

《Python程序设计基础》程序设计作品说明书

题目： 外星人入侵游戏

学院： 21计科

姓名： 覃思敏

学号： B20210302317

指导教师： 周景

起止日期： 2023.11.10-2023.12.10

摘要

本项目是一款名为“外星人入侵游戏”的基于Python编程语言的程序设计作品。该游戏以经典的射击游戏为基础，融合了创新的外星人入侵元素，旨在为玩家提供一种富有挑战性和娱乐性的游戏体验。

游戏的主要特点包括：

- 游戏背景与故事情节：** 通过引人入胜的故事情节，玩家将沉浸在一个虚构的外星世界中。外星人入侵地球的背景设定为游戏的核心，激发玩家的兴趣和好奇心。
- 玩家控制：** 游戏允许玩家操控一个太空飞船，以抵御不断涌来的外星人入侵。通过键盘或鼠标操作，玩家可以实现飞船的移动、射击和其他操作。
- 外星人入侵：** 游戏设计了多种类型的外星人，每一种都具有独特的特性和攻击方式。玩家需要巧妙地规避外星人的攻击，并运用武器摧毁它们，以保卫地球。
- 游戏难度升级：** 随着游戏的进行，外星人的数量和难度会逐渐增加，使得游戏变得更加具有挑战性。玩家需要不断提升自己的操作技能和战略思维，以在激烈的战斗中取得胜利。
- 得分系统和排名：** 游戏中设置了得分系统，根据玩家摧毁外星人的数量和效率来计算得分。玩家的得分将被记录并显示在排名榜上，激发了玩家之间的竞争与合作。

通过本项目，学习者将深入理解Python编程语言在游戏开发中的应用，包括图形界面设计、事件处理、游戏逻辑实现等方面。此外，项目还注重培养学习者的问题解决能力和创新思维，为其提供一个全面而有趣的编程实践经验。

关键词： Python， 游戏， 外星人， 射击游戏

1. 第1章 需求分析

1.1 引言

本章旨在对《Python程序设计基础》中的外星人入侵游戏项目进行需求分析。通过明确定义系统的功能和目标，为后续的分析与设计提供清晰的指导。

1.2 项目背景

外星人入侵游戏是一款基于Python编程语言的2D射击游戏。项目旨在通过游戏开发实践，帮助学习者深入理解Python编程的基础知识，并锻炼问题解决和创新思维能力。

1.3 系统功能需求

1. **用户界面**: 提供直观、友好的用户界面, 包括主菜单、游戏界面和得分板等。
2. **操作方式**: 支持键盘和鼠标操作, 确保玩家可以轻松控制太空飞船的移动和射击。
3. **外星人入侵**: 实现多种外星人类型, 具备随机生成、移动和攻击的功能, 增加游戏的复杂性。
4. **武器系统**: 设计多种武器, 包括基础武器和升级武器, 使玩家能够逐步提升战斗力。
5. **游戏难度**: 提供可调整的游戏难度设置, 以满足不同玩家的挑战需求。
6. **音效与视觉效果**: 集成引人入胜的音效和视觉效果, 提升玩家的游戏体验。
7. **得分系统与排名**: 实现准确的得分记录系统, 以及展示玩家排名和成绩的功能, 激发竞争和合作。
8. **故事情节**: 制定引人入胜的游戏故事情节, 增强玩家对游戏世界的沉浸感。

1.4 用户问题解决

- **学习Python编程**: 通过游戏项目, 学习者能够应用Python语言解决实际问题, 提高编程技能。
- **培养创新思维**: 项目要求学习者创造性地解决游戏开发中的问题, 培养创新和解决问题的能力。
- **提升团队协作**: 如果项目以团队形式完成, 学习者将有机会提升协作和沟通技能。

第2章 分析与设计

2.1 系统架构

游戏系统分为前端和后端两个主要模块。前端负责用户界面和游戏逻辑的展示, 后端处理游戏数据的存储和管理。

2.2 系统流程

1. **游戏启动**: 用户启动游戏应用, 系统加载游戏资源并显示主菜单。
2. **游戏进行**: 玩家选择难度等级后, 进入游戏界面, 与外星人进行战斗。
3. **游戏结束**: 游戏结束后, 系统展示得分和排名, 并提供重新开始或退出游戏的选项。

2.3 系统模块

1. **用户界面模块**: 负责显示主菜单、游戏界面和得分板等用户界面。
2. **游戏逻辑模块**: 包括外星人入侵机制、武器系统、难度设置等游戏核心逻辑。
3. **数据管理模块**: 负责游戏数据的存储、读取和排名管理。

2.4 数据库设计

由于本游戏的数据规模相对较小且主要包含玩家的最高分信息, 因此选择使用简单的文本文件而非传统的数据库来存储数据。

1. **数据存储结构** 为了保存最高分信息, 将采用简单的json文件格式。文件的第一行就是目前的最高分。
2. **数据访问方式** 通过Python的文件读写操作实现对json文件的访问。使用适当的文件读写方法来更新和检索最高分信息。
3. **数据库操作** (1) 读取最高分: 游戏启动时, 系统读取json文件以获取当前的最高分信息。 (2) 更新最高分: 当玩家在游戏中获得新的最高分时, 系统将更新json文件中对应的玩家分数

2.5 关键实现

1. **外星人入侵算法**: 设计合适的算法确保外星人的生成、移动和攻击具有足够的随机性和挑战性。

2. **武器系统实现**: 制定系统, 使得武器的选择和升级能够影响游戏进程。
3. **用户界面交互**: 实现用户界面与游戏逻辑的良好交互, 确保玩家能够顺利进行游戏。
4. **得分记录与排名**: 设计系统确保准确记录玩家得分, 并在需要时展示排名信息。

(1) 数据持久化 Python的json模块提供了对json文件的基本操作, 包括读取、写入、更新和检索。为了保留玩家的最高分记录, 需要实现数据持久化。将最高分保存到json文件中, 以便后续读取和更新最高分。关键代码如下:

```
def check_high_score(self):
    """检查是否诞生了新的最高分"""
    if self.stats.score>self.stats.high_score:
        self.stats.high_score=self.stats.score
        self.prep_high_score()
def write_high_score(self):
    """将最高分写入文件"""
    filename='hightest_score.json'
    with open(filename,'w') as file:
        json.dump(self.stats.high_score,file)
```

(2) 用户输入与控制 通过键盘和鼠标实现宇航员的移动和射击, 为玩家提供直观的游戏操控体验。确保对用户输入的实时响应, 以及相应事件的正确处理。部分关键代码如下:

```
def _check_keydown_events(self,event):
    """响应按下"""
    if event.key==pygame.K_RIGHT:#玩家按下向右键时moving_right为True
        #向右移动飞船
        self.ship.moving_right=True
    elif event.key==pygame.K_LEFT:
        #向左移动
        self.ship.moving_left=True
    elif event.key==pygame.K_q:#按Q键退出
        self.sb.write_high_score()
        sys.exit()
    elif event.key==pygame.K_SPACE:#按空格键发射一颗子弹
        self._fire_bullet()
    elif event.key==pygame.K_UP:
        self.ship.moving_up=True
    elif event.key==pygame.K_DOWN:
        self.ship.moving_down=True
```

(3) 碰撞检测算法 在外星人入侵游戏中, 碰撞检测算法是确保游戏元素交互正确的关键部分。当宇航员的位置与外星人的位置重叠时, 触发碰撞检测; 当子弹的位置与外星人的位置重叠时, 触发碰撞检测; 防止游戏元素越界, 即超出游戏界面的边界。部分关键代码如下:

```
def _check_bullet_alien_collisions(self):
    """响应子弹和外星人的碰撞"""
    #删除发生碰撞的子弹和外星人
```

```
#检查是否有子弹击中了外星人
#如果是，就删除相应的子弹和外星人
collisions=pygame.sprite.groupcollide(self.bullets,self.aliens,True,True)
if collisions:
    for aliens in collisions.values():
        self.stats.score += self.settings.alien_points*len(aliens)
        self.sb.prep_score()
        self.sb.check_high_score()
if not self.aliens:
    #删除现有的子弹并创建一个新的外星舰队
    self.bullets.empty()
    self._create_fleet()
    self.settings.increase_speed()

    #提高等级，整个外星舰队都被击落，就提高一个等级
    self.stats.level += 1
    self.sb.prep_level()
```

第3章 软件测试

1. 测试AlienInvasion类的check_keydown_events方法

#	测试目标	输入	预期结果	测试结果
1	按下向 右键	ai_game 为 AlienInvasion 实例，模拟 KEYDOWN 事件，按下右箭头键	ai_game.ship.moving_right 属性为 True	通过
2	按下向 左键	ai_game 为 AlienInvasion 实例，模拟 KEYDOWN 事件，按下左箭头键	ai_game.ship.moving_left 属性为 True	通过

2. 测试AlienInvasion类的check_keyup_events方法

#	测试目标	输入	预期结果	测试结果
1	松开向 右键	ai_game 为 AlienInvasion 实例，模拟 KEYUP 事件，松开右箭头键	ai_game.ship.moving_right 属性为False	通过
2	松开向 左键	ai_game 为 AlienInvasion 实例，模拟 KEYUP 事件，松开左箭头键	ai_game.ship.moving_left 属性为 False	通过

3. 测试button类

#	测试目标	输入	预期结果	测试结果
1	测试按键的 创建	msg="Play", expected=(0, 135, 0)	Button 对象被正确创建，msg_image 不为 None，按钮属性正确设置	通过

#	测试目标	输入	预期结果	测试结果
2	测试按键的可视化	msg="Play", color=(255, 0, 0), font_size=48	按钮的颜色、字体大小被正确设置	通过

结论

本项目成功地实现了一个基于Python的外星人入侵游戏，提供了一系列有趣的游戏玩法和挑战。玩家通过键盘或鼠标控制宇航员的移动，实现上、下、左、右的方向控制，以及通过按键或点击屏幕进行射击。同时，实现了计分模块，记录玩家的得分，并将最高分保存到文件。并且随着时间的推移，外星人的速度逐渐增加，提高游戏难度，为玩家提供更大的挑战。与此同时，还实现了随机生成随机个数的星星散落到屏幕中。但是，这个项目还有存在着一些不足之处，例如，缺少音效和动画元素，这可能影响游戏的沉浸感；图形和界面相对简单。改进方法：使用图形库（如Pygame、PyQt等）升级游戏的图形和界面，提升视觉效果。添加音效和动画元素，增强游戏的沉浸感。进行更全面的测试，包括性能测试，以确保游戏在不同环境下都能稳定运行。

参考文献

[1] 《Python教程：从入门到实践（第三版）》 [2] 《Python游戏编程入门：零基础玩转Python游戏开发》 [3] 《Python 游戏编程入门》