# 项目以及文件准备

|  |
| --- |
|  |

# 1.项目搭建

## 1.打开vscode-insider，新建一个文件夹：plane-war-heima，然后创建上面的4个文件res文件，放入游戏图片，音乐和字体素材到对应的文件夹

|  |
| --- |
|  |

# 2.游戏类的设计

## 1）游戏类的属性和方法说明

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |

## 2）代码实现

|  |
| --- |
|  |
|  |
|  |

### 给Game类的reset\_game方法添加代码

|  |
| --- |
|  |

### 给Game类创建一个start方法并且添加代码如下

|  |
| --- |
|  |

### 给Game类添加一个event\_handler方法并且添加退出事件监听代码

|  |
| --- |
|  |

### 然后在start方法中调用这个event\_handler方法

|  |
| --- |
|  |

### 把启动代码改为下面的代码

|  |
| --- |
|  |

### 经过测试，点击窗口关闭按钮和按esc键都是可以退出程序的

### 然后我们添加按空格键的事件处理

|  |
| --- |
|  |

### 在start方法中添加刷新界面的方法已经模拟工具游戏状态显示不同内容

|  |
| --- |
| #游戏主程序  import pygame  from game\_items import \*  from game\_hud import \*  from game\_music import \*  class Game(object):      def \_\_init\_\_(self) -> None:         #1.游戏主窗口         self.main\_window = pygame.display.set\_mode(SCREEN\_RECT.size)         #设置窗口标题         pygame.display.set\_caption('飞机大战')         #2.游戏状态属性         self.is\_game\_over = False #游戏结束标记         self.is\_pause = False    #游戏暂停标记           #3.音乐播放器           #4.创建精灵         #5.精灵组属性      def reset\_game(self):        """重置游戏"""        self.is\_game\_over = False #游戏结束标记        self.is\_pause = False    #游戏暂停标记      def creat\_enemies(self):          """创建敌机"""          pass      def creat\_supplies(self):          """创建道具"""          pass      def start(self):          """开始游戏"""          #创建游戏时钟          clock = pygame.time.Clock()          #游戏主循环          while True:              #监听事件，调用event\_handler方法              if self.event\_handler(): #返回True说明用户点击了关闭按钮                  return              # 根据游戏状态切换界面显示的内容              if self.is\_game\_over:                  print('game over press space start new game')              elif self.is\_pause:                  print('game paused...press space to resume')              else:                  print('game is playing...press space to pause')              #刷新界面              pygame.display.update()              #设置游戏的刷新频率              clock.tick(60)      def event\_handler(self):          """事件箭头"""          for event in pygame.event.get():              if event.type == pygame.QUIT: #处理点击窗口的关闭按钮的退出事件                  return True              elif event.type == pygame.KEYDOWN and event.key==pygame.K\_ESCAPE: # 按esc键程序也会退出                  return True              elif event.type == pygame.KEYDOWN and event.key==pygame.K\_SPACE: # 按空格键                  #如果游戏已经结束                  if self.is\_game\_over:                      self.reset\_game() #重置游戏                  else:                      self.is\_pause = not self.is\_pause # 如果没有结束，就可以进行暂停和继续的切换            return False #其他情况都不会退出程序      def check\_collide(self):          """碰撞检测"""          pass    if \_\_name\_\_ == '\_\_main\_\_':      game = Game()      game.start() |

### 效果：

|  |
| --- |
|  |

### 主代码里面添加任务流程

|  |
| --- |
|  |

## 这个阶段的代码

### game.py

|  |
| --- |
| #游戏主程序  import pygame  from game\_items import \*  from game\_hud import \*  from game\_music import \*  class Game(object):      def \_\_init\_\_(self) -> None:         #1.游戏主窗口         self.main\_window = pygame.display.set\_mode(SCREEN\_RECT.size)         #设置窗口标题         pygame.display.set\_caption('飞机大战')         #2.游戏状态属性         self.is\_game\_over = False #游戏结束标记         self.is\_pause = False    #游戏暂停标记           #3.音乐播放器           #4.创建精灵         #5.精灵组属性      def reset\_game(self):        """重置游戏"""        self.is\_game\_over = False #游戏结束标记        self.is\_pause = False    #游戏暂停标记      def creat\_enemies(self):          """创建敌机"""          pass      def creat\_supplies(self):          """创建道具"""          pass      def start(self):          """开始游戏"""          #创建游戏时钟          clock = pygame.time.Clock()          #游戏主循环          while True:              #监听事件，调用event\_handler方法              if self.event\_handler(): #返回True说明用户点击了关闭按钮                  return              # 根据游戏状态切换界面显示的内容              if self.is\_game\_over:                  print('game over press space start new game')              elif self.is\_pause:                  print('game paused...press space to resume')              else:                  print('game is playing...press space to pause')              #刷新界面              pygame.display.update()              #设置游戏的刷新频率              clock.tick(60)      def event\_handler(self):          """事件箭头"""          for event in pygame.event.get():              if event.type == pygame.QUIT: #处理点击窗口的关闭按钮的退出事件                  return True              elif event.type == pygame.KEYDOWN and event.key==pygame.K\_ESCAPE: # 按esc键程序也会退出                  return True              elif event.type == pygame.KEYDOWN and event.key==pygame.K\_SPACE: # 按空格键                  #如果游戏已经结束                  if self.is\_game\_over:                      self.reset\_game() #重置游戏                  else:                      self.is\_pause = not self.is\_pause # 如果没有结束，就可以进行暂停和继续的切换            return False #其他情况都不会退出程序      def check\_collide(self):          """碰撞检测"""          pass    if \_\_name\_\_ == '\_\_main\_\_':      #初始化游戏      pygame.init()      #开始游戏      Game().start()      #释放游戏资源      pygame.quit() |

### game\_item.py

|  |
| --- |
| #封装游戏元素对应的类  import pygame  # 定义全局常量  #1.游戏主窗口区域  SCREEN\_RECT = pygame.Rect(0,0,480,700) |

# 阶段总结