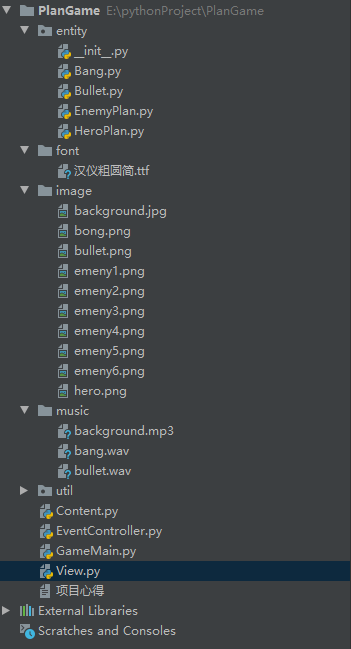
**项目名称：飞机大战**

第一次用Python从0开始做项目，虽然只是一个基础的飞机大战游戏，但是在做的过程中，还是遇到不少问题，但在这个过程中，我尽量将这学期所学应用其中，比如Python基础中的集合和其操作，不同数据类型和不同作用域的变量，以及装饰器模式等内容。同时尽量抽取其中的冗余代码，不断改进，力求代码的精简和修改的扩展性，同时几乎给每一行都添加了注释，来增强代码的可读性。



**首先是项目的目录结构介绍：**

1、在项目下包含两个python包体分别放置实体类和一个工具包，三个文件夹分别放置字体、图片和音效。

2、实体类包括我方飞机类、敌机类、子弹类、和爆炸类。

3、工具类包括距离计算。

4、在项目文件根目录下还有视图显示文件、全局变量及常量定义文件、和主方法类。

**游戏的主要功能：**

1、操控我方战机（鼠标操控模式，键盘操控两种模式）

2、发射子弹

3、攻击敌机

4、被敌机撞击

5、分数显示

6、血量显示

**主要角色流程：**

1、战机类：

战机

初始化战机（坐标、速度、样式、屏幕）

定义上下左右移动方法

定义开火方法

定义被撞击方法

鼠标拖动

键盘按键

发射子弹

监听

监听

移动位置

移动鼠标

移动位置

击中敌机

点击左键

未击中

WASD/↑↓←→

空格键

发射子弹

击中敌机

未击中

2、敌机类：

敌机

初始化敌机（随机横坐标、随机速度、随机样式、屏幕）

定义向下移动方法

定义撞击方法

定时器

产生敌机

向下飞行

随机速度

被子弹击中

与战机相撞

未被击中

移除

越界

**系统模块：**

1.系统初始化模块

功能描述：

（1） 打开游戏，初始化游戏，生成游戏角色

（2） 开始游戏

2.操控战机模块

功能描述

（1） 根据敌机出现的位置，手动改变我方战机的位置，避免相撞

（2） 采用鼠标操控或键盘操控战机运动方向和位置

3.记录分数模块

功能描述

（1） 根据游戏过程中消灭敌机数量增加分数

（2） 分数与销毁敌机同时变化，并记录到游戏界面指定位置

4.显示结果模块

功能描述

（1） 游戏过程战斗过程会随时根据结果记录数据

（2） 在战斗过程中记录游戏分数变化，并将结果反馈给游戏主界面

**系统完成过程优化总结、知识点总结：**

一、 面向对象的抽取

1、抽取出抽象的物体，比如我方飞机、子弹、敌机，以及抽取出键盘、鼠标操作的方法，界面显示的一些方法以及对一些常量的定义等

2、面向过程的开发和面向对象开发的对比：更加模块化，更加注重各个点的实现过程，不需要关注其他不相关模块的具体实现方式

二、在创建敌机的过程中，随机创建不同位置，不同样式的敌机想到的两种方式：

1、在敌机类中初始化时，随机产生一个浮点数，其值作为敌机出现横坐标，然后取整作为敌机图片名称，（emeny1.jpg，emeny2.jpg，...），从而达到敌机的随机化。

2、创建几种不同敌机实例，然后将其放入一个集合中，再随机从中取出一个敌机，从而实现随机化。

而在实现时采用第一种方式，原因是第二种比较死板，各种敌机类型需要自己手动创建，而第一种可以让系统自动实现不同的敌机，需要几架就创建几架，也体现了面向对象的思想。

三、装饰器模式

虽然游戏过程中非必须项，但是为了巩固知识点，还是在现实界面时，对显示分数、显示生命值和显示游戏结束的画面渲染添加了一个装饰器，来达到输出各函数名的功能。其中因为上述各函数均包含参数，且数量不一定，还使用到了\*args可变参数。

四、其他用到的课程内容：

1、随机生成数

2、集合、集合的操作

3、类型转换

4、可变量、不可变量的应用

5、包、模块的引用

五、课外补充：

Pygame库的学习

附：源码：

HeroPlan类

class HeroPlan(object):  
 def \_\_init\_\_(self, screen):  
 self.x = 385  
 self.y = 770  
 self.moveNum = 50 # 每次按键移动速度  
 self.image = pygame.image.load("image/hero.png")  
 self.screen = screen  
 self.life = 100 # 我方飞机总血量  
  
 # 显示我方飞机的方法  
 def display(self):  
 self.screen.blit(self.image, (self.x, self.y))  
 for bullet in bullet\_list:  
 bullet.display()  
 bullet.move()  
 if bullet.judge():  
 bullet\_list.remove(bullet)  
  
 # 想左移动的方法  
 def move\_left(self):  
 if self.x > 0:  
 self.x -= self.moveNum  
  
 def move\_right(self):  
 if self.x < 795:  
 self.x += self.moveNum  
  
 def move\_up(self):  
 if self.y > 0:  
 self.y -= self.moveNum  
  
 def move\_down(self):  
 if self.y < 770:  
 self.y += self.moveNum  
  
 # 飞机开火的方法  
 def fire(self):  
 # 向子弹集合中添加子弹  
 bullet\_list.append(Bullet.Bullet(self.screen, self.x, self.y))  
 con.bullet\_sound.play()  
  
 # 我方飞机被敌机撞击  
 def isHunt(self, emeny\_list):  
 # 计算每个敌机与我方飞机的距离是否构成撞击  
 for enemy in emeny\_list:  
 if(Distance.distance( self.x, self.y, enemy.x, enemy.y)): # 如果撞击  
 self.life -= 10 # 我方飞机血量减10  
 emeny\_list.remove(enemy) # 敌机坠毁，删除  
 con.bang\_sound.play() # 播放撞击有效  
 bong\_list.append(Bang.Bang(enemy.screen, enemy.x - 55, enemy.y - 55)) # 添加爆炸动画效果

2、敌机类

class EnemyPlan(object):  
 def \_\_init\_\_(self, screen):  
 # pygame.sprite.Sprite.\_\_init\_\_(self)  
 self.x = random.uniform(0, 950)  
 self.y = -100  
 self.image = pygame.image.load("image/emeny" + str(random.randint(1, 6)) + ".png")  
 self.screen = screen  
  
 # 显示敌机的方法  
 def display(self):  
 self.screen.blit(self.image, (self.x, self.y))  
  
 # 敌机以1-10的随机速度下降  
 def move(self):  
 self.y += random.uniform(1, 10)

3、子弹类：

# 子弹类  
class Bullet():  
 def \_\_init\_\_(self, screen, x, y):  
 self.x = x + 35  
 self.y = y + 5  
 self.moveNum = 10  
 self.image = pygame.image.load("image/bullet.png")  
 self.screen = screen  
  
 def display(self):  
 self.screen.blit(self.image, (self.x, self.y))  
  
 # 子弹向上移动方法  
 def move(self):  
 self.y -= 10  
  
 #判断子弹是否出界  
 def judge(self):  
 if self.y < -81:  
 return True  
 else:  
 return False  
  
 # 子弹击中敌机  
 def fit(self):  
 for enemy in emeny\_list:  
 for bullet in bullet\_list:  
 if (Distance.distance(bullet.x, bullet.y, enemy.x, enemy.y)): # 如果子弹和敌机相撞  
 emeny\_list.remove(enemy) # 则移除敌机  
 bullet\_list.remove(bullet) # 也移除子弹  
 con.bang\_sound.play() # 播放爆炸音效  
 con.score += 5 #分数+5  
 bong\_list.append(Bang.Bang(enemy.screen, enemy.x - 55, enemy.y - 55)) # 添加爆炸效果

4、爆炸类

# 爆炸效果类  
class Bang():  
 def \_\_init\_\_(self, screen, x, y):  
 self.x = x + 35  
 self.y = y + 5  
 self.image = pygame.image.load("image/bong.png")  
 self.screen = screen  
  
 def display(self):  
 self.screen.blit(self.image, (self.x, self.y))

5、常量、变量、自定义颜色、字体等

#初始化游戏  
pygame.init()  
#子弹的集合  
bullet\_list = []  
#敌机的集合  
emeny\_list = []  
#爆炸的集合  
bong\_list = []  
#全局变量分数值  
score = 0  
#设置分数和生命值的字体字号  
font = pygame.font.Font('./font/汉仪粗圆简.ttf', 25)  
#设定游戏结束字体和字号  
gameover\_font = pygame.font.Font('./font/汉仪粗圆简.ttf', 100)  
#创建敌机自定义事件  
CREATE\_ENEMY = pygame.USEREVENT+1  
pygame.time.set\_timer(CREATE\_ENEMY,1000)  
#移除爆炸效果自定义事件  
REMOVE\_BANG = pygame.USEREVENT+2  
pygame.time.set\_timer(REMOVE\_BANG,100)  
#设定游戏爆炸音效及音量  
bang\_sound = pygame.mixer.Sound("./music/bang.wav")  
bang\_sound.set\_volume(1)  
#设定子弹发射音效及音量  
bullet\_sound = pygame.mixer.Sound("./music/bullet.wav")  
bullet\_sound.set\_volume(0.7)

6、事件控制

def move\_contrller(hero, emeny\_list, screen):  
 for event in pygame.event.get():  
 if event.type == QUIT:  
 exit()  
 elif event.type == KEYDOWN: # 当操作方式为键盘时  
 if event.key == K\_LEFT or event.key == K\_a: #按下左键或者a  
 hero.move\_left()  
 elif event.key == K\_RIGHT or event.key == K\_d: # 按下右键或者d  
 hero.move\_right()  
 elif event.key == K\_DOWN or event.key == K\_s: # 按下下键或者s  
 hero.move\_down()  
 elif event.key == K\_UP or event.key == K\_w: #按下上键或者w  
 hero.move\_up()  
 elif event.key == K\_SPACE: # 按下空格键  
 hero.fire()  
 elif event.key == K\_ESCAPE: #按下esc键  
 exit()  
 elif event.type == MOUSEMOTION: # 当操作方式为鼠标时  
 pos = pygame.mouse.get\_pos() #获取鼠标坐标  
 hero.x = pos[0]-105  
 hero.y = pos[1]-50  
 elif event.type == MOUSEBUTTONDOWN: # 按下鼠标左键  
 pressed\_left = pygame.mouse.get\_pressed()[0]  
 if(pressed\_left):  
 hero.fire()  
 elif event.type == con.CREATE\_ENEMY: # 触发创建敌机事件  
 emeny\_list.append(EnemyPlan.EnemyPlan(screen))  
 elif event.type == con.REMOVE\_BANG: # 触发移除爆炸效果事件  
 if bong\_list:  
 bong\_list.pop(0)

7、显示元素

def show\_funcName(func):  
 def wrapper(\*args):  
 func(\*args)  
 print("执行了" + func.\_\_name\_\_ + "函数")  
 return wrapper  
  
# 渲染分数显示  
@ show\_funcName  
def show\_score(screen):  
 text = f"分数：{con.score}"  
 score\_render =con.font.render(text, True, (0, 255, 0))  
 screen.blit(score\_render, (10, 17))  
  
# 渲染显示生命值字  
@ show\_funcName  
def show\_life(screen, hero):  
 text = f"生命值：{hero.life}"  
 score\_render = con.font.render(text, True, (255, 255, 255))  
 screen.blit(score\_render, (750, 17))  
  
# 渲染游戏结束字符  
@ show\_funcName  
def gameover(screen, hero):  
 if(hero.life == 0):  
 emeny\_list.clear();  
 text = "GAME OVER"  
 score\_render =con.gameover\_font.render(text, True, (255, 0, 0))  
 screen.blit(score\_render, (200, 450))  
  
# 渲染血槽  
def show\_lifeRect(screen, hero):  
 pygame.draw.rect(screen, (255, 0, 0), (600, 15, 400, 30), 0)  
 pygame.draw.rect(screen, (0, 255, 0), (600, 15, hero.life \* 4, 30), 0)

8、主方法

import pygame  
import os  
from entity import HeroPlan  
import Content as con  
import EventController as ec  
import View as view  
  
#初始化游戏  
pygame.init()  
  
#获取Content中的全局集合变量  
bullet\_list = con.bullet\_list  
emeny\_list = con.emeny\_list  
bong\_list = con.bong\_list  
  
def main():  
 #设置窗口启动初始位置  
 os.environ['SDL\_VIDEO\_WINDOW\_POS'] = "%d,%d" % (30,40)  
 #创建窗体  
 screen = pygame.display.set\_mode((1006, 937), 0, 32)  
 #设置游戏名称  
 pygame.display.set\_caption("飞机大战")  
 #设置背景音乐  
 pygame.mixer.music.load("music/background.mp3")  
 pygame.mixer.music.set\_volume(0.42)  
 pygame.mixer.music.play(-1)  
 #设置刷新时钟频率  
 clock = pygame.time.Clock()  
 #创建背景图片  
 backgroundPath = "image/background.jpg"  
 background = pygame.image.load(backgroundPath)  
 #创建飞机  
 hero = HeroPlan.HeroPlan(screen)  
  
 while True:  
 # 设置背景图片  
 screen.blit(background, (0, 0))  
 # 设置飞机图片  
 hero.display()  
 # 控制飞机  
 ec.move\_contrller(hero, emeny\_list, screen)  
 for emeny in emeny\_list:  
 # 显示敌机  
 emeny.display()  
 # 敌机向下移动  
 emeny.move()  
 # 移除越界敌机  
 if(emeny.y > 937):  
 emeny\_list.remove(emeny)  
  
  
 # 子弹击中敌机  
 for bullet in bullet\_list:  
 bullet.fit()  
 # 被敌机撞击  
 hero.isHunt(emeny\_list)  
 # 显示爆炸烟花  
 for bong in bong\_list:  
 bong.display()  
 # 显示成绩  
 view.show\_score(screen)  
 # 显示血槽1  
 view.show\_lifeRect(screen,hero)  
 # 显示血槽2  
 view.show\_life(screen,hero)  
 # 显示游戏结束  
 view.gameover(screen, hero)  
 # 更新游戏画面  
 pygame.display.update()  
 clock.tick(30)  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 main()