基于 QT 实现的植物大战僵尸游戏代码报告

游戏基本介绍

植物大战僵尸是一款游戏非常经典的益智类游戏，画风友好，玩法多样，整体逻辑比较清晰。本次我选择使用 c++的 GUI 库 Qt 来写复现游戏。总体上来说完成了游戏加载界面，菜单选择界面，主游戏逻辑和交互界面和游戏的基本功能。我本人是一个鬼畜爱好者，平时会去做一些鬼畜，出于趣味性，我将鬼畜元素融入了我的游戏，但由于时间等方面的限制，本游戏实现了了七种鬼畜僵尸和九种鬼畜植物，以及白天和黑夜两个场景模式，具有了一定的可玩性。

基于自己的水平考虑，我选择了非常常规的经典面向对象的思路，定义良好的复用性强的基类，如果想要添加新的对象和功能，只需要从基类派生并将新类加入生成器。在逻辑和 UI 界面的关系选择上，让每个游戏对象负责处理自己的逻辑和动画，而不是将逻辑和画面 完全分开，每回合逻辑执行完成后渲染画面。因为前者虽然效率低了一些，但是总体上说更符合人的思路，对前期的架构设计要求也没有那么高，可以后期逐步完善，拓展性更强一些。

而在要求上，我完成了本学期所学的以下内容：

函数重载

类的继承和抽象基类

动态分配内存和文件操作

使用了 list 和 set 等数据结构

总体上来说，需要实现的模块和功能有：

渲染场景：包括加载欢迎界面和游戏主界面

多种鬼畜植物：游戏的主要实体对象之一

多种鬼畜僵尸：游戏的主要实体对象之一

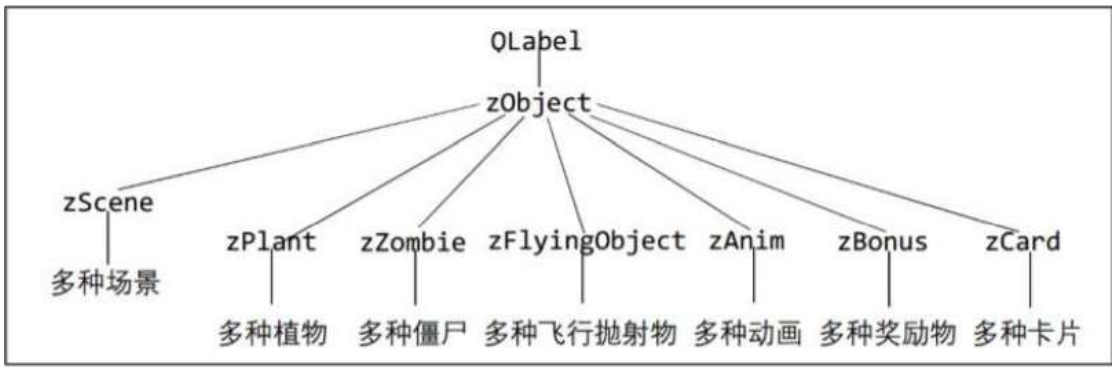
多种飞行物：如篮球、孢子等，也是游戏的重要对象

多种动画：在游戏中需要根据实物的状态来更新各种动画

奖励物：阳光等

多种卡片：主要指的是种植植物是选择的卡片，需要完成冷却功能

考虑到他们都具有相同的一些特性（例如都是实体对象，都有贴图，位置，自己的主逻辑函数函数），而且为了方便主逻辑中队每个对象的遍历（如更新状态，删除对象），因此我定义了 zObjcet 类作为共同的基类，由 zObject 类派生出其余的类别。而 zObject 类本身由 Qt 的基本类别 QLabel 派生，因为 QLabel 本身属于最简单的基本对象，而且具有贴图，播放动图，设置大小等基本功能，非常适合派生出其他对象并在此基础上增加功能。



基本对象 —— 架构、类的派生关系

本程序一共定义了 52 个类，除了一个继承自 Qt 最基本空间 QDialog 的 mianDialog 类用于生成游戏基本图形界面框，其余所有类都继承自 zObject 类，而 zObjcet 类自身继承自 Qt 自带基本类型 QLabel。

按照功能，zObject 类直接派生出七个类，分别对应上文所说的七种实体类

别和功能，七个大类分别再派生出具体的小类别，如 zPlant 类下派生出 9 种具体的植物类，而七种基类不参与具体的对象实例化，这样的优点是逻辑清楚，方便管理。下面我将主要介绍基类和其派生出的七个基本大类。

基类：zObject

其代码实现如下：

```
class zObject : public QLabel
    {
    Q_OBJECT
    public:
    explicit zObject(QWidget* parent = 0);
    bool alive = true;
    virtual void act()=0;
    int strength = 1;
    zScene* scene;
    };
```

基类 zObject 自派生 QLabel，因为 QLabel 非常合适定义基本实体：QLabel 的 SetMovie 配合 QMovie 是显示游戏动画（素材为 gif 格式）的便捷利器；而且 QLabel 也没有什么多余的属性与方法，基本来自 QWidget，只是一个普通的窗口组件。bool alive，表示它是否活着。本属性不一定表示通俗意义的“活着”，比如一个动画播放完了，我们就可以把 alive 置成 false，等着逻辑来把它删掉，可以用将本属性当作标记，交给逻辑中专门的死亡处理机制，从外

部释放掉内存空间，并把对象删除。

virtual void act()=0。这个虚函数就是游戏内所有对象的核心逻辑函数。主逻辑的重要功 能就是调用场上一切“活着”的对象的 act()来构成整体逻辑。

int strength 表示生命值。只有植物和僵尸有这样的属性，但是还是写进了接口。其余对象默认置 1 即可。

zScene* scene 提供了一个指向自己所处的 zScene 的指针，便于 act()逻辑来访问一些全局的对象。zScene 是游戏场景类，也由 zObject 派生。由于 parent 指针指向的 是 QWidget 而不是 zScene，此处 parent 指针没法优雅的访问 zScene 的一些特殊属性。parent 强制转化为 scene 以后就没有了这个问题。

场景管理类：zScene

zScene 是负责管理其他类的类，在它的属性里面有包含其余类对象的容器 QList，而其余类也有指向所在场景的指针。ZScene 派生出四个具体类：zStartScene, zStartScreen, zLawnScene, zDarkScene，分别对应开始界面，开始选关界面，白天关卡界面，和夜晚关卡 界面。

主要实体类：zPlant 类和 zZombie 类

这两个类具有很多相似的之处，因此放在一起介绍。zPlant 和 zZombie 是两种基本的对象，负责派生出各种丰富多彩的植物和僵尸。其共同点是都有 virtual void hit(int damage, bool silence = true)方法，供其余对象调用，对它产生伤害。植物有表示它所处网格位置的 int raw, column，僵尸有表示它在某行上的一维位置的 double xpos，另外僵尸还有诸多特殊状态都写在了僵尸基类里面，

比如冰冻。

飞行抛射类：zFlyingObjcet 类

飞行抛射物类，其特点是不断被发射和碰撞判定，代表物有豌豆、火球、孢子（小蘑菇 发射的）等。其 act()较为复杂，且有对僵尸 hit()方法的直接调用。目前碰撞判定采取的是纯一维逻辑，只判定同行上的碰撞。这里对于原游戏做了一些简化，未加入杨桃这种可以向五个方向发射小星星的机制。

动画类：zAnim 类

```
class zObject : public QLabel
{
Q_OBJECT
public:
explicit zObject(QWidget* parent = 0);
bool alive = true;
virtual void act()=0;
int strength = 1;
zScene* scene;
};
```

zAnim 动画类较为简单，只负责播放动画，放完就自动销毁，没有任何逻辑处理内容。zAnim 的用途十分广泛，

任何逻辑执行时都可能抛出一个或多个动画，用来可视化的展现逻辑的效果。

而动画本身用快速更新贴图和一些 gif 实现。

奖励物：zBonus 类

特点是响应用户的鼠标点击，目前主要是阳光，可以很容易的拓展出金币等奖励物对象。

卡牌类：zCard 类：

放在游戏画面最左侧的植物卡槽。包括铲子也是一种卡牌。具有冷却时间、判定阳光消 耗等功能。并且由拖动放置的效果。

作为游戏的测试，就是一遍一遍的运行+花式玩耍。这一点我在后期进行了较多的测试， 例如长时间运行，同时产生多个对象等。经过测试和修改，程序已经能够在比较极端的局面 下教稳定运行。以下是运行界面的一些效果：

为了方便测试，我定义了一些快捷键，用于参考：

数字键盘 1：产生普通僵尸

数字键盘 2：产生旗子僵尸

数字键盘 3：产生路障僵尸

数字键盘 4：产生铁桶僵尸

数字键盘 5：产生铁门僵尸

数字键盘 6：产生撑杆僵尸

数字键盘 7：产生报纸僵尸

数字键盘 8：增加 100 点阳光

数字键盘 9：跳过准备阶段，进入僵尸全面进攻状态

通过这样的一个项目，我的读代码，写代码，debug，测试能力都得到了极大的提升，也提高了我的自己解决问题的能力，比如搜索，和同学讨论，看书等。通过这样一个项目的锻炼，我明白了 C++是一个威力巨大的武器，其中的多样的机制例如重载，继承，派生和多态的用处极大，熟练使用这些机制能够很大程度上影响开发的效率，提升代码的质量。