pygame.image.load("./feiji/enemy2\_down3.png"))

self.bomb\_list.append(pygame.image.load("./feiji/enemy2\_down4.png"))

self.bomb\_list.append(pygame.image.load("./feiji/enemy2\_down5.png"))

self.bomb\_list.append(pygame.image.load("./feiji/enemy2\_down6.png"))

def display(self, hero):

#显示敌人的飞机

global g

#print(g)

self.screen.blit(self.bomb\_list[g], (self.x,self.y))

for bullet in self.bullet\_list:

bullet.display()

if(bullet.move(hero)):

self.bullet\_list.remove(bullet)

def move(self):

d1 = 0

self.y += 0

def fire(self):

global s

s += 1

bullet1 = Bullet\_Boss(self.screen, self.x, self.y)

bullet2 = Bullet\_Boss1(self.screen, self.x, self.y)

bullet3 = Bullet\_Boss2(self.screen, self.x, self.y)

if s == 20:

s = 0

self.bullet\_list.append(bullet1)

self.bullet\_list.append(bullet2)

self.bullet\_list.append(bullet3)

def judge1(hero,enemy):

#判断敌机的炸毁

for bullet1 in hero.bullet\_list:

if bullet1.y in range(int(enemy.y),int(enemy.y + 30)) and bullet1.x in range(int(enemy.x-10),int(enemy.x + 50)):

hero.bullet\_list.remove(bullet1)

enemy.bomb()

if bullet1.y < 0 or bullet1.x < 0 or bullet1.x > 480: #删除越界的玩家子弹

hero.bullet\_list.remove(bullet1)

def judge3(hero,boss):

#判断boss的炸毁

global goal, g, goal0

for bullet3 in hero.bullet\_list:

if bullet3.y in range(int(boss.y), int(boss.y + 60)) and bullet3.x in range(int(boss.x), int(boss.x + 100)):

hero.bullet\_list.remove(bullet3)

g += 1

boss.image = boss.bomb\_list[g]

print("g = %d"%g)

if g >= 6:

boss.y, g, goal = 0, 0, 0

boss.bomb()

goal0 += 10

def clear\_enemy(enemies):

#清除敌机群类中被炸毁的敌机

global goal, goal0

for enemy in enemies.enemy\_list:

if enemy.hit == True and enemy.image\_index == 3:

enemies.enemy\_list.remove(enemy)

enemies.num -= 1

goal += 1

goal0 += 5

print("goal = %d"%goal)

if enemy.y >= 850:

enemies.enemy\_list.remove(enemy)

enemies.num -= 1

def judge\_num(enemies):

#判断频幕上敌人的数量，如果为零，继续添加敌人

n = random.randint(1,5)

if len(enemies.enemy\_list) == 0:

enemies.add\_enemy(n)

def show\_text(screen\_temp):

#在屏幕上显示文字

text = "GOAL:" + str(goal0) + "Life:" + str(life)

font\_size = 50

pos = (0,0)

color = (0,255,0)

cur\_font = pygame.font.SysFont("宋体",font\_size)

text\_fmt = cur\_font.render(text, 1, color)

screen\_temp.blit(text\_fmt, pos)

def creat\_bomb(screen\_temp):

bomb = bomb\_bullet(screen\_temp)

bomb\_list = []

bomb\_list.apend(bomb)

#定义的全局变量

goal = 0 #玩家得分

goal0 = 0

g = 0 #击中boss的次数

life = 100#生命值

s = 0 #判断大boss是否发射子弹

q = 0

def main():

#主函数执行

#获取事件，比如按键等

bb = False

move\_x = 0

move\_y = 0

pygame.init()

screen = pygame.display.set\_mode((480,852),0,32)

# 210,400

background = pygame.image.load("./feiji/background.png")

pygame.display.set\_caption("飞机大战")

atlas = pygame.image.load("./feiji/New Atlas.png")

#创建玩家飞机

hero = Hero(screen)

#创建敌机群

enemis = EnemyPlanes(screen)

enemis.add\_enemy(5)

#创建boss对象

boss = Boss(screen)

#创建炸弹对象

bomb = bomb\_bullet(screen)

#创建补给对象

supply0 = supply(screen)

clear = clear\_bullet(screen)

left\_key, right\_key, up\_key, down\_key, done = 0, 0, 0, 0, 0

# mark = 0#用来判断boss发射子弹

while True:

if done:

if done % 8 == 0:

done = 1

hero.fire()

else:

done += 1

for event in pygame.event.get():

#判断是否是点击了退出按钮

if event.type == QUIT:

print("exit")

exit()

#判断是否是按下了键

if event.type == KEYDOWN :

#down

#检测按键是否是a或者left

if event.key == K\_a or event.key == K\_LEFT:

#print('left')

move\_x = -5

left\_key += 1

#检测按键是否是d或者right

elif event.key == K\_d or event.key == K\_RIGHT:

#print('right')

move\_x = 5

right\_key += 1

elif event.key == K\_w or event.key == K\_UP:

move\_y = -5

up\_key += 1

elif event.key == K\_s or event.key == K\_DOWN:

move\_y = 5

down\_key += 1

#检测按键是否是空格键

elif event.key == K\_SPACE:

#print('space')

hero.fire()

done = 1

#enemis.fire()

elif event.key == K\_b:

print('b')

hero.bomb()

if event.type == KEYUP:

if event.key == K\_a or event.key == K\_LEFT:

left\_key -= 1

if right\_key == 0:

move\_x = 0

else:

move\_x = 5

if event.key == K\_d or event.key == K\_RIGHT:

right\_key -= 1

if left\_key == 0:

move\_x = 0

else:

move\_x = -5

if event.key == K\_w or event.key == K\_UP:

up\_key -= 1

if down\_key == 0:

move\_y = 0

else:

move\_y = 5

if event.key == K\_s or event.key == K\_DOWN:

down\_key -= 1

if up\_key == 0:

move\_y = 0

else:

move\_y = -5

if event.key == K\_SPACE:

done = 0

screen.blit(background, (0, 0))

hero.move(move\_x, move\_y)

hero.display()

hero.judge()

enemis.display(hero)

enemis.move()

enemis.fire()

bomb.display()

bomb.judge(hero)

bomb.move()

supply0.display()

supply0.judge(hero)

supply0.move()

#clear.display()

#clear.judge(hero, enemis)

#clear.move()

for i in range(enemis.num):

judge1(hero, enemis.enemy\_list[i])

#enemis.enemy\_list[i].judge(hero)

clear\_enemy(enemis)

judge\_num(enemis)

show\_text(screen)

if goal >= 15:

boss.display(hero)

boss.move()

# mark+=1

# if mark==8:

boss.fire()

# mark = 0

#boss.judge

judge3(hero, boss)

pygame.display.update()

if \_\_name\_\_ == "\_\_main\_\_":

main()