# 第四次作业

## 1.项目介绍

本项目主要参考了《Python编程：从入门到实践》一书中的“项目1 外星人入侵”。并按照自己的理解对其中的部分代码和游戏内容进行了修改，从而创作出了这款魔改版的“玩家大战马化腾”。点击中间的PLAY按钮开始游戏，上面是一排排马化腾的表情包，提示玩家：“年轻人，该充钱了”，下面是我们呆萌的玩家。玩家通过左右方向键控制移动，空格键发射子弹。本游戏没有设置玩家胜利，游戏会一直循环直到玩家失败，失败后会弹出充值￥648的界面，和马化腾充八万舞蹈视频，寓意玩家永远玩不过老马IMG_256。

项目依赖：python-3.8.5，库：pygame-2.0.1、moviepy-1.0.3

运行游戏的方式：①进入exe文件/中直接双击“启动游戏”运行[exe](exe文件/启动游戏.exe)文件。

②安装以上依赖项后，进入到python源文件/，运行alien \_invasion.py文件。

运行结果示例：[操作视频](操作视频.mp4)

## 2.Python代码介绍

本项目的所有文件如下图所示，images目录下存放着项目的图片，sounds目录下存放着项目的音乐，video目录下存放着项目的视频。alien.py用于设置怪物的信息，alien\_invasion.py是用于控制游戏运行的主文件，bullet.py用于设置子弹的信息，button.py用于设置游戏的开关按钮，game\_functions.py用于记录游戏功能相关的函数，game\_stats.py用于记录游戏的统计信息，settings.py用于储存游戏的一些参数，ship.py用于设置玩家的信息。下面分别介绍每个文件的代码结构和含义，详细信息请见各文件代码。



**2.1 alien\_invasion.py**

定义了一个run\_game()函数，最后一行调用，用于运行游戏。pygame.init() 初始化背景设置。screen = pygame.display.set\_mode()创建一个名为screen 的显示窗口。pygame.display.set\_caption()设置窗口名称。ai\_settings = Settings()、 play\_button = Button()、ship = Ship()、bullets = Group()、aliens = Group()、stats = GameStats()调用其他文件中设置的对象信息。while循环用于侦听事件，并根据发生的事件执行相应的任务。

**2.2 settings.py**

self.screen设置了屏幕参数，self.bullet设置了子弹参数，self.alien、self.fleet设置了怪物参数，self.ship设置了玩家参数。

**2.3 ship.py**

self.image = pygame.image.load('images/ship.jpg')导入玩家图片。self.rect = self.image.get\_rect()获取位置属性。self.rect.centerx、self.rect.bottom设置玩家位置。定义了update()函数，根据移动标志调整玩家的位置。定义了blitme()函数，用于在指定的位置将图像绘制到屏幕上。定义了center\_ship()函数，让玩家图片在屏幕上居中。

**2.4 alien.py**

创建了表示单个怪物的类Alien。self.image、self.rect设置了怪物图片和位置。check\_edges()用于检测怪物是否位于屏幕边缘，update()让怪物移动。

**2.5 bullet.py**

创建了一个对子弹进行管理的类Bullet。self.image、self.rect设置了子弹图片和位置。self.speed\_factor设置子弹的速度。update()设置子弹移动，draw\_bullet()将子弹绘制在屏幕上。

**2.6 button.py**

创建了一个对开始按钮进行管理的类Button。self.image=pygame.image. load('images/button.png')导入按钮图片。self.rect、prep\_msg()设置按钮的位置。

**2.7 game\_stats.py**

定义了GameStats类用于跟踪游戏的统计信息， self.game\_active = False让游戏一开始处于非活动状态。reset\_stats初始化在游戏运行期间可能变化的统计信息。

**2.8 game\_functions.py**

定义了check\_keydown\_events()用于响应键盘按下，check\_keyup\_events()用于响应键盘松开，check\_events()响应键盘和鼠标事件，update\_screen()用于更新屏幕上的图像，update\_bullets用于更新子弹的位置，并删除已消失的子弹，check\_bullet\_alien\_collisions()用于响应子弹和怪物的碰撞，get\_number\_aliens \_x()、get\_number\_rows()、create\_alien()、create\_fleet()用于创建多行怪物，check\_fleet\_edges()用于响应有怪物到达边缘时采取相应的措施，change\_fleet\_direction()用于将整群怪物下移，并改变它们的方向，update\_aliens()用于检查是否有怪物位于屏幕边缘，并更新整群怪物的位置,ship\_hit()用于响应怪物撞到玩家，check\_aliens\_bottom()检查是否有怪物到达了屏幕底端,check\_play\_button()使玩家单击启动按钮时开始新游戏。

## 创新点

**3.1 增添游戏背景**

模板中采用bg\_color = (230, 230, 230)来设置背景颜色，太过单调，因此在game\_functions.py文件中修改了代码为：

background = pygame.image.load("images/background.jpg")

screen.blit(background, (0, 0))  # 对齐的坐标

pygame.display.update()  # 显示内容

并添加背景图片background.jpg：

**3.2 修改游戏子弹**

模板中采用self.bullet\_width = 3；self.bullet\_height = 15；self.bullet\_color = 60, 60, 60来创建子弹，并通过self.rect = pygame.Rect(0, 0, ai\_settings.bullet\_width, ai\_settings.bullet\_height)来设置子弹位置。本项目中修改了该部分，使得能够导入子弹的图片，在bullet.py文件中修改了代码为：

self.image = pygame.image.load('images/zhidan.bmp')

self.rect = self.image.get\_rect()

self.rect.centerx = ship.rect.centerx

zhidanself.rect.top = ship.rect.top

并添加子弹图片zhidan.bmp：

**3.3 新增游戏失败时弹出视频**

模板中游戏失败后会自动回到开始前的状态。本项目中新增了游戏失败后弹出标志游戏失败的视频，在game\_functions.py文件中新增了代码为：

from moviepy.editor import \*

def ship\_hit(ai\_settings, stats, screen,  ship, aliens, bullets): ...

     else:

        stats.game\_active = False

        pygame.mouse.set\_visible(True)

        #fail.mp4

        clip = VideoFileClip('./video/fail.mp4')

        clip.preview()

        pygame.quit()

并添加标志游戏失败的视频video/fail.mp4：

**3.4修改游戏的开始按钮**

模板中采用self.width, self.height = 200, 50；self.button\_color = (0, 255, 0)；self.text\_color = (255, 255, 255)；self.font = pygame.font.SysFont(None, 48)来设置按钮的尺寸和样式。本项目中修改了该部分，使得能够导入按钮的图片，在button.py文件中修改了代码为：

self.image = pygame.image.load('images/button.png')

def prep\_msg(self, msg):

        """将msg渲染为图像，并使其在按钮上居中"""

        self.msg\_image = self.image

        #self.font.render(msg, True, self.text\_color,self.button\_color)

        self.msg\_image\_rect = self.msg\_image.get\_rect()

        self.msg\_image\_rect.center = self.rect.center

并添加按钮的图片button.png：



**3.5添加游戏音乐**

模板中游戏没有任何声音。本项目中添加了背景音乐sounds/bgm.mp3，在alien\_ invasion.py文件中新增了代码：

  # 游戏音乐

    pygame.mixer.init()  # 初始化

    pygame.mixer.music.load('./sounds/bgm.mp3')  # 加载音乐文件

    pygame.mixer.music.play()  # 开始播放音乐流

添加了子弹发射时的音乐sounds/zhutou.mp3，在game\_functions.py文件中新增了代码：

elif event.key == pygame.K\_SPACE:

     # 创建一颗子弹，并将其加入到编组bullets中

     new\_bullet = Bullet(ai\_settings, screen, ship)

     bullets.add(new\_bullet)

     s = pygame.mixer.Sound("./sounds/zhutou.mp3")

     s.play()

**3.6修改游戏图像**

模板中采用的是飞船大战飞碟。本项目中将这些图像改为自己想要的样子，在alien.py文件中修改代码为：

self.image = pygame.image.load('images/laoma.bmp')

在ship.py文件中修改代码为：

self.image = pygame.image.load('images/ship.jpg')

并添加图片laoma.bmp和ship.jpg：

**3.7将python文件打包成exe文件**

为了方便在windows上运行游戏，我将本项目的所有python文件和依赖库打包成了exe文件。打包过程是通过pyinstaller完成，打包步骤如下：首先找到依赖库文件pygame和moviepy，以及想要的图标.ico文件，移动到python源文件中，然后运行以下命令：pyinstaller -i dnf.ico -F alien\_invasion.py -p alien.py -p bullet.py -p button.py -p game\_functions.py -p settings.py -p game\_stats.py -p ship.py -p pygame -p moviepy。打包完后会生成一个exe文件，将其重命名为“启动游戏”,再将images、sounds、video三个文件移动到exe文件相同的目录中，如下：



## 5.总结

因为最近正好在学习《Python编程：从入门到实践》这本书，所以选择了这个项目。通过这个项目的练习我学到了在开发大型项目时如何做好规划，如何利用多个依赖文件和类来有逻辑地来编写代码。学习到了如何给游戏添加背景图片、增添游戏中的元素、添加音乐、添加视频。通过在课堂上学习的python基础知识，加上自己自学pygame库，对书中的项目进行了修改，创作出了这部魔改版的游戏。因为本人非常喜欢DNF这款游戏，因此项目中采用了大量DNF游戏中的元素，仅供娱乐。本项目游戏的功能很简单，并且还有些地方有待优化，如图像闪烁等问题还需要更多时间来研究解决。

## IMG_2566.参考资料

1.《Python编程：从入门到实践》 项目1 外星人入侵

2.pygame官网：<https://www.pygame.org/news>

3.pygame教程：<https://blog.csdn.net/zha6476003/article/details/>829403 50

4.moviepy教程：<https://blog.csdn.net/ucsheep/article/details/8099993>9

5.pygame打包成exe教程：<https://www.bilibili.com/video/BV19t411M7ZD?t=> 539

6.素材来源：https://www.baidu.com、https://dnf.qq.com/