

陕西科技大学



外星人入侵游戏开发实验报告

学 生： _____

学 院： 电子信息与人工智能学院

专 业： 网络工程

指导教师： 丁磊

2022 年 5 月 2 日

实验一 外星人入侵游戏开发

班级： 网络 201 学号： _____

实验预习报告

一、实验目的

- 1、掌握常用的 python 语法
- 2、熟悉 python 的面向对象编程思维
- 3、学习 python 开发的流程

二、实验要求

参照教材《Python 编程：从入门到实践》第 12、13 章编写外星人入侵游戏，要求做到：

- 1、实现游戏的主要功能；
- 2、按照项目开发稳定的形式说明开发流程、各模块函数的功能；
- 3、实现部分额外的要求功能：

(1)、编写一个游戏，开始时屏幕中央有一个火箭，玩家可以使用四个方向键上下左右移动火箭，需要保证火箭不能被移动到屏幕以外；

(2)、找来一副星星的图像，随机的在屏幕上放置星星，使得星星的分布更加逼真。

三、实验原理

1、“外星人入侵”游戏，使用 pygame 软件包来开发，基本操作为：在玩家每次消灭一群向下移动准备“入侵”的外星人后，将玩家的等级和游戏难度均提高，随着难度的提高，最直观的表现移动的速度越来越快，难以击落移动的外星人，而玩家也会获得相应的激励，如得分越来越多、最高分会被记录下来。

2、通过不断地重构代码以管理文件内容，使得项目组织得更加的有序，同时也能提高项目的总体的效率。

3、在游戏中，玩家控制着一艘最初出现在屏幕底部中央的飞船。玩家可以

使用箭头键左右移动飞船，还可使用空格键进行射击。游戏开始时，一群外星人出现在天空中，他们在屏幕中向下移动。玩家的任务是射杀这些外星人。

玩家将所有外星人都消灭干净后，将出现一群新的外星人，他们移动的速度更快。只要有外星人撞到了玩家的飞船或到达了屏幕底部，玩家就损失一艘飞船。玩家损失三艘飞船后，游戏结束。

四、实验预习内容

1、什么是 Pygame?

pygame 是一个免费开源的跨平台库，用于使用 Python 开发多媒体应用程序，如视频游戏。它使用了 Simple DirectMedia Layer 库和其他一些流行的库，来抽象最常用的功能，使得编写这些程序更加直观。pygame 还提供了一个网站，用于展示和分享使用 pygame 制作的游戏、艺术、音乐、声音、视频和多媒体项目。pygame 还提供了一个高级的 sprite 模块，用于帮助组织游戏。

简而言之，pygame 是一个用 Python 编写多媒体应用程序的简单而强大的库，它可以用来开发常见的 python 应用，极大的提高了开发效率。

2、安装 pygame 库的方法

使用 pip 命令安装 pygame 库：

pip install pygame

由于国内网络环境的特殊性，可以使用清华大学开源软件镜像站来加速安装、提高安装库的成功率，如下命令：

pip install -i <https://pypi.tuna.tsinghua.edu.cn/simple> pygame

实验报告

一、实验目的

- 1、掌握常用的 python 语法
- 2、熟悉 python 的面向对象编程思维
- 3、学习 python 开发的流程

二、实验要求

参照教材《Python 编程：从入门到实践》第 12、13 章编写外星人入侵游戏，要求做到：

- 1、实现游戏的主要功能；
- 2、按照项目开发稳定的形式说明开发流程、各模块函数的功能；
- 3、实现部分额外的要求功能：

(1)、编写一个游戏，开始时屏幕中央有一个火箭，玩家可以使用四个方向键上下左右移动火箭，需要保证火箭不能被移动到屏幕以外；

(2)、找来一副星星的图像，随机的在屏幕上放置星星，使得星星的分布更加逼真。

三、实验原理

1、“外星人入侵”游戏，使用 pygame 软件包来开发，基本操作为：在玩家每次消灭一群向下移动准备“入侵”的外星人后，将玩家的等级和游戏难度均提高，随着难度的提高，最直观的表现移动的速度越来越快，难以击落移动的外星人，而玩家也会获得相应的激励，如得分越来越多、最高分会被记录下来。

2、通过不断地重构代码以管理文件内容，使得项目组织得更加的有序，同时也能提高项目的总体的效率。

3、在游戏中，玩家控制着一艘最初出现在屏幕底部中央的飞船。玩家可以使用箭头键左右移动飞船，还可使用空格键进行射击。游戏开始时，一群外星人出现在天空中，他们在屏幕中向下移动。玩家的任务是射杀这些外星人。

玩家将所有外星人都消灭干净后，将出现一群新的外星人，他们移动的速度

更快。只要有外星人撞到了玩家的飞船或到达了屏幕底部,玩家就损失一艘飞船。玩家损失三艘飞船后,游戏结束。

四、实验内容

1、安装 pygame 库

安装 python, 勾选将 python 添加到系统环境变量选项, 使用命令

pip -version

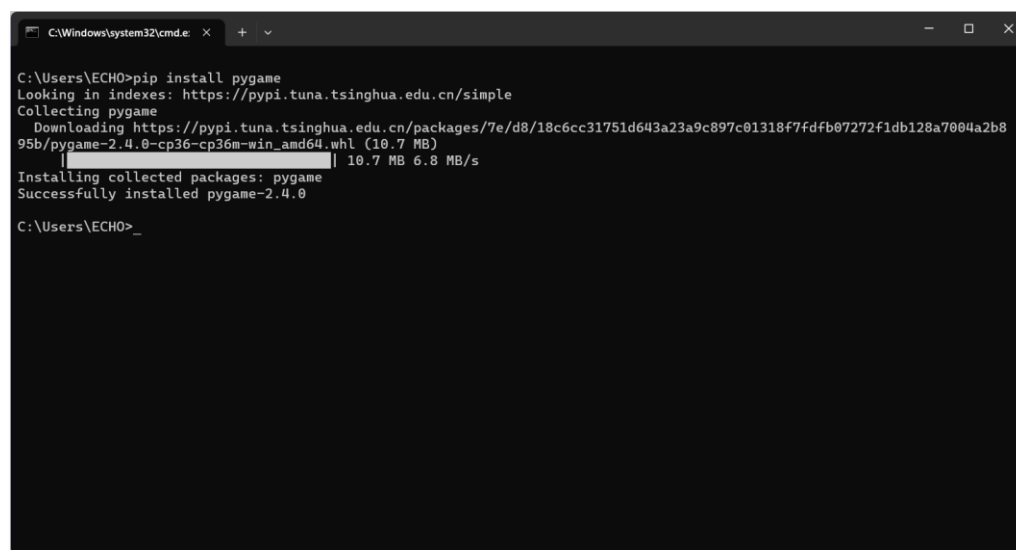
查看 pip 是否成功安装, 安装成功后, 使用以下命令将 pip 设为默认使用清华源:

pip config set global.index-url *https://pypi.tuna.tsinghua.edu.cn/simple*

设置清华源之后, 使用命令安装 pygame 库:

pip install *pygame*

安装结果如下:



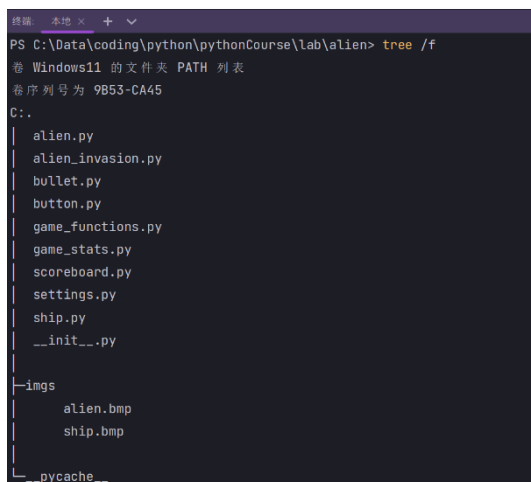
```
C:\Windows\system32\cmd.e...
C:\Users\ECHO>pip install pygame
Looking in indexes: https://pypi.tuna.tsinghua.edu.cn/simple
Collecting pygame
  Downloading https://pypi.tuna.tsinghua.edu.cn/packages/7e/d8/18c6cc31751d643a23a9c897c01318f7fdfb07272f1db128a7004a2b895b/pygame-2.4.0-cp36-cp36m-win_amd64.whl (10.7 MB)
    | 10.7 MB 6.8 MB/s
Installing collected packages: pygame
Successfully installed pygame-2.4.0
C:\Users\ECHO>_
```

2、创建 python 项目, 开始外星人入侵游戏开发

在存放代码的目录下, 新建 alien-revision 文件夹, 用来组织、存放项目的 python 脚本文件, 在集成开发环境当中打开这个目录, 这里使用 Pycharm Professional 2023.1.1, 按照实验指导书创建相关代码。

3、项目文件结构

项目所有文件存储于~/python/pythonCourse/lab/alien 目录下, 该目录的详细结构如下:



```
PS C:\Data\coding\python\pythonCourse\lab\alien> tree /f
查看 Windows11 的文件夹 PATH 列表
卷序列号为 9B53-CA45
C:.
|  alien.py
|  alien_invasion.py
|  bullet.py
|  button.py
|  game_functions.py
|  game_stats.py
|  scoreboard.py
|  settings.py
|  ship.py
|  __init__.py
|
|_ imgs
|   |  alien.bmp
|   |  ship.bmp
|
|_ __pycache__
```

其中，`imgs` 文件夹存储了项目所需要的静态图片文件，`pycache` 文件夹为 IDE 自动生成的缓存文件夹，`__init__.py` 为自动生成的包标记文件，纯空。主要文件有：`alien_invasion.py`、`alien.py`、`bullet.py`、`ship.py`、`button.py`、`game_states.py`、`settings.py`、`scoreboard.py`、`game_functions.py`

共 9 个主要文件，它们的作用分别如下：

(1)、`alien_invasion.py`

该文件中定义了游戏的入口、对游戏的相关参数进行了初始化、实现了对各功能的调用，保证了游戏可持续运行。

(2)、`alien.py`

该文件定义了“外星人”的行为及相应的特性、方法，配置了外星人对象的表现形式，使得可以被外部灵活调用。

(3)、`bullet.py`

该文件定义了飞船发出的子弹的行为及相应的特征。

(4)、`ship.py`

该文件定义了飞船的相关行为和特征。

(5)、`button.py`

由于 `pygame` 没有提供按钮，这个文件定义并实现了一个按钮的样式及其相应的功能。

(6)、`game_state.py`

这个文件当中实现了一个类用来在其实例里面动态存储当前游戏的状态。

(7)、`setting.py`

这个文件里面定义了一个类用来静态的保存游戏运行的时候的相关配置，修改这里面的对应属性可以实现相应的效果，同时也便于开发过程中的调试。

(8)、`scoreboard.py`

该文件里面定义了相应的属性和方法，来向用户展示当前的相关的游戏的状态，如计分、统计等功能。

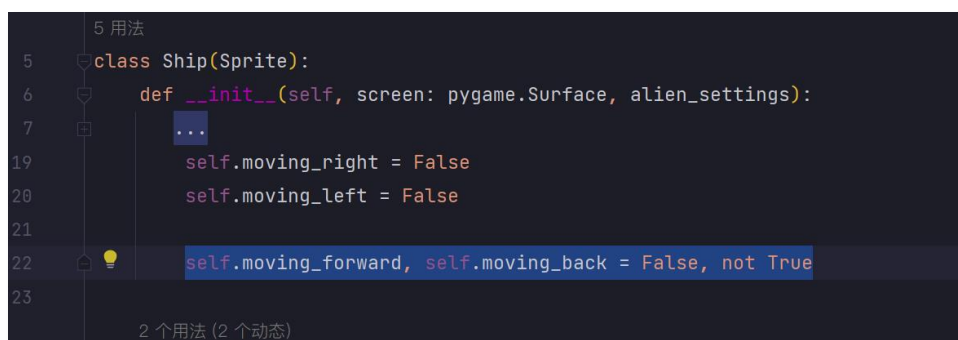
(9)、game_functions.py

这个文件是游戏当中最为重要的文件，游戏的总体执行逻辑被包括在了里面，相关函数实现了游戏当中的对应的功能。

4、要求的功能实现：

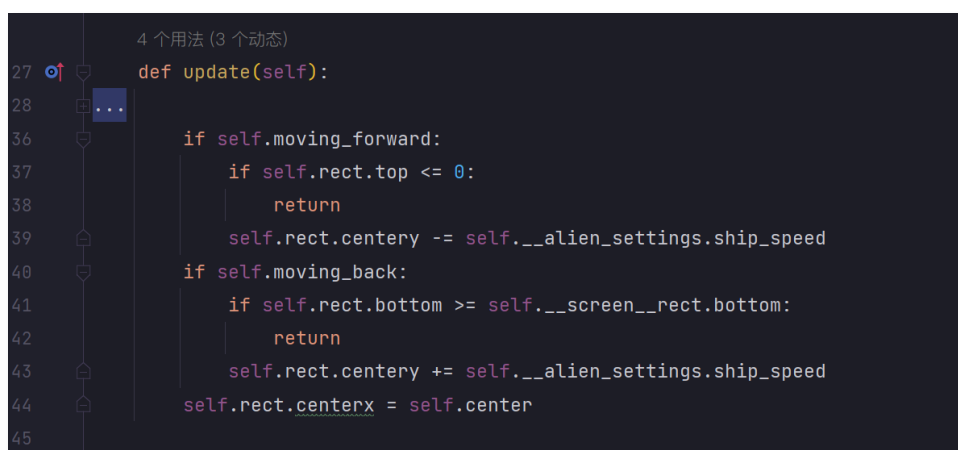
(1)、飞船的上下移动

要实现飞船的上下移动，可以借鉴左右移动的方法，为其增加一个是否允许上下移动的标志，这决定了它是否可以超过上下边界，则在 ship.py 中飞船的初始化函数中加入代码：



```
5 用法
5 class Ship(Sprite):
6     def __init__(self, screen: pygame.Surface, alien_settings):
7         ...
19         self.moving_right = False
20         self.moving_left = False
21
22         self.moving_forward, self.moving_back = False, not True
23
2 个用法 (2 个动态)
```

第 22 行代码表示，默认不允许前后移动，接着，在其更新方法里面加入前进、后退的代码：



```
4 个用法 (3 个动态)
27 def update(self):
28     ...
36     if self.moving_forward:
37         if self.rect.top <= 0:
38             return
39         self.rect.centery -= self.__alien_settings.ship_speed
40     if self.moving_back:
41         if self.rect.bottom >= self.__screen__rect.bottom:
42             return
43         self.rect.centery += self.__alien_settings.ship_speed
44     self.rect.centerx = self.center
45
```

其中，36 行到 43 行定义了飞船在上下方向上移动的动作，即：不允许 top 小于 0 时继续移动，亦不允许 bottom 超过底边，这样，将飞船的移动空间限制在了窗口内部，随着外部调用 update 方法，飞船的上下位置也可以跟着在限定的

范围内移动。最终，在事件侦听部分，加入对左右按键的监听，适时改变飞船的左右移动的状态：

```
9 def check_key_down(event, ship, alien_setting, screen, bullets):
10     ...
14     if event.key == pygame.K_UP:
15         ship.moving_forward = True
16     if event.key == pygame.K_DOWN:
17         ship.moving_back = True
18     if event.key == pygame.K_SPACE:
19         fire(alien_setting, screen, ship, bullets)
20     if event.key == pygame.K_q:
21         sys.exit()
22
23
24 1 个用法
25 def check_key_up(e, ship):
26     ...
29     if e.key == pygame.K_UP:
30         ship.moving_forward = not True
31     if e.key == pygame.K_DOWN:
32         ship.moving_back = not True
33
```

如图，分别监听了上下键的按下和松开，相应的把飞船的上下移动的允许状态设为了 True 或者 False。

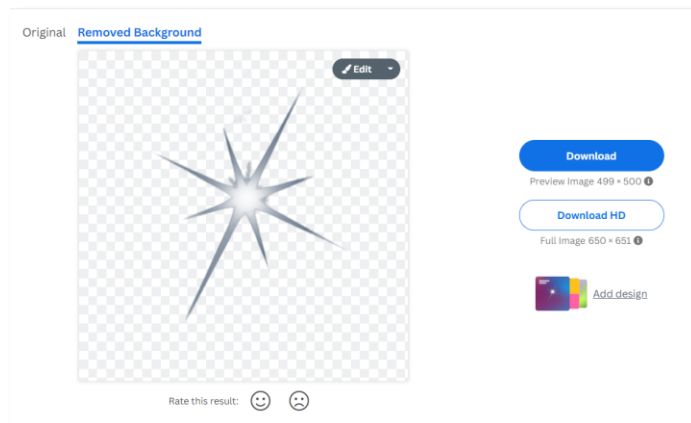
最终飞船移动到窗口中的效果：



如图，飞船击落中间部分的外星飞船，移动到达了顶部。

(2)、找来一副星星的图像，让星星在背景图上随机分布，使得星星看起来更加的逼真。

选用的星星图像：



要在背景当中添加随机的星星，可以仿照添加外星人的思路，只不过坐标是随机生成的。

为了保证每一次刷新画面星星不会乱闪，则只应该在游戏开始的时候便完成星星坐标的生成，在后面的刷新不应该更改位置，所以，可以在 `alien_invasion.py` 里面将星星的坐标预先生成，每一次更改画布的时候传入这组数据即可，这里采用 `numpy` 生成一定数量的星星的坐标，如下：

```
1 个用法
13 def run_game():
14     pygame.init()
15     pygame.font.SysFont("SimHei", 20)
16     alien_settings = settings.Settings()
17     stats = GameStats(alien_settings)
18     screen = pygame.display.set_mode((alien_settings.screen_width, alien_settings.screen_height))
19     coords = np.stack((np.random.uniform(0, screen.get_rect().right, alien_settings.star_num),
20                       np.random.uniform(0, screen.get_rect().bottom, alien_settings.star_num)), axis=1)
21     pygame.display.set_caption("外星人入侵")
22     ship = Ship(screen, alien_settings)
```

其中，`coords` 表示随机生成的坐标，`alien_setting.star_num` 是一个建立在 `settings.py` 里面的类变量，定义了画面当中星星的数量

```
3 用法
1 class Settings:
2     def __init__(self):
3         self.screen_width = 1200
4         self.screen_height = 800
5         self.bg_color = (230, )*3
6         self.ship_speed = 2
7
8         self.bullet = {...}
15
16 self.star_num = 100
17
18 self.alien_speed = 10
```

在 `game_functions.py` 文件里面定义如下函数，用来显示画面当中的星星：

```
1 个用法
188 def draw_stars(surface, coords):
189     import pygame
190     star = pygame.image.load("imgs/star-2.png")
191
192     x_coors = coords[:, 0]
193     y_coors = coords[:, 1]
194
195     for x, y in zip(x_coors, y_coors):
196         scaled_star = pygame.transform.scale(star, (50, 50))
197         surface.blit(scaled_star, (x, y))
198
```

该函数从指定路径读取上方的星星图片，图片经过 ps 处理，添加了黄色描边，在浅色里面看着更加直观，然后将坐标从二维数组当中切片出来，将缩放过后的图片放置在了画面中的相应位置。

接下来，在更新窗口的函数上调用这个函数：

```
1 个用法
68 def update_screen(alien_setting, screen: pygame.Surface, stats, scoreboard, ship, aliens, bullets, play_button, coords):
69     screen.fill(alien_setting.bg_color)
70     draw_stars(screen, coords)
71     for bullet in bullets.sprites():
72         bullet.draw_bullet()
73     ship.blt_me()
74     aliens.draw(screen)
75     scoreboard.show_score()
76     if not stats.game_active:
77         play_button.draw_button()
78     pygame.display.flip()
```

值得注意的是，星星应该作为画面靠近底层的元素来绘制，所有应该尽早绘制，将其放在屏幕填充的后面，如果靠后绘制的话，星星会浮在别的元素之上。最后，只需要在 `alien_invasion.py` 当中修改对 `update_screen` 函数的调用，传入随机生成的坐标即可，最终效果如下：



如图，星星已经成功的加入了画面当中，且位于底层，没有遮挡住其它的元素。

5、对原有代码的一些改进

(1)、使用字典存储相关对象的设置

为了能够更加方便的读取和保存相关对象的设置，比如子弹的速度、宽度、颜色等，使用一个字典进行保存设置，这样避免了 `setting.py` 中 `Setting` 类属性过多、难以找到相关属性的问题，便于开发时记忆和理解相关属性的作用，也便于日后修改代码中属性时能够及时定位。

```
3 用法
1 class Settings:
2     def __init__(self):
3         ...
9
10
11     self.bullet = {
12         'speed': 150,
13         'width': 3,
14         'height': 30,
15         'rgb': (255, 60, 60),
16         'max': 3
17     }
```

(2)、在类中使用单下划线和双下划线的形式保护相关类变量和实例变量，由于 IDE 具有代码提示和补齐的功能，没有对相关类变量设置保护的时候会一并列出且部分属性的变量名较为相似，容易误操作使用了错误的属性，造成程序运行的错误和排除问题的困难。使用类保护属性可以避免无意之间对某一重要类变量的修改，提高开发效率。

```
5 用法
5 class Ship(Sprite):
6     def __init__(self, screen: pygame.Surface, alien_settings):
7         super(Ship, self).__init__()
8         self.__screen = screen
9
10         self.__img = pygame.image.load("./imgs/ship-removebg-preview.png")
11         self.image = self.__img
12         self.rect = self.__img.get_rect()
13         self.__screen__rect = self.__screen.get_rect()
14
15         self.rect.centerx = self.__screen__rect.centerx
16         self.rect.bottom = self.__screen__rect.bottom
17         self.__alien_settings = alien_settings
18         self.center = float(self.rect.centerx)
19         self.moving_right = False
20         self.moving_left = False
21
22         self.moving_forward, self.moving_back = False, not True
```

(3)、使用 png 格式的图片

原有的 bmp 位图格式的图片导致了图像的外面始终会有一个白色的边框，如果屏幕背景色被设置的很深，那么这个矩形框便会显得很碍眼，而有的时候却有使用深色背景色的区别，比如，我们可以使用 `rgb(6,67,168)` 作为背景色，该颜色在蓝色分量上值较大，同时红色、绿色分量不是很大，为深蓝色接近星空的颜色，用作背景色较为合适，这个时候若将原有位图转换为 png 格式再掏空周边，界面上将会更加美观。

同时，需要考虑到使用深色背景色的时候，字体颜色应当相应的变浅色，可以将字体色在 `settings` 类当中计算好，记分牌部分引用当前字体颜色即可。具体的计算字体颜色，可以使用白色减去背景色即可，这里可以使用 `numpy` 进行，用标量 255 减去一维的背景色向量即可得到字体的颜色，如下，我们只需要设置背景色为星空色，将星星的图片换成白色：

```
3 用法
1 class Settings:
2     def __init__(self):
3         import numpy
4         self.screen_width = 1200
5         self.screen_height = 800
6         self.bg_color = (6, 67, 168)
7         self.font_color = tuple(255-numpy.array(self.bg_color))
8         self.ship_speed = 6
```

如上图，其中定义了背景色为深蓝色，字体颜色为纯白色减去背景色。最终的效果如下：



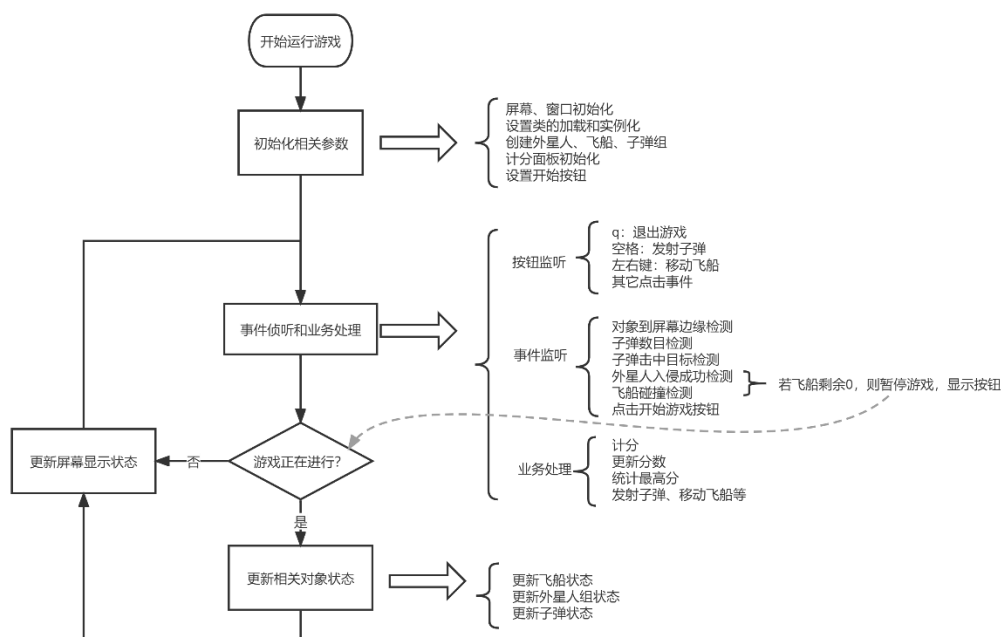
五、实验结论

1、安装 pygame 可以使用命令 **pip install pygame**, 这个命令同样适用于安装其它的库, 若由于相关原因导致正常下载太慢或者失败, 可以使用清华大学开源软件镜像站来进行下载。

2、为了便于维护和管理, python 项目的文件一般组织在一个文件夹里面。

3、python 项目的代码开发时会经过多次的迭代和修改, 若不能在开发的时候进行合理的规划, 每一次的修改和迭代可能是十分的费时费事的, 因为大多数修改并非单独的, 它们之间存在各个地方的调用和值传递, 往往是牵一发而动全身的修改。例如在本项目当中, 一次又一次的修改相关函数的参数表, 将本身只有两三个参数的函数最终修改成了八九个参数, 而且往往修改一个函数, 它的调用者及其调用者的调用者的参数表都需要修改, 这一点是十分的让人恶心的, 如果能在开发之初规划好这些函数实现的功能, 便无需这般大费周章。

4、本游戏项目《外星人入侵》的大致逻辑流程如下:



5、通过对项目的修改、测试以及一些属性的自定义, 更加深刻的掌握了 python 对象的属性及常用的 python 语法和特性, 更加熟练的掌握了常见库如 pygame 库和 numpy 库的应用。