高级语言程序设计

实验报告

南开大学 工科试验班（信息科学与技术）

姓名 蒋林瀞

学号 2413507

班级 0978

2025年5月12日

目录

[高级语言程序设计大作业实验报告 3](#_Toc20109)

[一、 作业题目 3](#_Toc14002)

[二、 开发软件 3](#_Toc30935)

[三、 课题要求 3](#_Toc22814)

[四、 主要流程 3](#_Toc16595)

[第一部分：学习gitee与github的使用 3](#_Toc14854)

[第二部分：考虑需要实现的功能： 3](#_Toc8102)

[第三部分：具体游戏实现 4](#_Toc5262)

[1. 创建游戏框架 4](#_Toc14380)

[2. 设计游戏角色和怪物 4](#_Toc11074)

[3. 实现角色控制 5](#_Toc18777)

[4. 设计背景和图像 6](#_Toc5500)

[5. 增加开始游戏窗口 6](#_Toc22671)

[6. 添加音效与音乐 6](#_Toc2261)

[7. 优化与调试 6](#_Toc31672)

[第四部分：遇到的问题及解决方法 6](#_Toc1886)

[五、学习过程中的收获 6](#_Toc10539)

[一、面向对象编程 6](#_Toc24698)

[二、Qt编程技能 7](#_Toc25497)

[三、游戏逻辑控制 7](#_Toc8580)

[四、资源管理 7](#_Toc2719)

# 高级语言程序设计大作业实验报告

## 作业题目

（自选题目）运用qt制作一个简化版本的植物大战僵尸

## 开发软件

Qt5

## 课题要求

完成一个图形化小程序

## 主要流程

### 第一部分：学习gitee与github的使用

### 第二部分：考虑需要实现的功能：

**确定游戏机制：** 参考知名游戏植物大战僵尸，控制植物种植并打怪，收集阳光等。将游戏改为积分制，但达到最高分数后强制胜利。

****打怪系统：****

1.怪物的种类、攻击方式、血量等。子弹发射，击中命中敌人，命中效果。

2.**僵尸生成与行为：** 创建怪物生成机制（在固定的线路终点几个格子内随机生成）。每个怪物有不同的攻击方式和行为模式（如移动、食用植物等）。

**3.攻击系统：** 设计一个攻击机制，当植物的子弹或者爆炸植物类的本体碰到僵尸，僵尸掉血，血条<=0之后僵尸死亡。

**自定义图像：** 人物、怪物、背景、道具等图像资源。

**关卡设计：** 简化版本不做特殊地图，调整游戏结构，使得僵尸为随机出现且积分制

**阳光系统：**

1.生成：阳光自动生成，向日葵植物生成。

2.阳光总数：一个计数器，记录阳光总数。

3.植物种植需要阳光：阳光不足时，不能种相关植物。

### 第三部分：具体游戏实现

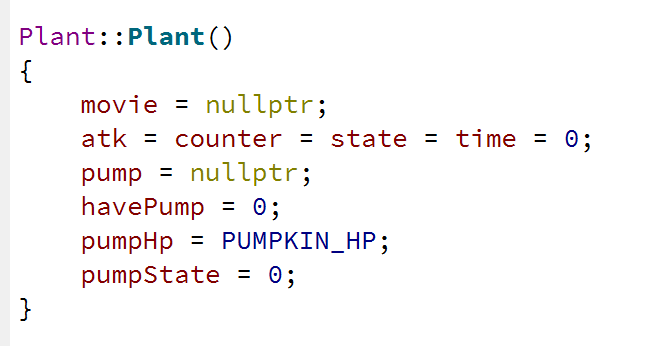
#### 1. ****创建游戏框架****

**初始化Qt项目：** 创建一个Qt MainWindow应用，选择合适的模板。

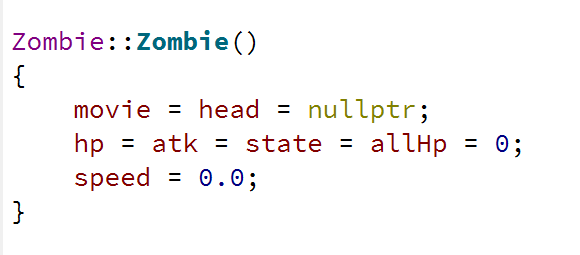
**设置游戏窗口：** 设置一个适当的窗口大小，作为游戏的主界面。（与找到的背景图片相符合）

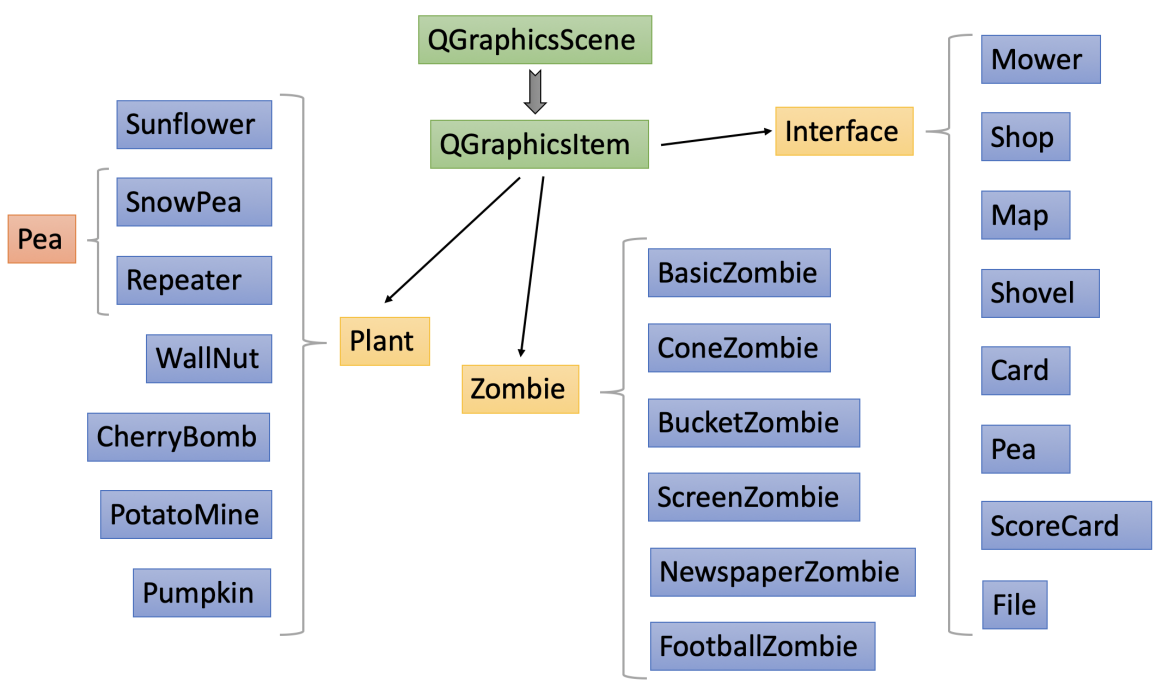
**管理游戏主循环：** 创建一个定时器（QTimer）来不断更新游戏状态，刷新显示（update）。

#### 2. ****设计游戏角色和怪物****

**植物类（包括派生类）：** 负责处理角色的移动、跳跃、攻击等逻辑。可以使用QMovie加载人物图像并进行显示。

**僵尸类（包括派生类）：** 用于管理僵尸的生成、移动、食用植物和生命值。





**碰撞检测：**

1.植物与僵尸，僵尸碰到植物之后需要更新状态，更新图像。

2.子弹与僵尸：子弹碰到僵尸，僵尸扣血。

3.僵尸与小推车之间的碰撞。僵尸碰到小推车会死亡。

#### 3. ****实现角色控制****

****植物的拖拽：**拖拽植物到“草坪”上进行种植，运用铲子可以移除植物。**

**植物动画：** 运用gif重复播放，实现植物“看起来在动”的状态

**僵尸动画与移动：**同样运用gif重复播放。僵尸的移动是固定的，需要运用坐标平移，加上碰撞检测。

**确定输赢条件：**僵尸的坐标进入房子则输，分数到达一定指标则获胜。

#### 4. ****设计背景和图像****

**自定义背景：** 网络上寻找植物大战僵尸的图片。

商店：用于存放植物卡牌

计分板：用于计分

按钮：包括pause（continue），exit等按钮，用于交互

#### 5.增加开始游戏窗口

Startwindow用于开屏显示，增加互动

#### 6. 添加音效与音乐

**背景音乐：** 使用QSound播放背景音乐。

#### 7. 优化与调试

**BUG修复：** 通过调试发现并修复游戏中的各种问题。

### 第四部分：遇到的问题及解决方法

1. 原先运用的是多个图片拼凑，运用QTimer实时更新zombie或者plant的图像，难以实现且效率低，难以管理。

解决方法：运用QMovie，寻找现有gif动图，直接改进。

2.游戏逻辑调度混乱  
 植物攻击僵尸、子弹打中僵尸、阳光收集等事件交织在一起，容易造成逻辑错误。

**解决方法：**使用 **状态机**（如僵尸有“行走”->“攻击”->“死亡”）管理角色行为。定义好每类对象的职责，例如：Plant只处理植物以及攻击逻辑。Zombie 处理行走和被击中。

3.植物放置逻辑难以确定

采用精确position放置，防止错放。

## 学习过程中的收获

### 一、****面向对象编程****

类的封装：如 Plant、Zombie类。

继承与多态：不同植物/僵尸行为差异的实现。

模块化设计：把功能划分成清晰的类和模块。

### 二、****Qt编程技能****

图形界面：熟悉 QWidget、QGraphicsScene 等绘图机制。

动画实现：使用 QTimer 控制帧动画、QPainter 绘图。

事件处理：处理鼠标点击、键盘操作、定时事件。

信号与槽：组件之间的通信机制。

### 三、****游戏逻辑控制****

游戏循环：定时刷新、角色状态更新。

碰撞检测：如子弹击中僵尸的判断。

状态机：植物和僵尸的不同状态管理（行走、攻击、死亡等）。

### 四、****资源管理****

图像资源加载与缓存：QPixmap、QImage 等。

动态对象管理：合理创建/销毁植物、僵尸，优化效率。