

植物大战僵尸第二阶段

201220061 陈冠杰

1、这一阶段我实现了什么

1-1整个项目推翻重构

由于上个阶段重心放在学习使用QT软件和多线程的工作原理上，所以在数据结构和子类基类设计上严重不足，虽然还原了简易的植物大战僵尸，但是，游戏非常卡顿，难以debug，并且重复代码极多，复用性差。此外，上一阶段，由于不熟悉多线程，所以建立了一大堆难以管理的时钟，导致程序线程非常紊乱，有概率在游戏中触发程序崩溃。此外，上一阶段由于不了解parent在QT中的作用，所以内存管理非常失败，这一次重构后，几乎可以保证没有内存泄漏。

1-2推翻重构具体做了什么

1-2-1 实现植物和僵尸以及子弹的基类，并通过继承完成全部子类

由于基类完成大部分重复工作，实现一个子类的植物只需要二十行。

1-2-2 借助基类指针，进一步完善了草坪的中介功能，草坪同时存储了存储的植物，草坪的状态，以及位于草坪上的僵尸的链表

```
// 简单展示一下草坪的类
class grass : public QWidget
{
    Q_OBJECT
public:
    //略去部分不重要的成员函数
    static bool line_has_zom(int);
    bool In(int x,int y);
    bool Anchor(Plants type);//草坪和植物的链接
    //以下是僵尸和草坪的链接，植物可以间接和僵尸通信
    void anchor_zombie(zombies* newzom);
    void delete_zombie(zombies* newzom);
    bool hit_first(int power);
    bool hit_all(int power);
    bool hit_all_G(int power);
    int firs_zom_x();
    bool has_zom();
signals:
    void cleared();
public slots:
private:
    Plant* plant;//存储植物的指针
    QLinkedList<zombies*> zoms;//存储僵尸的链表
    //略去无关紧要的成员
};
```

1-2-3 优化数据结构，统一管理帧率，降低各算法的复杂度，优化程序的运行效果

最显著的是对子弹攻击僵尸的算法和数据结构的优化，具体细节可以参看Bullet.cpp和Bullet.h以及grass类来了解。简单介绍一下，就是讲僵尸存储在每个草坪的僵尸指针的链表中，子弹每次可以选择攻击自己所在的草坪中的第一个僵尸，或者攻击整个草坪上的僵尸。僵尸每次移动都会更新自己新进入的草坪和离开了的草坪。原始的算法则是将全部僵尸存储在一个vector中，每次子弹试图攻击时都要历遍全部僵尸，并且僵尸的删改增添也非常耗时。

经此优化，子弹和僵尸的攻击算法复杂度由 $O(n*m)$ 降低为 $O(n)$ ，在假设僵尸有一百只时，耗时减少百倍以上。因此，游戏可以支持更高的帧率，并且运行时也流畅许多。

1-2-4 精简时钟，降低多线程的复杂度，提高程序稳定性

本次实验中，每个类都有且仅有一个时钟，时钟以全局变量FPS（帧率）为间隔时间，稳定发出信号，每个动作通过FPS_gap来进一步触发，FPS_gap每个时钟信号来临时加一，达到一定值时再触发某个动作，FPS_gap再置为0，重新开始计时，由此实现了时钟的统一，因此也降低了多线程的复杂度，不再会出现前一阶段那样运行到某一时刻程序异常终止的情况。

1-3 拓展功能

1-3-1 用户自主导入地图，僵尸灵活改变行进方向

由于PVZ本身的限制，直接在地形上改变游戏并不现实，也不有趣，所以我另外构想，用植物本身作为地形上的障碍，僵尸遇到植物时，第一选择不是吃植物，而是上下看一眼，如果上方无阻碍，则向上移动，绕过当前植物，如果都有植物的话才会选择吃植物。在这种设计下，游戏的策略性就更高了，游戏也更有意思了，也实现了用户对于地图的自主改变。这个过程其实并不容易，因为之前重构时改变了数据结构，所以为了保持对游戏的优化，需要在此花费很多功夫。

1-3-2 增加飞行单位：飞行僵尸

飞行僵尸可以躲避来自地面的伤害，比如不会触发土豆地雷（土豆地雷尚未有时间实现），免疫地刺伤害，此外，飞行僵尸也可以直接越过植物（高坚果除外，但尚未实现），另外，为了有趣，飞行僵尸还可以向身下喷火（另外有Fire.gif），边走边攻击植物。

具体形象如下：



1-3-3 进一步增加植物的种类

本阶段中，由于重构和实现僵尸的移动花费了大量时间，所以新拓展的植物只有三个

Repeater:



CherryBomb:



FireTree:



其具体功能与游戏本身一致，在此不再赘述。

另外，之前已经实现的植物/工具如下：

SunFlower:



NutWall:



PeaShooter:



SpikeRock:



Shovel:



Car:



其具体功能也同游戏一样，不在赘述。

1-4 细节优化

射手类植物只会在当前所在列有僵尸的情况下进行攻击动作。

给地刺加上了攻击动作。

安插植物时如果阳光不足，会在鼠标点击位置暂时显示“阳光不足”的label。

resize了整个游戏以及游戏中各个物件的大小、比例，使得游戏画面更加可观。

调高帧率至30帧，运行更加流畅，并且可以进一步调高。

火炬树桩可以改变豌豆粒的威力和图像等。

1-5 删除部分内容

决定将小车移出植物栏，因为在这次重构之后，小车继承的是Bullet的基类，不再享有植物的属性。

删掉了开头场景和地图选择场景，那两个属实是累赘，是当时学QT时练习的产物。删了之后画面反而更加简洁。

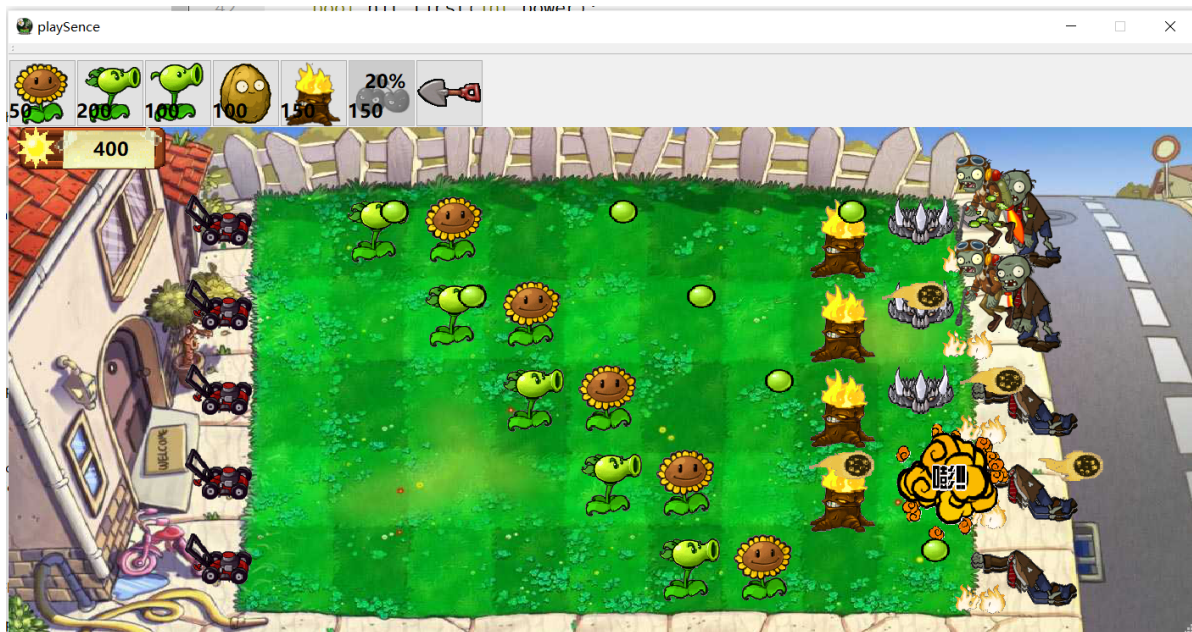
删去了“游戏失败”的画面，太丑了主要是。。。之后有空会考虑重新设计。

删去了“游戏胜利”的画面，理由同上。。。或许因为我想要把它变成无限版？

删去了旧式尸潮（短时间创建100或200个僵尸，觉得不咋有趣），主要是因为重构之后，僵尸的内存可以自动管理，创建尸潮更加容易，这个版本中，尸潮的形式将会更加丰富有趣。

2 实现的效果如何

直接上图：（僵尸的移动不方便图片展示，待线下验收再展示）



3.下一阶段我会做什么

3-1 提供“内挂”

比如能量豆，使得短时间内某植物能力翻十倍？

在最上方的按钮栏中增加形式丰富的挂，比如点击后在全场的空草坪上添加坚果墙、樱桃炸弹等。

自主调节游戏速度。

等等

3-2 增加植物

初步设想：寒冰射手，高坚果，土豆地雷，南瓜保护罩，食人花，火爆辣椒。

由于这次重构更改了子弹和僵尸的数据结构，所以使得胡杨的实现成为可能。

3-3 增加僵尸

初步设想：读报僵尸，铁桶僵尸，路障僵尸，舞王僵尸，自爆僵尸，跳高僵尸

3-4 游戏失败画面和胜利画面

其实不怎么想做这两个来着。。。