# 准备工作：把03文件复制一份，改名04-dad-son-planewar.py，其他模块一样

|  |
| --- |
|  |

# 开始开发

## 1.新建一个bullet模块，在里面定义一个Bullet类

|  |
| --- |
|  |

## 2.在constants模块里面添加两个精灵组all\_sprites和bullets

|  |
| --- |
|  |

## 3.新建一个missile模块,在里面新建一个Missile类

|  |
| --- |
|  |

## 4..完善Plane类的shoot方法

|  |
| --- |
| def shoot(self):          now = pg.time.get\_ticks() # 获取现在的时间          if now - self.last\_shot > self.shoot\_delay:              self.last\_shot = now # 保存最新的时间              #单火力              if self.power ==1:                  bullet0 = Bullet(self.rect.centerx,self.rect.top)                  # bullet0 = Missile(self.rect.centerx,self.rect.top)                    # # 子弹需要添加到2个精灵组                  all\_sprites.add(bullet0)                  bullets.add(bullet0)                  shooting\_sound.play()              #双火力              if self.power ==2:                  bullet1 = Bullet(self.rect.left,self.rect.centery)                  bullet2 = Bullet(self.rect.right,self.rect.centery)                  all\_sprites.add(bullet1)                  bullets.add(bullet1)                  all\_sprites.add(bullet2)                  bullets.add(bullet2)                  shooting\_sound.play()              #三火力              if self.power >=3:                  bullet1 = Bullet(self.rect.left,self.rect.centery)                  bullet2 = Bullet(self.rect.right,self.rect.centery)                  missile1 = Missile(self.rect.centerx,self.rect.top)                  all\_sprites.add(bullet1)                  bullets.add(bullet1)                  all\_sprites.add(bullet2)                  bullets.add(bullet2)                  all\_sprites.add(missile1)                  bullets.add(missile1)                  shooting\_sound.play()                  missile\_sound.play() |

## 5.然后在04-dad-son-planewar.py里面添加创建飞机的代码

|  |
| --- |
|  |

## 只是添加了他们还是没有用，需要在游戏循环里面添加绘制和更新代码

|  |
| --- |
|  |

# 本节学习到此为止，本节所有模块的代码如下

## bullet.py

|  |
| --- |
| import pygame as pg  from constants import \*  shooting\_sound = pg.mixer.Sound(sound\_path+'pew-gunshot-13.wav')  class Bullet(pg.sprite.Sprite):      def \_\_init\_\_(self, x,y) -> None:          super().\_\_init\_\_()          self.image = pg.transform.scale(pg.image.load(pic\_path+'pd333.png'),(15,40))          # self.image = pg.transform.scale(pg.image.load(pic\_path+'enemy\_bullet2.png'),(15,40))          self.image.set\_colorkey(BLACK)          self.rect = self.image.get\_rect()          self.rect.bottom = y          self.rect.centerx = x          self.speed = -10      def update(self, ) -> None:         self.rect.y += self.speed         if self.rect.bottom < 0:             self.kill() # 子弹出界了就销毁 |

## constants.py

|  |
| --- |
| import pygame as pg  from os import path  pg.init()       #初始化pygame模块  pg.mixer.init() #初始化pygame模块的音效模块  font\_name = pg.font.match\_font('arial') # 获取系统里面安装了的字体名称  # 实战图片和音效图片路径  pic\_path = './res/images/'  sound\_path = './res/sounds/'  WIDTH = 480      #窗口宽度  HEIGHT = 600     #窗口高度  SIZE = (WIDTH,HEIGHT) #  FPS = 30   # 帧率  # 定义颜色  BLACK = (0,0,0)  WHITE = (255,255,255)  RED = (255,0,0)  GREEN = (0,255,0)  BLUE = (0,0,255)  YELLOW = (255,255,0)  bg = pg.image.load(pic\_path+'startfield.jpg')  bg = pg.transform.scale(bg,(WIDTH,1536))  height = -936  POWERUP\_TIME = 5000  #飞机的火力持续时间  # 精灵组  all\_sprites = pg.sprite.Group()  bullets = pg.sprite.Group() |

## funcs.py

|  |
| --- |
| import pygame as pg  from constants import \*  #绘制文本的函数  def draw\_text(surf,text,size,x,y):      font = pg.font.Font(font\_name,size)      text\_surface = font.render(text,True,WHITE)      text\_rect = text\_surface.get\_rect()      text\_rect.midtop = (x,y)      surf.blit(text\_surface,text\_rect) |

## menu.py

|  |
| --- |
| from os import path  import pygame as pg  from constants import \*  import funcs  def main\_menu(screen):      #加载菜单音乐      pg.mixer.music.load(sound\_path + 'menu.ogg')      #循环播放菜单音乐      pg.mixer.music.play(-1)      # 加载开始图片      start\_img = pg.image.load(pic\_path+'menu.png')      start\_img = pg.transform.scale(start\_img,SIZE)      screen.blit(start\_img,(0,0))      pg.display.update()      while True:          event = pg.event.poll() # 只获取一个事件          if event.type == pg.KEYDOWN:              if event.key == pg.K\_RETURN: # 回车键                  break          elif  event.type == pg.QUIT:              pg.quit()              quit()          else:              funcs.draw\_text(screen,"Press [Enter] To Begin",30,WIDTH/2,HEIGHT/2)              funcs.draw\_text(screen,"[W] ↑",30,WIDTH/2,2\*HEIGHT/3-40)              funcs.draw\_text(screen,"[A]← [S] ↓ [D]→ ",30,WIDTH/2,2\*HEIGHT/3)              pg.display.update()              # 加载ready音效      pg.mixer.music.load(sound\_path + 'getready.ogg')      #循环播放菜单音乐      pg.mixer.music.play()      # ready = pg.mixer.Sound(sound\_path + 'getready.ogg')      # ready.play()      screen.fill(BLACK)      funcs.draw\_text(screen,"Get Ready!!!",40,WIDTH/2,HEIGHT/3)      pg.display.update()  def menu\_display(screen):      main\_menu(screen)      pg.time.wait(3000)      pg.mixer.music.stop()      pg.mixer.music.load(sound\_path+'tgfcoder-FrozenJam-SeamlessLoop.ogg')      pg.mixer.music.play(-1) |

## missile.py

|  |
| --- |
| import pygame as pg  from constants import \*  missile\_sound = pg.mixer.Sound(sound\_path+'237071-Rocket\_Launcher-02.wav')  class Missile(pg.sprite.Sprite):      def \_\_init\_\_(self, x,y) -> None:          super().\_\_init\_\_()          self.image = pg.transform.scale(pg.image.load(pic\_path+'missile3.png'),(20,55))          self.image.set\_colorkey(BLACK)          self.rect = self.image.get\_rect()          self.rect.bottom = y          self.rect.centerx = x          self.speed = -10      def update(self, ) -> None:         self.rect.y += self.speed         if self.rect.bottom < 0:             self.kill() # 子弹出界了就销毁 |

## plane.py

|  |
| --- |
| from typing import Any  import pygame as pg  from constants import \*  from bullet import \*  from missile import \*  # 加载玩家飞机图片  player\_img1 = pg.image.load(pic\_path+"my1.png")  player\_mini\_img1 = pg.transform.scale(player\_img1,(20,19))  player\_mini\_img1.set\_colorkey(BLACK)  player\_img2 = pg.image.load(pic\_path+"my2.png")  player\_mini\_img2 = pg.transform.scale(player\_img2,(30,19))  player\_mini\_img2.set\_colorkey(BLACK)    class Plane(pg.sprite.Sprite):      """"玩家飞机类"""      def \_\_init\_\_(self, playerImg,K\_LEFT,K\_RIGHT,K\_UP,K\_DOWN) -> None:          super().\_\_init\_\_()          self.image = pg.transform.scale(playerImg,(50,38))          self.image.set\_colorkey(BLACK)          self.rect = self.image.get\_rect()          self.rect.centerx = WIDTH/2          self.rect.bottom = HEIGHT-10          self.speed = 5          self.shield = 100 #血量          self.redius = 20 #杀伤力          self.shoot\_delay = 250 #子弹延迟          self.last\_shot = pg.time.get\_ticks() #最后一次射击时间          self.lives = 3 # 飞机架数          self.hidden = False          self.hide\_timer = pg.time.get\_ticks()          self.power = 1          self.power\_timer = pg.time.get\_ticks() # 火力时间          self.K\_LEFT = K\_LEFT          self.K\_RIGHT = K\_RIGHT          self.K\_UP = K\_UP          self.K\_DOWN = K\_DOWN      def update(self) -> None:          # super().update()          if self.power >=2 and pg.time.get\_ticks() - self.power\_timer > POWERUP\_TIME:              self.power -= 1              self.power\_timer = pg.time.get\_ticks()          if self.hidden and  pg.time.get\_ticks() - self.hide\_timer > 1000:              self.hidden = False              self.rect.centerx = WIDTH/2              self.rect.bottom = HEIGHT - 30          self.shoot()    # 是自动发射子弹的          self.move()     # 设置玩家移动边界        def move(self):          keys = pg.key.get\_pressed() # 获取所有按下的键          if keys[self.K\_RIGHT]:              if self.rect.right > WIDTH: # 右边越界                  self.rect.right = WIDTH              else:                  self.rect.centerx += self.speed          if keys[self.K\_LEFT]:              if self.rect.left < 0 :# 左边越界                  self.rect.left = 0              else:                  self.rect.centerx -= self.speed          if keys[self.K\_UP]:              if self.rect.y < 10:                  self.rect.top = 10              else:                  self.rect.top -= self.speed          if keys[self.K\_DOWN]:              if self.rect.bottom > HEIGHT-10:                  self.rect.bottom = HEIGHT-10              else:                   self.rect.bottom += self.speed      def shoot(self):          now = pg.time.get\_ticks() # 获取现在的时间          if now - self.last\_shot > self.shoot\_delay:              self.last\_shot = now # 保存最新的时间              #单火力              if self.power ==1:                  bullet0 = Bullet(self.rect.centerx,self.rect.top)                  # bullet0 = Missile(self.rect.centerx,self.rect.top)                    # # 子弹需要添加到2个精灵组                  all\_sprites.add(bullet0)                  bullets.add(bullet0)                  shooting\_sound.play()              #双火力              if self.power ==2:                  bullet1 = Bullet(self.rect.left,self.rect.centery)                  bullet2 = Bullet(self.rect.right,self.rect.centery)                  all\_sprites.add(bullet1)                  bullets.add(bullet1)                  all\_sprites.add(bullet2)                  bullets.add(bullet2)                  shooting\_sound.play()              #三火力              if self.power >=3:                  bullet1 = Bullet(self.rect.left,self.rect.centery)                  bullet2 = Bullet(self.rect.right,self.rect.centery)                  missile1 = Missile(self.rect.centerx,self.rect.top)                  all\_sprites.add(bullet1)                  bullets.add(bullet1)                  all\_sprites.add(bullet2)                  bullets.add(bullet2)                  all\_sprites.add(missile1)                  bullets.add(missile1)                  shooting\_sound.play()                  missile\_sound.play()      def powerup(self):          self.power += 1          self.power\_timer = pg.time.get\_ticks()      def hide(self):          self.hidden = True          self.hide\_timer = pg.time.get\_ticks()          self.rect.center = (WIDTH/2,HEIGHT+200) |

## 04-dad-son-planewar.py

|  |
| --- |
| # 第4个版本，添加子弹  import sys  from constants import \*  import menu  from plane import \*  screen = pg.display.set\_mode(SIZE)  pg.display.set\_caption("飞机大战")  clock = pg.time.Clock()  def main():      # 1.显示启动画面      menu.menu\_display(screen) # 调用menu模块的显示菜单方法      # 2.创建一个all\_sprites精灵组      # all\_sprites = pg.sprite.Group()      # 3.创建玩家飞机      player1 = Plane(player\_img1,pg.K\_LEFT,pg.K\_RIGHT,pg.K\_UP,pg.K\_DOWN)      player2 = Plane(player\_img2,pg.K\_a,pg.K\_d,pg.K\_w,pg.K\_s)      # 4.将他们添加到精灵组,这是必须的，因为只有精灵组才有绘制方法      all\_sprites.add(player1)      all\_sprites.add(player2)      # 调用精灵组的更新方法      all\_sprites.update()      global height      running = True      while running: # 游戏主循环            clock.tick(FPS)          for event in pg.event.get():              if event.type == pg.QUIT:                  running = False          # 背景图片向下滚动          screen.blit(bg,(0,height))          height += 2          if height > -168:              height = -936          all\_sprites.draw(screen)   # 绘制精灵          all\_sprites.update()       # 调用精灵组的更新方法          pg.display.update()        pg.quit()      sys.exit()  if \_\_name\_\_ == '\_\_main\_\_':      main() |