# 这个游戏没有结束处理，我们来实现以下

# 先把上面的优化项目单机版本复制一份，改名：dad-son-plane-war - simple-player-better

|  |
| --- |
|  |

# 代码改进

## 1.在menu.py模块里面新增一个quit\_menu方法

|  |
| --- |
| def quit\_menu(screen):      pg.mixer.music.stop()      screen.fill(BLACK)      funcs.draw\_end\_text(screen,"Game Over,Press Enter To Exit",30,WIDTH/2,HEIGHT/2)      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()      pg.display.update() |

## 2.在这个函数里面调用了一个叫做draw\_end\_text的方法，我们在funcs模块里面定义这个方法

|  |
| --- |
|  |

## 3.然后在主循环里面调用quit\_menu方法

|  |
| --- |
|  |

# 效果：不是很完美但是基本上解决了没有游戏结束功能的问题，以后我们需要把它改造成为游戏结束后可以重新再玩的版本

# 有代码更新的模块的完整代码

## 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 + 'tgfcoder-FrozenJam-SeamlessLoop.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,"[↑]",30,WIDTH/2,2\*HEIGHT/3-40)              funcs.draw\_text(screen,"[←] [↓]  [→] ",30,WIDTH/2,2\*HEIGHT/3)              pg.display.update()        pg.display.update()  def menu\_display(screen):      main\_menu(screen)      pg.mixer.music.stop()      pg.mixer.music.load(sound\_path+'battle.ogg')      pg.mixer.music.set\_volume(0.8)      pg.mixer.music.play(-1)  def quit\_menu(screen): # 显示game over界面的功能      pg.mixer.music.stop()      screen.fill(BLACK)      funcs.draw\_end\_text(screen,"Game Over,Press Enter To Exit",30,WIDTH/2,HEIGHT/2)      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()      pg.display.update() |

## funcs.py

|  |
| --- |
| import pygame as pg  from constants import \*  from plane import \*  from enemy import \*  from explosion import \*  import random as rnd  from power import \*  from lava 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)  def draw\_end\_text(surf,text,size,x,y): #显示Game Over信息      font = pg.font.Font(font\_name,size)      text\_surface = font.render(text,True,RED)      text\_rect = text\_surface.get\_rect()      text\_rect.midtop = (x,y)      surf.blit(text\_surface,text\_rect)  def draw\_screen\_text(screen,player1):      draw\_text(screen,str(score),18,WIDTH/2,10) #显示分数      draw\_shield\_bar(screen,5,5,player1.shield)      draw\_lives(screen,10,20,player1.lives,player\_mini\_img1)    # 绘制血条  def draw\_shield\_bar(screen,x,y,pct):      pct = max(pct,0)      fill = (pct/100) \* BAR\_LENGTH      outline\_rect = pg.Rect(x,y,BAR\_LENGTH,BAR\_HEIGHT)      fill\_rect = pg.Rect(x,y,fill,BAR\_HEIGHT)      pg.draw.rect(screen,GREEN,fill\_rect)      pg.draw.rect(screen,WHITE,outline\_rect,2)  def draw\_lives(surf,x,y,lives,img):      for i in range(lives):          img\_rect = img.get\_rect()          img\_rect.x = x + 30\*i          img\_rect.y = y          surf.blit(img,img\_rect)  # 创建敌机的函数  def new\_enemy():      enemy = Enemy()      all\_sprites.add(enemy)      enemies.add(enemy)  def bullet\_hit\_enemy():      """我方子弹打中敌人的函数"""      # 先进行我方子弹和敌机的碰撞检测      global score      hits = pg.sprite.groupcollide(enemies,bullets,True,True,pg.sprite.collide\_mask)      for hit in hits:          score += 50-hit.radius          pg.mixer.Sound(sound\_path+"exp.wav").play()          #创建一个爆炸对象需要调用Explosion类          expl = Explosion(hit.rect.center,'lg')          # 将爆炸对象添加到所有精灵组          all\_sprites.add(expl)          if rnd.random()> 0.9:              pow = Power(hit.rect.center)              all\_sprites.add(pow)              powers.add(pow)          # 每消灭一个敌机，又会创建一个敌机          new\_enemy()  # 我方飞机获取补给的方法，元素碰撞检测  def plane\_get\_power(player):      sound = pg.mixer.Sound(sound\_path+'FX054\_cut.wav')      hits = pg.sprite.spritecollide(player,powers,True)      for hit in hits:          if hit.type == 'shield':              sound.play()              player.shield += rnd.randrange(20,40)              if player.shield >=100:                  player.shield = 100 # 血量不能超过100          elif hit.type == 'gun':              sound.play()              player.powerup()    #敌机子弹打中我方飞机  def enemy\_hit\_me(player):      hits = pg.sprite.spritecollide(player,enemy\_bullets,True,pg.sprite.collide\_mask)      for h in hits:          player.shield -= h.radius \*2 # 被打中会掉血          expl = Explosion(h.rect.center,'sm') # 创建爆炸对象，添加到小爆炸里集合面          all\_sprites.add(expl) # 将爆炸对象添加到所有精灵组          if player.shield <=0: # 血量掉光了就死掉了              pg.mixer.Sound(sound\_path+'exp.wav').play() #播放爆炸音效              dead\_expl = Explosion(player.rect.center,'player')              all\_sprites.add(dead\_expl)              player.hide() # 调用这个方法后几秒钟就会显示player              player.lives -= 1 # 死掉了，就要减少一条命              player.shield = 100 # 把player的血量设置位100，那么他就相当于新创建的了  # 我方飞机和敌机的碰撞检测  def plane\_crash(player):      hits = pg.sprite.spritecollide(player,enemies,True,pg.sprite.collide\_mask)      for hit in hits:          pg.mixer.Sound(sound\_path+"exp.wav").play()          #创建一个爆炸对象需要调用Explosion类          expl = Explosion(hit.rect.center,'lg')          # 将爆炸对象添加到所有精灵组          all\_sprites.add(expl)          if rnd.random()> 0.9:              pow = Power(hit.rect.center)              all\_sprites.add(pow)              powers.add(pow)          # 每消灭一个敌机，又会创建一个敌机          new\_enemy()  #创建熔岩碎石  def new\_lava():      if rnd.random() >= 0.6:          lava = Lava()          all\_sprites.add(lava)          lavas.add(lava)  # 子弹到熔岩碎石的碰撞检测，应该双方的子弹都功能打熔岩  def bullet\_hit\_lava():      hits = pg.sprite.groupcollide(bullets,lavas,True,True,pg.sprite.collide\_mask)      for hit in hits:          expl = Explosion(hit.rect.center,'lg')          all\_sprites.add(expl)          new\_lava()      hits = pg.sprite.groupcollide(enemy\_bullets,lavas,True,True,pg.sprite.collide\_mask)      for hit in hits:          expl = Explosion(hit.rect.center,'lg')          all\_sprites.add(expl)          new\_lava()  # 熔岩碎石撞击双方飞机，我方是单人玩家，没有玩家精灵组  def lava\_hit\_both\_planes(player):      hits = pg.sprite.spritecollide(player,lavas,True,pg.sprite.collide\_mask)      for hit in hits:          expl = Explosion(hit.rect.center,'sm')          all\_sprites.add(expl)          player.shield -= hit.radius      for enemy in enemies:          hits = pg.sprite.spritecollide(enemy,lavas,True,pg.sprite.collide\_mask)          for hit in hits:              expl = Explosion(hit.rect.center,'lg')              all\_sprites.add(expl)              enemy.kill()              new\_enemy()  # 双方子弹的碰撞检测  def bullet\_vs\_enemy\_bullet():      hits = pg.sprite.groupcollide(bullets,enemy\_bullets,True,True,pg.sprite.collide\_mask)      for hit in hits:          expl = Explosion(hit.rect.center,'sm')          all\_sprites.add(expl) |

## 10-dad-son-planewar.py

|  |
| --- |
| # 第10个版本，我方受伤  import sys  from constants import \*  import menu  from plane import \*  from funcs 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.精灵组可以直接使用constants模块里面的all\_sprites和bullets      # 3.创建玩家飞机      player1 = Plane(player\_img1,WIDTH/2,HEIGHT-30,pg.K\_LEFT,pg.K\_RIGHT,pg.K\_UP,pg.K\_DOWN)      # 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)        # 创建敌机      for i in range(4):          new\_enemy() # 这个方法可以创建敌机并且添加到all\_sprites和enemies精灵组里面      # 创建熔岩碎石      for i in range(4):          new\_lava()      # 调用精灵组的更新方法      all\_sprites.update()        global height      running = True      while running: # 游戏主循环          if player1.lives == 0: # 判断是否是游戏结束             player1.shield = 0             running = False             menu.quit\_menu(screen)            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.update()       # 调用精灵组的更新方法          # 调用精灵组的更新方法          all\_sprites.draw(screen)        # 绘制精灵          bullet\_hit\_enemy()              # 子弹打中敌人的碰撞检测          plane\_get\_power(player1)        #我方飞机获取补给的碰撞检测          enemy\_hit\_me(player1)           #敌机子弹打中我方飞机          plane\_crash(player1)            #我方飞机和敌机碰撞检测          bullet\_hit\_lava()               #双方子弹打熔岩碎石的碰撞检测          lava\_hit\_both\_planes(player1)   #熔岩碎石碰撞双方飞机的碰撞检测          bullet\_vs\_enemy\_bullet()         # 双方子弹的碰撞检测          draw\_screen\_text(screen,player1) # 绘制血条和飞机架数            pg.display.update()        pg.quit()      sys.exit()  if \_\_name\_\_ == '\_\_main\_\_':      main() |