**完成飞机大战游戏设计**

【作业要求】

1. 自己敲一遍源代码，对自己负责
2. 写出代码的整体框架
3. 写出每个类及每个函数的作用
4. 整理代码实现过程中遇到的问题
5. 实验结果截图

1.创建窗口

import pygame  
import time

def main():

#1.创建窗口

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

#2 创建一个背景图片

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

while True:

#3 显示到屏幕

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

pygame.display.update()

time.sleep(0.05)

if \_\_name\_\_ == '\_\_main\_\_':

main()

2.创建一个玩家飞机，按方向键可以左右移动

import pygame

import time

from pygame.locals import \*

class HeroPlane(object):

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

self.x = 210

self.y = 700

self.screen = screen\_temp #游戏窗口

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

def display(self):

self.screen.blit(self.image,(self.x,self.y)) #加载玩家飞机到窗口

def move\_left(self):

self.x -= 5

def move\_right(self):

self.x += 5

def key\_control(hero\_temp):

# 获取事件，比如按键等

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

elif event.type == KEYDOWN:

# 检测按键是否是a或者left

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

print('left')

hero\_temp.move\_left()

# 检测按键是否是d或者right

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

print('right')

hero\_temp.move\_right()

# 检测按键是否是空格键

elif event.key == K\_SPACE:

print('space')

def main():

#1.创建窗口

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

#2 创建一个背景图片

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

# 创建一个飞机对象

hero = HeroPlane(screen)

while True:

#把背景图片放到窗口显示

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

#显示玩家飞机到定义的（x,y）坐标位置

hero.display()

pygame.display.update()

#检测键盘，控制玩家飞机移动

key\_control(hero)

time.sleep(0.05)

if \_\_name\_\_ == '\_\_main\_\_':

main()

3.给玩家飞机添加按空格键发射子弹功能

import pygame

import time

from pygame.locals import \*

class HeroPlane(object):

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

self.x = 210

self.y = 700

self.screen = screen\_temp #游戏窗口

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

self.bullet\_list = [] #存储发射出去的子弹

def display(self):

# 加载玩家飞机到窗口

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

for bullet in self.bullet\_list:

bullet.display()

bullet.move()

def move\_left(self):

self.x -= 5

def move\_right(self):

self.x += 5

def fire(self):

self.bullet\_list.append(Bullet(self.screen,self.x,self.y))

class Bullet(object):

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

self.x = x + 40

self.y = y - 20

self.screen = screen\_temp

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

def display(self):

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

def move(self):

self.y -= 20

def key\_control(hero\_temp):

# 获取事件，比如按键等

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

elif event.type == KEYDOWN:

# 检测按键是否是a或者left

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

print('left')

hero\_temp.move\_left()

# 检测按键是否是d或者right

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

print('right')

hero\_temp.move\_right()

# 检测按键是否是空格键

elif event.key == K\_SPACE:

print('space')

hero\_temp.fire()

def main():

#1.创建窗口

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

#2 创建一个背景图片

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

# 创建一个飞机对象

hero = HeroPlane(screen)

while True:

#把背景图片放到窗口显示

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

#显示玩家飞机到定义的（x,y）坐标位置

hero.display()

pygame.display.update()

#检测键盘，控制玩家飞机移动

key\_control(hero)

time.sleep(0.05)

if \_\_name\_\_ == '\_\_main\_\_':

main()

4.创建一个敌机

enemy = EnemyPlane(screen)

while True:

--snip--

enemy.display()

--snip

5.敌机自动左右移动

--snip--

class EnemyPlane(object):

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

--snip--

self.direction = 'right' #定义敌机默认往右移动

--snip--

def move(self):

if self.direction == 'right':

self.x += 8

elif self.direction == 'left':

self.x -= 8

if self.x > 430:

self.direction = 'left'

elif self.x < 0:

self.direction = 'right'

--snip--

def main():

--snip--

#4 创建一个敌机

enemy = EnemyPlane(screen)

enemy.move()

　　--snip—

6.删除子弹，只要发射出去的子弹超过边界就删掉

--snip--

class HeroPlane(object):

--snip--

--snip--

bullet.move()

if bullet.judge(): #判断子弹是否越界

self.bullet\_list.remove(bullet)

--snip--

class Bullet(object):

--sinp--

def judge(self):

if self.y < 0:

return True

else:

return False

--snip

7.敌机自动发射子弹

--snip--

import random

class EnemyPlane(object):

--snip - -

self.bullet\_list = [] #存储发射出去的子弹

self.direction = 'right' #定义敌机默认往右移动

def display(self):

--snip - -

bullet.move()

if bullet.judge(): #判断子弹是否越界

self.bullet\_list.remove(bullet)

--snip--

def fire(self):

#控制子弹频率

random\_num = random.randint(1,80)

if random\_num == 10 or random\_num == 40:

self.bullet\_list.append(EnemyBullet(self.screen,self.x,self.y))

--snip--

class EnemyBullet(object):

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

self.x = x + 25

self.y = y + 40

self.screen = screen\_temp

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

def display(self):

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

def move(self):

self.y += 10

def judge(self):

if self.y > 852:

return True

else:

return False

--snip - -

while True:

--snip - -

enemy.move()

enemy.fire()

--snip--