

**面向对象程序设计**

**项目报告**

**专业班级： 软件工程2003班**

**小组名称： FivePeople小组**

**指导教师： 区士颀**

目录

[一、 项目提出 3](#_Toc91169715)

[(一) 项目简介 3](#_Toc91169716)

[(二) 项目目标 4](#_Toc91169717)

[二、 团队组成与分工 5](#_Toc91169718)

[三、 项目需求分析 6](#_Toc91169719)

[(一) 功能分析 6](#_Toc91169720)

[四、 系统设计 8](#_Toc91169721)

[(一) UI设计 8](#_Toc91169722)

[(二) 系统架构 10](#_Toc91169723)

[五、 问题、解决与所学 12](#_Toc91169724)

# 项目提出

## 项目简介

本次项目的名称是“PlantVsZombie”，即植物大战僵尸简易版小游戏，是以宝开公司所开发的经典游戏植物大战僵尸为蓝本，经过简化并分析功能后，通过Java语言实现的简易版本。

本游戏的基本功能与主体框架大致与原版相同，玩家通过收集阳光、消耗阳光以在草地上种植植物来对抗僵尸，阻止僵尸通过草地。在我们的项目中，我们总共设计了火爆辣椒、食人花、樱桃炸弹、火炬树桩、豌豆射手、寒冰射手、双发射手、三枪射手、机枪射手、向日葵、双子向日葵、地刺、刺石、坚果、高坚果、土豆地雷这十六种植物，它们的功能与特性类似于原版植物大战僵尸；同时，我们又设计了普通僵尸、路障僵尸、铁桶僵尸、读报僵尸、撑杆跳僵尸、橄榄球僵尸这六种僵尸，它们的特性也都类似于原版；除此之外，我们还实现了一个用于除去单位地址上的植物的工具——铲子。

除了游戏的内容，游戏的UI设计也由组内成员通过在网上寻找贴图资料模仿原版UI风格，通过Java中的Swing框架进行设计，一共分为三个界面:进入游戏时所显示的主界面、开始后选择植物与预览僵尸种类的准备界面、正式开始后的游戏操作界面。

游戏方式整体模仿原版，而代码实现方面则完全由小组成员自行设计、重构并实现。

## 项目目标

* 1. 项目开发目标

使用Java及其Swing工具编写一个界面精美，UI交互完善的“植物大战僵尸”游戏。

计划编写一个关卡，包含向日葵、豌豆射手，双重射手，坚果，地磁等多个植物，普通僵尸，路障僵尸，橄榄球僵尸等多种僵尸，配以除草车，铲子等游戏工具，以及计算阳光，选择卡牌的交互框条，和游戏音效与动画设置。

用户可以在主页面进入游戏，然后选择他想使用的植物，开局拥有150阳光，之后便是大家熟悉的植物大战僵尸游戏中的一关，失败后或者本关卡完成后结束游戏。

* 1. 重点功能和内容

1. 界面展示的UI交互
2. 选择卡牌
3. 产生阳光，收集阳光
4. 种植植物，完成植物与植物，植物与僵尸间的交互关系
5. 产生僵尸
6. 铲除植物
7. 游戏结束
   1. 关键技术，学习重点
8. 学会使用git，github管理代码，完成多人分工合作。
9. 学会进行架构分析，建立起完善的可靠的程序设计模型，使用UML的类图等工具建立起完整的软件模型，明确任务。
10. 学会使用Swing框架编写完整的UI交互界面，配合音乐，动作，照片，呈现出一个完整的游戏。
11. 对Java面向对象机制进行更深入的理解，充分学习类，继承，组合，多态等相关概念。
12. 完善异常管理机制，减少程序崩溃的可能性。
13. *如果时间充足，加入多个关卡与保存游戏进度功能，配合文件IO输入与数据库相关知识（按计划无时间实现）*

# 团队组成与分工

**组长：**

侯皓斐：主要负责统筹全局工作，分析模型，安排任务。和马川湖一起负责编写前端显示层和控制层。

**组员：**

马川湖：和侯皓斐一起负责编写前端显示层和控制层。维护github开源代码。

刘铭宸：负责模型层植物类的编写。

甘凤轩：负责模型层僵尸类的编写。

郑宇开：主要设计关卡难度，编写了ZombieProducer，SoundEffect类等关键类。

# 项目需求分析

## 功能分析

**1.主界面**

本游戏只包含冒险模式一个可玩性模式，而该模式中包含一个关卡。因此，主界面仅包含一个可供交互的按钮，即“冒险模式”的进入按钮，通过点击该按钮以进入游戏的准备界面。

**2.冒险模式**

**1）准备界面**

在准备界面，会分出已选PlantCard和未选择的PlantCard两个栏目，玩家需要在未选择的PlantCard中选择九个，之后方可点击页面下方的开始游戏按钮进入游戏操作界面，正式开始游戏。而在页面的右侧，则是在游戏中会出现的所有种类的僵尸的预览，即会在游戏开始后刷新的僵尸都是何种类。

**2）游戏操作界面**

游戏开始后，玩家会拥有150点初始阳光，玩家需要通过种植向日葵以生产出更多的阳光来种植其他诸如豌豆射手、土豆地雷等植物来抵御僵尸的侵袭，游戏过程中草地上会随机掉落一些阳光。每种植物的种植需要满足两个条件：拥有的阳光数量大于该植物所需的数量；该植物的冷却期结束——即每次该植物被种植后都将进入长短不一的禁用阶段，该阶段结束后方可继续种植该种植物。如果玩家想要替换单位草地上的植物，可以通过页面右上角的铲子按钮来除去原来的植物，之后再种植新的植物。

# 系统设计

## UI设计

电脑游戏的屏幕

描述已自动生成图形用户界面, 网站

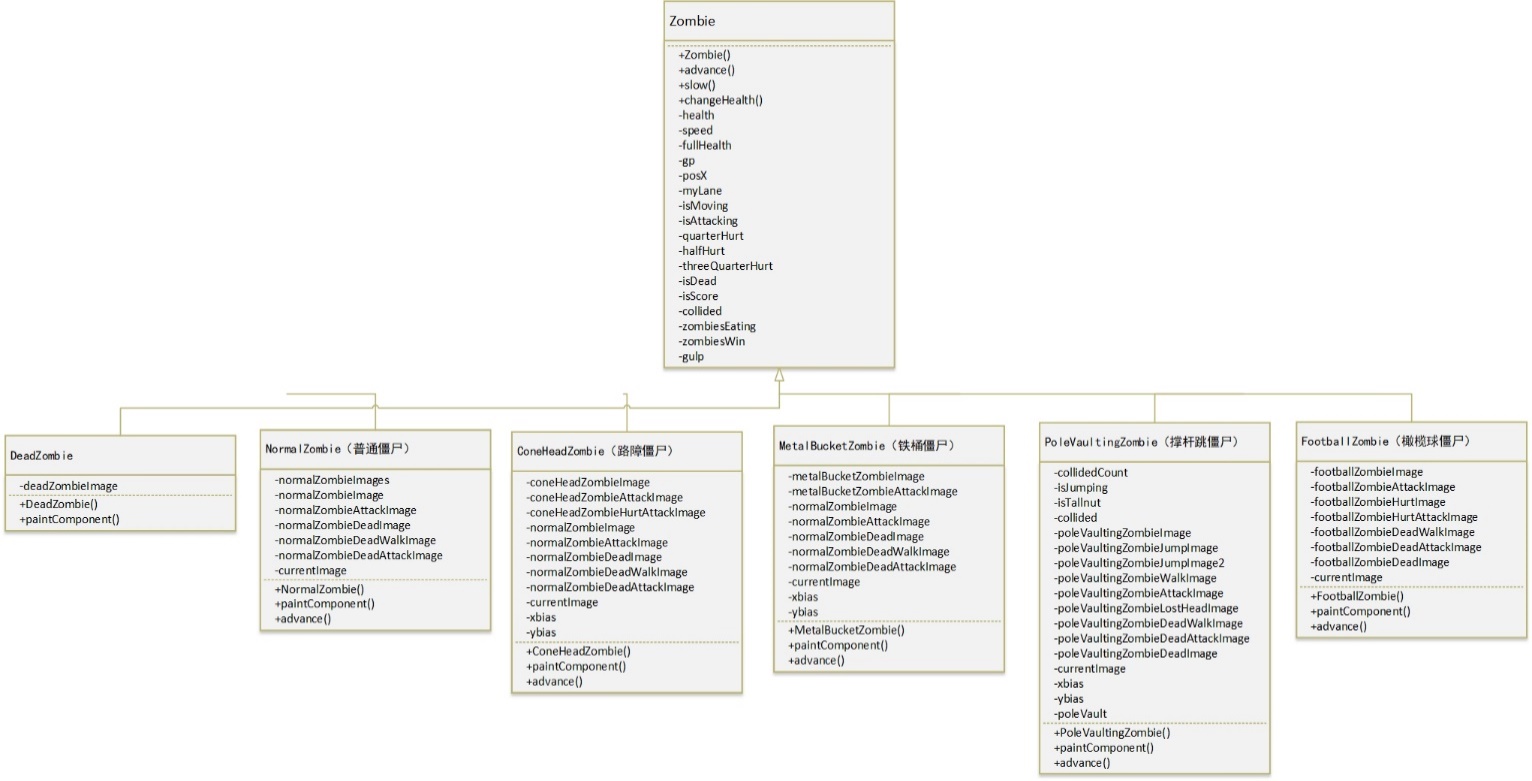
描述已自动生成

图形用户界面, 应用程序

描述已自动生成

我们的UI设计主要使用了Java的GUI设计中的awt.LayouManager包中的各种配件以及Swing事件响应机制，通过按钮的布局以及MouseListener的鼠标事件监听来对形成可供玩家交互的UI界面、能够对玩家的行为作出反应的各种标签按钮。

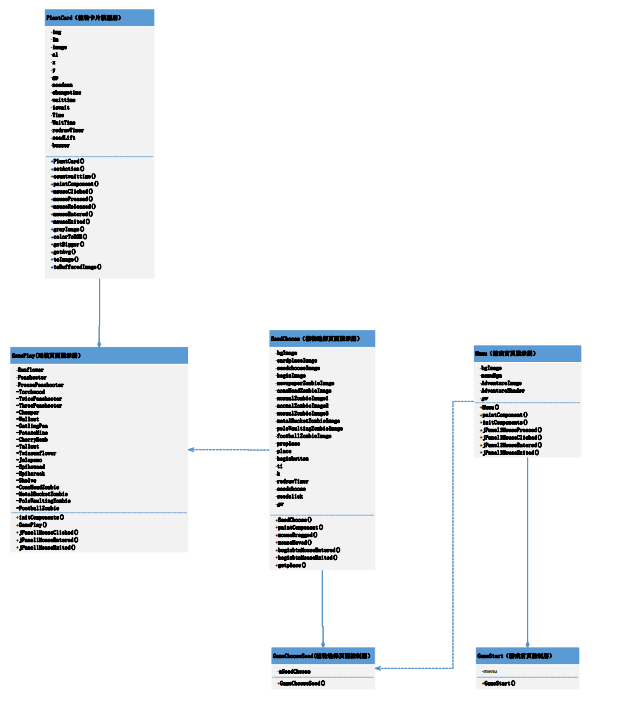
## 系统架构



在各种植物的设计中，我们使所有的植物都继承于一个集合了所有植物共同特点的Plant类，它含有诸如位置、血量、死亡、冷却时间等属性，而所有植物全部继承于Plant类且级别相同，并且每种植物类拥有自己的特性，如食人花有1、2、3三种状态分别代表待机、攻击、消化；土豆地雷有1、2、3三种状态分别代表准备、待机、爆炸等。同时还有一种专门为豌豆射手、双发射手、三枪射手、机枪射手设计的Pea类作为它们的子弹，并且还有继承于Pea类的FreezePea类作为寒冰射手的子弹和BurnPea类作为通过了火炬树桩后的特殊子弹。

对于所有的僵尸，它们都继承于一个Zombie类，Zombie类集合了移速、位置、血量、攻击、半血、残血、死亡、胜利越过草地等属性，Zombie类下继承了普通僵尸、路障僵尸、铁桶僵尸、撑杆跳僵尸、读报僵尸和橄榄球僵尸，它们各自拥有不同特点。比如普通僵尸、路障僵尸和铁桶僵尸只有image和health不同，而撑杆跳僵尸初始移速更快且可以越过一个除高坚果以外的植物，之后移速恢复普通等。

前端的控制层和显示层结构如下：（有所减省）



# 问题、解决与所学

在完成本次面向对象程序设计的过程中，我们小组遇到了很多困难和问题，包括在面对未知的java知识时的无从下手和程序出现bug时的束手无策，但最终在全小组成员的共同努力下，我们一个个克服了这些阻碍，并在解决问题的过程中得到了锻炼，学到了新的知识。

首先，在项目刚刚启动之时给我们造成很大困扰的就是对java图形界面编程的不熟悉。由于我们的项目无可避免地需要使用java Swing和AWT等工具包，而这又正好触及到了我们的知识盲区，所以在一开始我们只能先去学习这些包的使用。然而这些知识浩如烟海，纷繁复杂，并非一朝一夕就能学会的；再加之当时正逢考试周，时间安排十分紧张，我们的项目似乎难以再继续进行。为了解决这一问题，我们分工合作，通过在CSDN上查阅资料、在Github上参考学习等途径大致掌握了本次项目需要用到的awt，Swing等框架，JPanel、JFrame等组件，事件监听机制等的大概使用，结合阅读github开源代码，最终较为成功地设计并实现出了植物大战僵尸的图形界面。

另外，由于本次项目的代码量较大并且小组成员较为分散，如何将每个人所负责的部分进行统一汇总成为了一个难题。由此我们集体学习了利用git远程管理代码的方法。在完成本次项目的过程中，我们一直使用Github来分享汇总每个人的代码和进行版本的迭代，极大地提升了工作效率，同时也避免了因代码版本不统一引起的一系列问题。

除此之外，我们更深入的理解了，Java异常处理机制强制性的保证了程序在一些地方的安全性。尽可能地将运行时错误，转化为编译时错误。我们处理了Java虚拟机强制要求处理的异常，也自己尽可能地利用好异常处理减少bug产生。

每个植物卡片，僵尸，植物，豌豆等都是独立的对象，他们要时时刻刻控制着运动速率和刷新等性质，多线程机制对于处理问题必不可少。通过本项目，团队中每个人对多线程机制的理解更加深入。

与此同时，本次课程设计提升了我们建模、分析问题和解决问题的能力。从一开始对游戏机制、植物和僵尸的种类的设计到一步步实现游戏的功能，这一过程中我们分工协作，全员参与，摒弃了之前拿到问题直接写代码的陋习，而学会了用流程图、画UML图等方式使问题更加清晰，并利用接口、继承的方式先完成对程序的整体架构，再按层次逐步实现。其中我们体会到了什么才叫真正的面向对象编程，感受到了面向对象的设计方式在实际项目中的优越性和便捷性。

总之，在本次面向对象程序设计的过程中，我们不仅对java理解得更为深刻，学到了不少新的知识，更让我们领悟到了在项目中团队协作的重要性，让每个人都对应如何协调自身与团队的关系，以便能够为团队做出更大的贡献有了自己的理解。