

植物大战僵尸PVZ 第二阶段

植物大战僵尸PVZ 第二阶段

课程设计分析

课程设计目标

课程设计基础内容

设计分析

花园

植物

僵尸

核心类的设计展示

程序的亮点

显示清晰

难度合适

程序的运行方法

遇到的问题与困难

类的多态的实现

解决了商店调用类似于轮循的问题

游戏结束的优化

可供改进之处

课程设计分析

课程设计目标

在第一次课设的基础上，实现植物、僵尸、商店、显示面板的一系列拓展。

课程设计基础内容

- 对于植物，增加以下新种类：
 - 攻击性 (Shooter): 双发射手、寒冰射手
 - 防御性 (Defense) : 坚果墙、高坚果
 - 炸弹型 (Bomb) : 窝瓜, 樱桃炸弹
 - 效果型 (Efficient) : 大蒜, 南瓜头
- 对于僵尸，增加以下新种类：
 - 路障僵尸, 读报僵尸, 撑杆僵尸, 小丑僵尸, 投石僵尸
- 商店设计，增加以下新功能：
 - 购买有冷却时间
- 界面设计，增加以下新功能：
 - 一个地块可以有多个僵尸
 - 显示植物和僵尸的生命值

设计分析

基于第一次设计，利用已有的类和函数，利用继承的方式做拓展；对于有类似需求但是实现不同的类（如植物所具有的攻击僵尸的功能根据植物种类的不同而有不同），考虑用多态。

花园

对花园框架进行了修改，之前的花园一格对应一个对象的指针，不能容纳多个对象。现在把garden_pos改成了每行对应一个vector，需要获取行列信息时通过遍历vector来完成。另外设置一个二维的vector组garden_pos_cnt，用来记录每个格子内对象的数量，为了保证显示清晰，上限是每个格子最多能打印字符串的行数。

（耗时大约3h）

植物

和之前相比，这里的植物类有了更加细致的划分，如shooter就又分为了普通射手、寒冰射手、双发射手，为了在攻击时对这些细分植物进行区分，同时又保证在状态更新时可以根据共性进行统一处理，在Plant类内新开辟了func_type，以便于和plant_name区分。func_type包括射手类(Shooter)，炸弹类(Bomb) 防御类(Defense) 效果型(Efficient)等。例如，对于双发射手，其func_type是Shooter，plant_name是Double Shooter。

对于植物的附加功能，例如寒冰射手和普通射手的区别，一开始希望采用在子弹基础上继承得到新的子弹来完成，但是使用的迭代器对指针的类型有限制，因此无法区分普通子弹和带有冷冻减速作用的寒冰子弹。于是考虑将这一判断交给花园，当花园判断出子弹来自寒冰射手的时候，另外调用减缓僵尸前进速度的函数作用于僵尸。

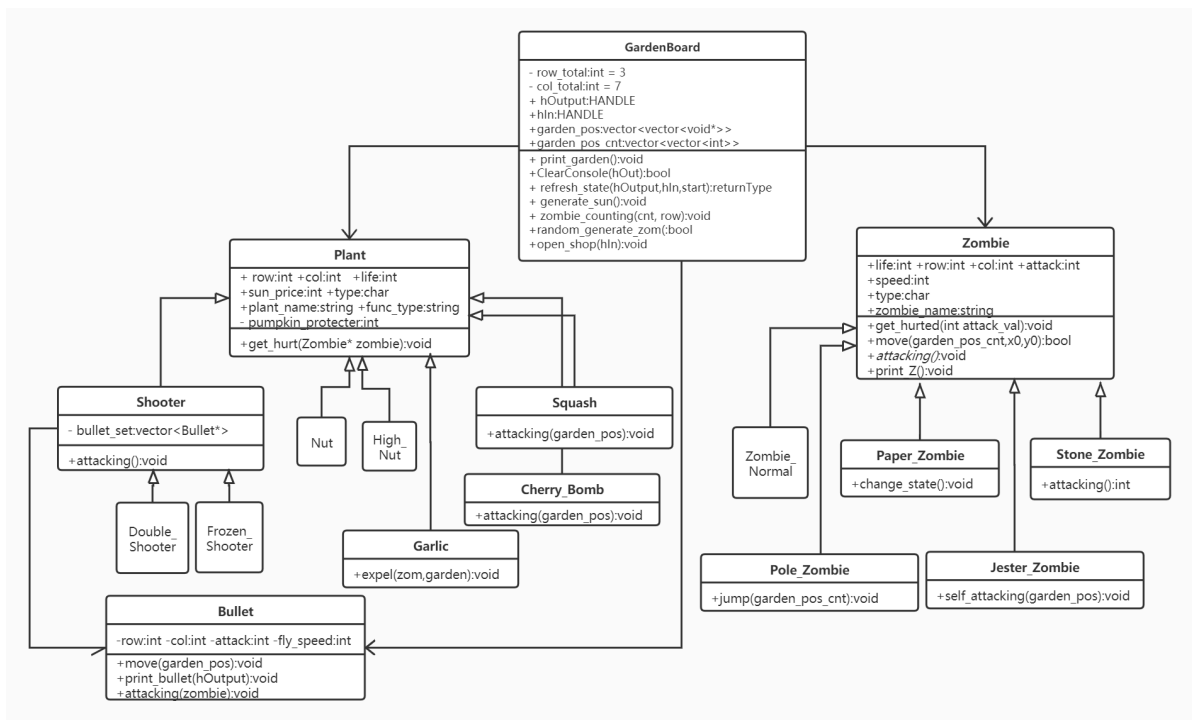
在考虑添加功能的方式耗时2h左右，实际上由于第一次课设对游戏内部对象的交互是基于花园来传递信息而非对象之间独立判断彼此，因此的确改成现有方案更加可行。

僵尸

一开始考虑从zombie类直接继承，发现无法有效判断僵尸的种类，于是添加了新的父类Zombie_Normal，在Zombie_Normal中添加zombie_name来区分。在花园中对僵尸类型做判断来实现特殊的功能。

核心类的设计展示

采用uml图来展示：



程序的亮点

显示清晰

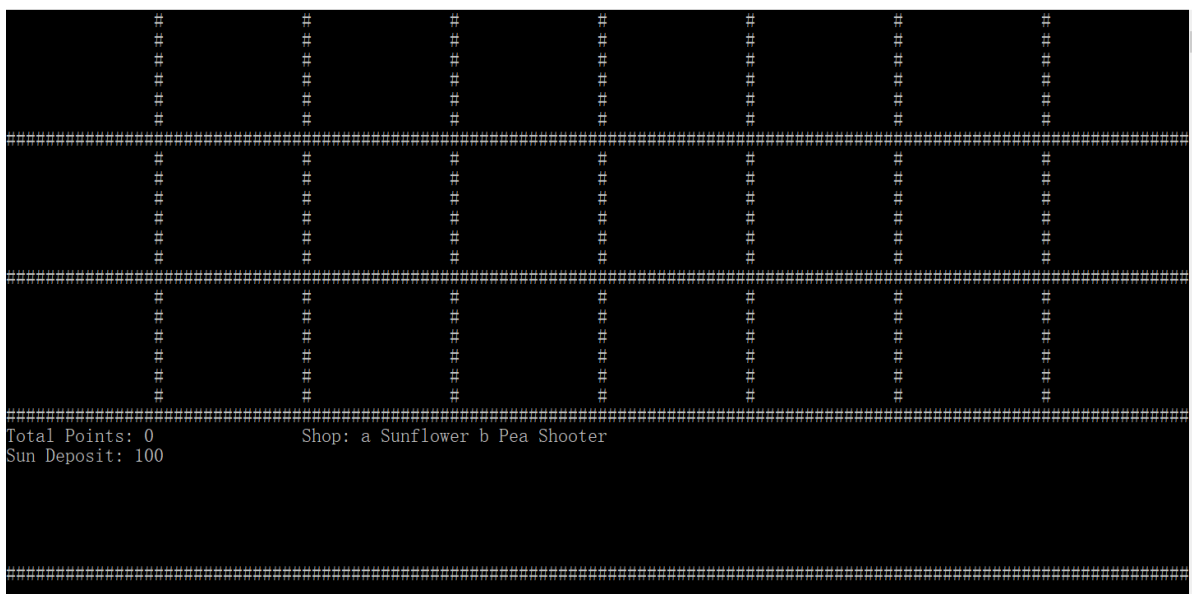
同一地块内僵尸不会重叠，会在不同行显示

难度合适

经过多次调试，调整了僵尸运行的速度，普通、加速、减速的僵尸有明显的区别，让玩家能够充分享受游戏

程序的运行方法

打开exe或者在项目按下ctrl+F5



双击回车进入初始化的花园：

```
#####
#               #               #               #               #
#       Shooter       #               #               #               #
#               #               #               #               #
#####
#               #               #               #               #
#               #       Shooter       #               #               #
#               #               #               #               #
#####
#               #               #               #               #
#       Shooter       #               #               #               #
#               #               #               #               #
#####
Total Points: 0           Shop: a Sunflower b Pea Shooter
Sun Deposit: 100
#####
```

需要商店：单击鼠标，按下键盘b（或者双击键盘b），根据展示的shop；按下键盘上对应的字母选择植物，然后用鼠标点击希望种植的地块，单击选择，按下回车键确认；如果第一次点错了，在按下回车键前都可以通过双击新位置来更改位置。

```
#####
# Sunflower  Shooter  *      *      *Zombie  Zombie  Zombie
#               #               #               #               #
#####
# Sunflower      Shooter  *Zombie died!
#               #               #               #               #
#####
#       Shooter
#               #               #               #               #
#####
Total Points: 40           Shop: a Sunflower b Pea Shooter
Sun Deposit: 300

open shop

Use keyboard to choose one:
```

遇到的问题与困难

类的多态的实现

课堂上介绍多态是基于指针已经明确是基类/派生类的情况，然而在实现中，最难的部分是如何确定此处的指针是哪个类的指针。

解决方案是设置一个统一的基类，在基类中添加派生类中必定会继承的标签，花园根据标签对植物和僵尸做强制类型转换。

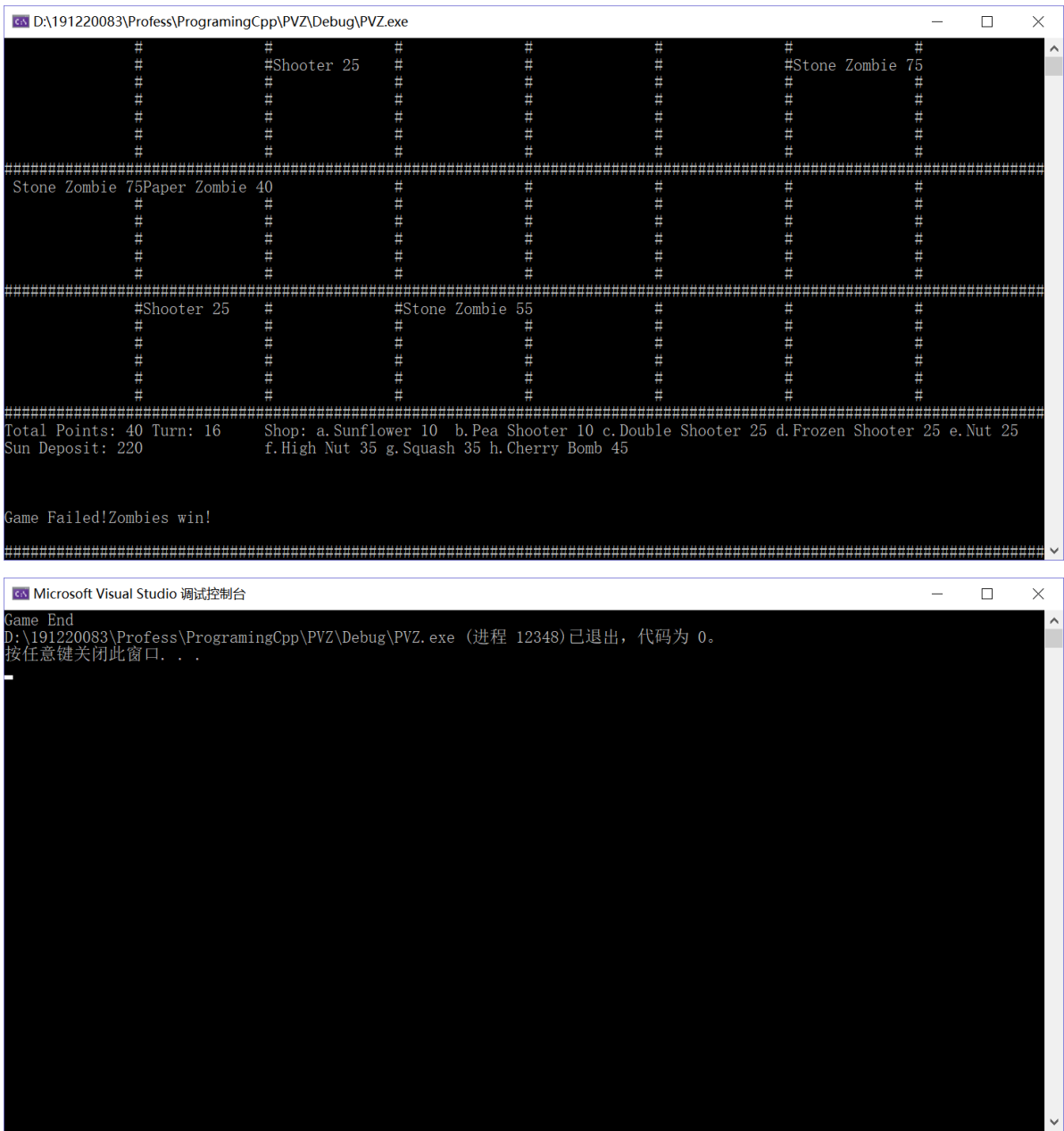
解决了商店调用类似于轮循的问题

通过在缓冲区保留来避免因为读取不到就block的情况，同时在判断结束后及时清空缓冲区避免影响下一回合的判断

<https://intfiction.org/t/windows-console-non-blocking-mode/43154/5>

游戏结束的优化

之前游戏结束是一闪而过，现在通过调整显示和游戏结束判断的顺序，以及添加交互，让用户能够看到游戏结束时的花园状态，并通过任意键盘的按下来彻底结束游戏。



可供改进之处

目前应用仍然基于控制台，可以改进为更加友好漂亮的图形化界面

目前进入商店就无法退出，必须购买，可以改进成有退出效果

debug的时间比上一次长，主要卡在对迭代器不是很熟练，没有达到预期的功能的效果所以重新构思了；代码规模变大，有一些分支调整的时候有疏漏；而且对新增的植物/僵尸做功能支持的时候容易混乱，更改的不完全容易出现不一致的情况

