计算机程序设计课程实践项目报告

Python 语言版 voidKnight

完成人：　　胡宇凡　**PB20051141**

（主循环、Player 类、寻路机制、随机地图）

张比豪　**PB20051119**

（Enemy 类、敌人生成、贴图、音效）

完成时间：　**2020 年 12 月 28 日**

摘　　要

本项目：

voidKnight 是基于 Python3.7 开发环境开发出来的第三视角平面打斗游戏，主要使用 pygame 模块实现图形化界面和人机交互。

关键词：

voidKnight

Python

pygame

目　　录

1. 项目概述

1.1 项目特点

voidKnight 是基于 Python3.7 开发环境开发出来的第三视角平面打斗游戏，主要使用 pygame 模块实现图形化界面和人机交互。游戏简单易于操作，用户界面友好，人机交互方便。该小游戏系统稳定性好，逻辑思维紧凑而细腻。可以很好的锻练编程能力。

1.2 项目设计

在对 C 语言的学习，使得不少人觉得 C 语言枯燥无味，没有实用价值，继而没有深入学习的动力。而 Python 语言则很好地解决了这个问题，有趣的 voidKnight 游戏充分体现了其价值。相较于 C 语言，更加友好的 Python 语言丰富而不枯燥，而且因其基于 C 语言，在实践中常常出现 C 语言的身影（比如幽灵般的指针），帮助复习 C 语言相关知识。

1.2.1 功能分析

a. 单机游戏
b. 难度梯度
c. 画面美工
d. 游戏音效

1.2.2 性能分析

a. 游戏画面流畅无卡顿，稳定 30fps（在极限的测试情况下，我们发现计算压力主要来自于图形界面的绘制）
b. 具有吸引力和耐玩性（足够吸引组外测试人员帮助测试）
c. 开源项目，完全免费

2. 系统设计

2.1 系统总体设计

小游戏有简单的菜单选项，用来选择相应的功能，包括：
（1） 挑战模式
（2） 无尽模式
（3） 游戏说明

在游戏中应体现的功能包括：
（1） 游戏的暂停和开始
（2） 游戏的得分记录和得分的实时更新
（3） 退出游戏

2.2 主要数据结构

定义类与函数，如
类：
Box：记录矩形框位置信息，负责碰撞检测
Platform：平台类，存储平面图信息，提供寻路基础
Player：玩家类，根据键盘输入更新玩家状态
Enemy：敌人类，负责敌人的寻路
函数：
main：主循环

2.3 模块设计

该项目包含主函数、碰撞箱模块、随机地图模块、寻路模块、生怪模块等模块结构。

2.3.1 主函数

该模块（也即游戏逻辑模块）是程序入口，承担游戏主循环。管理菜单界面、游戏循环函数、暂停界面间的切换，负责检测并传递玩家的输入事件，以及界面绘制与音效控制。

2.3.2 碰撞箱模块

该模块存储玩家与敌人的位置信息，用于贴图的绘制，同时负责检测矩形碰撞，用于判定碰撞伤害与攻击伤害。

2.3.3 随机地图模块

该模块的功能是：在开始游戏时随机生成地图，并计算连接各个平台的寻线路径。

2.3.4 寻路模块

该模块基于寻线路径、敌人所在位置与玩家所在位置计算敌人的运动，并作为虚拟输入控制敌人移动。承担敌人的 AI。

2.3.5 生怪模块

该模块主要功能是生成敌人，根据不同游戏模式会有不同生成机制。

2.4 开发语言和开发工具

本系统采用 Python 语言作为开发语言，Python 语言的主要特点如下：

(1) 语言简洁、紧凑，使用方便。

(2) 支持类与对象。

(3) 数据类型丰富，限制少，使用自由，且类型转换比较智能。

(4) 语法限制不太严格，程序设计自由度大。

(5) 用 Python 语言编写的程序可移植性比较好，能运用于装有 Python 解释器的各种型号的操作系统中。

(6) Python 语言允许通过“列表”间接访问物理地址，方便不同模块间数据传输。虽然这一定程度上破坏了模块化，降低了可读性，但大大优化了系统性能。

因为 Python 语言瑕不掩瑜，所以本系统使用 Python 语言作为开发语言，简化了编写过程，优化了游戏界面。

使用了 pygame 库。该库提供如下功能：

(1) 图形界面的支持

(2) 键盘鼠标输入事件

(3) 播放音效

3. 系统测试

3.1 系统运行

游戏控制方法：详见 https://github.com/Hyffer/voidKnight。

项目编写过程中有邀请多个非项目设计者进行体验，我们根据其反馈不断修改，最终其反映体验良好（甚至有点玩上头）。

3.2 系统测试

(1) 游戏的每个版本都经由作者和他人进行多次测试后才发布更新。

(2) 编写时考虑所有输入情况，包括最极端情况，因此不会出现因操作不当而崩溃的情况，且输出数据均在预期范围内。

(3) 代码经常优化，以提高可读性，避免潜在隐患。

(4) 定期对单个文件进行测试。

4. 结论和体会

经过以上各章节的功能，已经基本完成了 voidKnight 游戏的开发，voidKnight 是一个趣味性和逻辑性很强的游戏。游戏从简单的菜单选项界面、死亡界面，到屏幕绘制、事件检测、游戏逻辑，再到随机地图、怪物寻路，逐步由简到难，锻练编程能力。游戏不仅有灵活的交互功能，还具备界面友好、操作简单和趣味性强的性质。界面美观友好、多彩多姿，完全的图形化设计，操作者易于上手，同时多种媒体技术的集成利用，可以方便地完成用户乐于接受的各种界面设计。本系统虽具备了基本的功能，但由于时间关系，还有很多功能待以实现，在此基础上结合实际游玩中所出现的问题，加入更多游戏模式与玩法，后续将不定期更新。

通过这次编程我深深的感受到代码的变量命名、代码内注释格式、类与方法的封装的重要性。良好的编写习惯、定期的代码优化，不但有助于代码的移植和纠错，也有助于不同人员之间的协作。我们还要有模块化思维能力，模块化思维就是编程任何一个功能模块或函数的时候，要多想一些，不要局限在完成当前任务的简单思路上，想想看该模块是否可以脱离这个系统存在，是否可以通过简单的修改参数的方式在其他系统和应用环境下直接引用，这样就能极大避免重复性的开发工作。善于总结，也是学习能力的一种体现，每次完成一个编程任务，完成一段代码，都应当有目的地跟踪该程序的应用状况，随时总结，找到自己的不足，这样所编写的程序才能逐步提高。

在此列出不足之处，提醒我们改进：

1. 一心向前推进，缺少回头修改、优化的过程，在后期的进展中造成不少麻烦。
2. 缺乏模块思维，类的封装不完善，有较多单独的语句，造成文件之间关联关系较乱。随着项目规模逐步扩大，其弊端愈发凸显，造成代码修改困难，也限制了项目的进一步发展。

5. 参考文献

1. pygame 官方文档：https://www.pygame.org/docs
2. Making Games with Python & Pygame - By Al Sweigart