

# *PVZ base on unity*

## 设计分析报告

所属学院：计算机与大数据科学学院

小组名称：当然是 *Relax* 最重要啦

小组成员：张一鸿、林建斌、饶清、许哲琿

杨涵轩、范彬洋、郑宜湉

俞健坤、郑子豪、王真平

指导老师：程永利

## 目录

一、引言 .....	3
1.1 编写目的 .....	3
1.2 项目介绍 .....	3
1.3 参考资料 .....	3
二、系统体系结构设计 .....	4
2.1 体系结构风格 .....	4
2.1.1 面向对象的体系结构风格 .....	4
2.1.2 主要特点 .....	4
2.1.3 优点 .....	4
2.2 整体视图 .....	5
2.2.1 整体视图 .....	5
2.2.2 系统的构件 .....	5
2.2.3 构件之间的依赖关系 .....	5
三、构件设计 .....	5
3.1 构件间的接口 .....	5
3.2 构件细化 .....	6
3.2.1 植物部分 .....	6
3.2.2 僵尸部分 .....	7
3.2.3 其他 .....	9
四、界面设计 .....	12
4.1 欢迎界面 .....	12
4.2 主菜单界面 .....	12
4.3 选关界面 .....	13
4.4 主菜单设置界面与游戏中设置界面 .....	13
4.5 游戏选植物界面 .....	14
4.6 游戏画面 .....	15

# 一. 引言

## 1.1 编写目的

本份报告目的在于明确说明“PVZ base on unity”的软件体系风格以及进行系统体系结构设计和各个构件的细化设计，便于团队成员基于此报告，进行软件的代码编写、测试等工作

## 1.2 项目介绍

- (1) 软件名称: PVZ base on unity;
- (2) 开发团队: 软件工程-2021-C班 (福州大学) —— 当然是 Relax 最重要啦;
- (3) 团队人数: 10 人;
- (4) 项目简介: 基于 unity 实现的可玩性、娱乐性更强，受众更广的植物大战僵尸。

## 1.3 参考资料

软件需求说明书 (GB856T-88)

《软件工程导论》第三版

—————张海藩编著  
清华大学出版社

《基于项目的软件工程一面向对象研究方法》

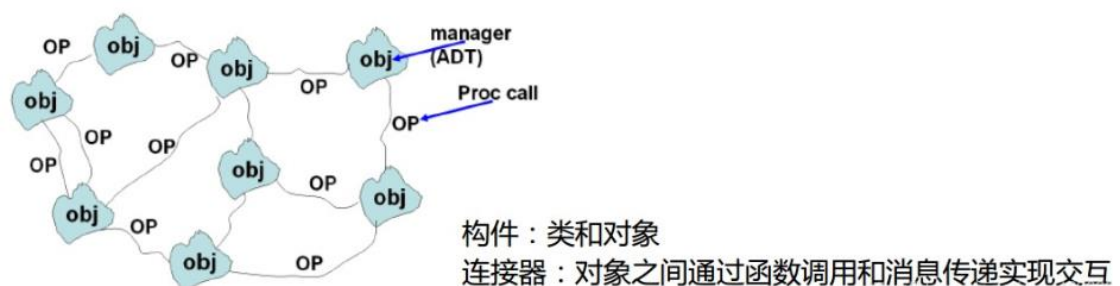
—————Cathie LeBlanc & Evelyn stiller 编著  
机械工业出版社

## 二. 系统体系结构设计

### 2.1 体系结构风格

#### 2.1.1 面向对象的体系结构风格

本项目基于面向对象的体系结构风格，通过构件（类和对象）和连接件（函数和过程的调用）来进行软件体系结构的建立。



#### 2.1.2 主要特点

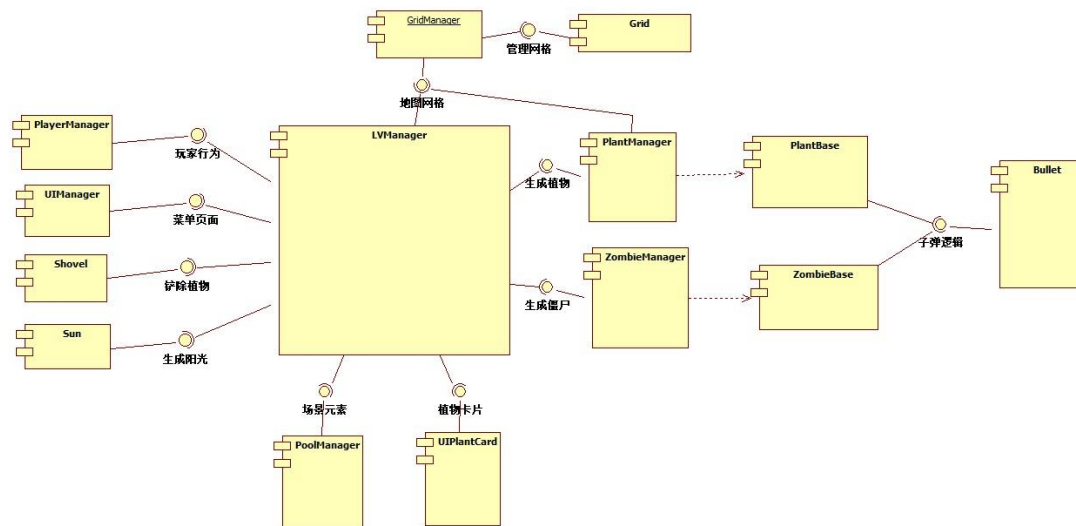
- ①对现实世界的自然建模方法
- ②封装(数据表示和相关的基本操作被封装，限制访问)
- ③继承(共享功能定义，可以在继承一个已定义好的类的所有特征的基础上逐渐定义更具体的对象)-重用

#### 2.1.3 优点

- ①强调设计，全面的系统考虑，和现实世界的对应关系
- ②对象对客户隐藏了实现的细节，可以在不影响客户的情况下改变对象的实现(低耦合，高重用，可维护)，方便系统升级
- ③ 内部表达的保护(封装数据/状态的完整)

## 2.2 整体视图

### 2.2.1 整体视图



### 2.2.2 系统的构件

系统的构件如整体视图中所示，主要分为三个大类：植物、僵尸、UI 及其相关控制器和游戏逻辑控制器。

### 2.2.3 构件之间的依赖关系

构件之间的依赖关系如整体视图中所示，玩家通过玩家控制器(PlayerManager)中的方法，围绕关卡构件(LVManager)及其中包含的方法，来管理和控制其他构件，完成游戏逻辑的实现。

例如：玩家通过 PlayerManger 调用 Game 中的方法去控制 SunManager 进行阳光的收取。

## 三、构件设计

### 3.1 构件间的接口

构件间的接口如上述构件图中所示。玩家通过

## 3.2 构件细化

将三个大类（植物、僵尸、UI 及其相关控制器和游戏逻辑控制器）的构件进行如下细化设计。

### 3.2.1 植物部分

植物基类：



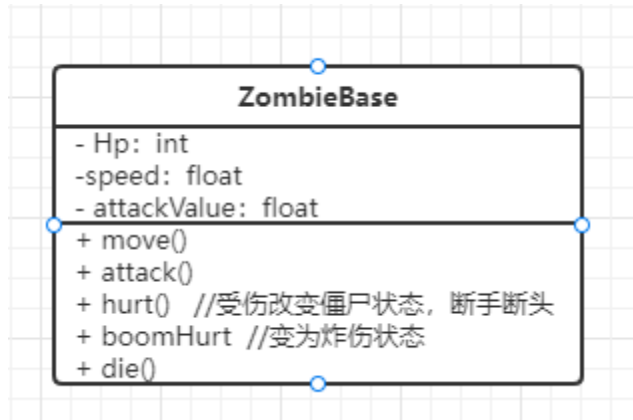
各个植物：植物在继承了基类属性的基础上定义各自的特殊属性，如血量，攻击力，攻击速度，攻击动画，特殊功能等。

<b>SunFlower//太阳花</b> ~createSunTime:float//创建阳光需要的时间 ~goldWartTime:float//安全色需要的时间 ~MaxHp:float//血量 ~OnInitForPlace():void //初始化 ~CreateSun():void//创建阳光 ~InstantiateSun():void//阳光关联	<b>Peashooter//豌豆射手</b> ~MaxHp:float//血量 ~AttackCD:float//攻击冷却 ~AttackValue:int//攻击力 ~canAttack:bool//是否可以攻击 ~createBulletOffsetPos:Vector3//子弹位置偏移量 ~OnInitForPlace():void//初始化 ~Attack():void//检测攻击 ~CDEnter():void//子弹进入冷却	<b>Repeater//双发豌豆</b> ~MaxHp:float//血量 ~AttackCD:float//攻击冷却 ~AttackValue:int//攻击力 ~canAttack:bool//是否可以攻击 ~createBulletOffsetPos:Vector3//子弹位置偏移量 ~OnInitForPlace():void//初始化 ~Attack():void//检测攻击 ~CDEnter():void//子弹进入冷却	<b>Threepeater//三线豌豆</b> ~MaxHp:float//血量 ~AttackCD:float//攻击冷却 ~AttackValue:int//攻击力 ~canAttack:bool//是否可以攻击 ~createBulletOffsetPos:Vector3//子弹位置偏移量 ~OnInitForPlace():void//初始化 ~Attack():void//检测攻击 ~CDEnter():void//子弹进入冷却	<b>SnowPea//寒冰豌豆</b> ~MaxHp:float//血量 ~AttackCD:float//攻击冷却 ~AttackValue:int//攻击力 ~canAttack:bool//是否可以攻击 ~createBulletOffsetPos:Vector3//子弹位置偏移量 ~OnInitForPlace():void//初始化 ~Attack():void//检测攻击 ~CDEnter():void//子弹进入冷却
<b>WallNut//坚果墙</b> ~MaxHp:float//血量 ~animationName:string//状态动画名 ~#IfUpdateEvent():void//状态动画转换	<b>Spike//地刺</b> ~MaxHp:float//血量 ~AttackCD:float//攻击冷却 ~AttackValue:int//攻击力 ~ZombieCanEat:bool//僵尸能否攻击 ~offset:Vector2//相对网格偏移量 ~OnInitForPlace():void//初始化 ~Attack():void//检测攻击	<b>Cherry//樱桃炸弹</b> ~MaxHp:float//血量 ~AttackValue:int//攻击力 ~OnInitForPlace():void//初始化 ~Attack():void//检测攻击 ~Boom():void//爆炸过程	<b>Sun//太阳</b> ~Instance:Sun//实例 ~downTargetPos:Vector3//下落的目标点Y ~isPicked:bool//是否已被拾取 ~isFormSky:bool//是否来自天空 ~Awake():void//初始化阳光 ~Update():void//阳光下落判断 ~OnMouseDown():void//鼠标拾取阳光 ~AutoPickSun():void//自动拾取阳光 ~InitForSky():void//初始化阳光从天空下落 ~InitForSunflower():void//初始化阳光来自太阳 ~DoJump():void//阳光跳动 ~FlyAnimation():void//飞行动画 ~DestroySun():void//销毁自身	<b>Bullet//子弹</b> ~CanDown:bool//能否下落 ~CanUp:bool//能否上升 ~y:float//记录一开始y ~rigidbody:Rigidbody2D//子弹刚体 ~#spriteRenderer:SpriteRenderer//精灵 ~AttackValue:int//攻击力 ~isHit:bool//是否击中 ~OnInitForPlace():void//初始化 ~Attack():void//检测攻击 ~CDEnter():void//子弹进入冷却 ~Init1():void//初始化子弹直飞 ~Init2():void//初始化子弹先斜上飞再直飞 ~Update():void//更新子弹状态 ~OnTriggerEnter2D():void//检测刚体碰撞僵尸状态 ~Destroy():void//销毁自身

<b>Tall- Nut</b> ~ HP: int	<b>Spikerock</b> ~HP: int ~AttackValue: int ~AttackCD: int ~Die()	<b>Doom-Shroom</b> ~prepareTime: int //准备时间 ~AttackValue: int ~Attack() ~makeCrater() //生成弹坑	<b>Garlic</b> ~Hp: int ~Hurt() ~ChangeRoute()
<b>Chomper</b> ~ HP: int ~AttackCD: int //咀嚼时间 ~AttackValue: int ~Attack()	<b>Squash</b> ~HP: int ~AttackValue: int ~Attack()	<b>Scaredy-Shroom</b> ~HP: int ~AttackCD: int ~AttackValue: int ~sleep() //白天睡觉 ~hiding() //有僵尸最近时躲藏	<b>Sun-Shroom</b> ~HP: int ~productValue: int //生产阳光的值 ~productCD: int //生产间隔 ~growTime: int //长大时间 ~product() ~grow() //长大后改变生产值
<b>Twin Sunflower</b> ~HP: int ~productValue: int //生产阳光的值 ~productCD: int //生产间隔 ~product()			

## 3.2.2 僵尸部分

僵尸基类：



各个僵尸：僵尸在继承了基类属性的基础上定义各自的特殊属性，如血量，攻击力，速度，行走、攻击动画，特殊功能等。

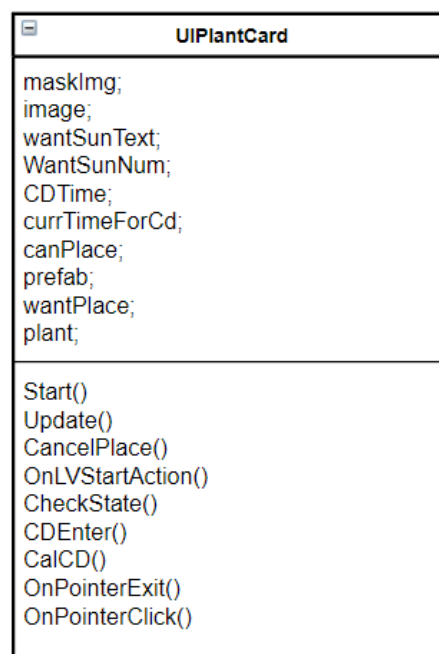
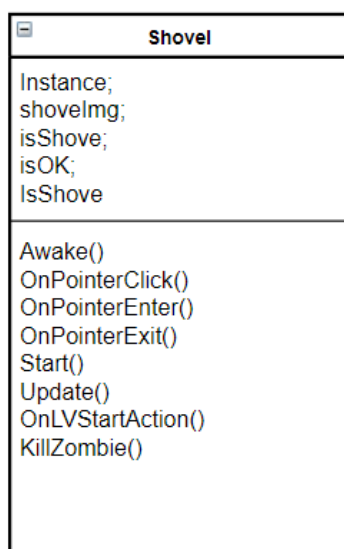
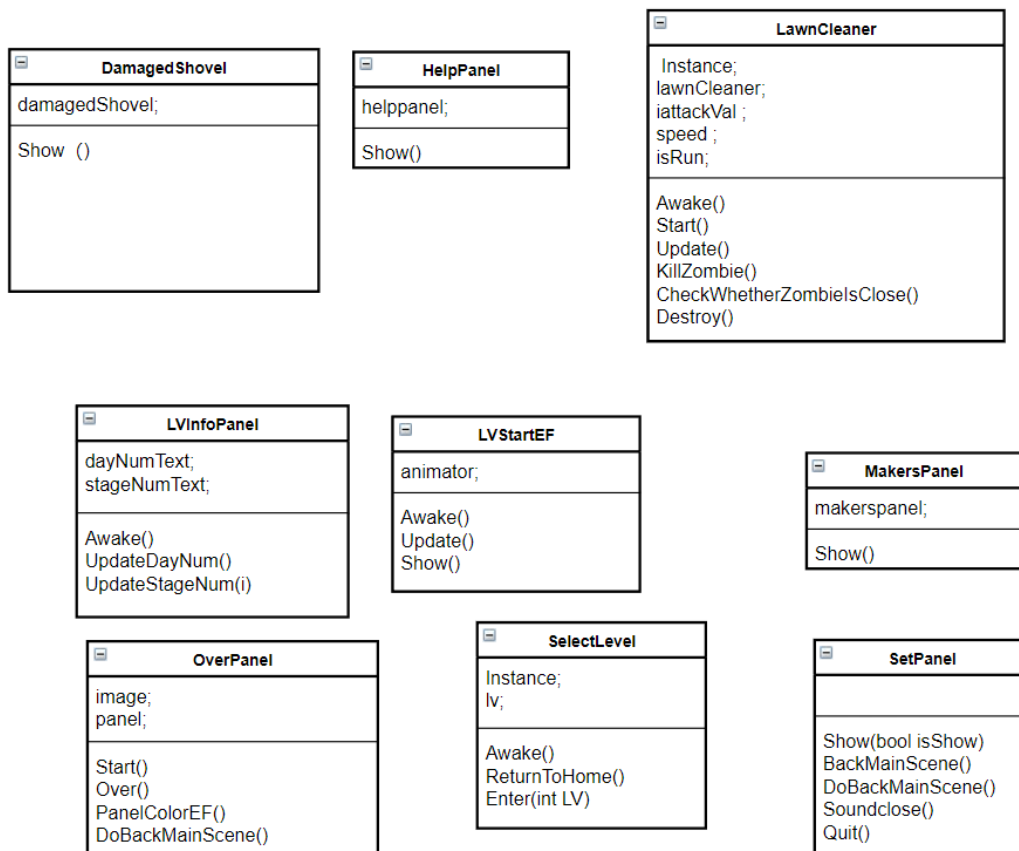




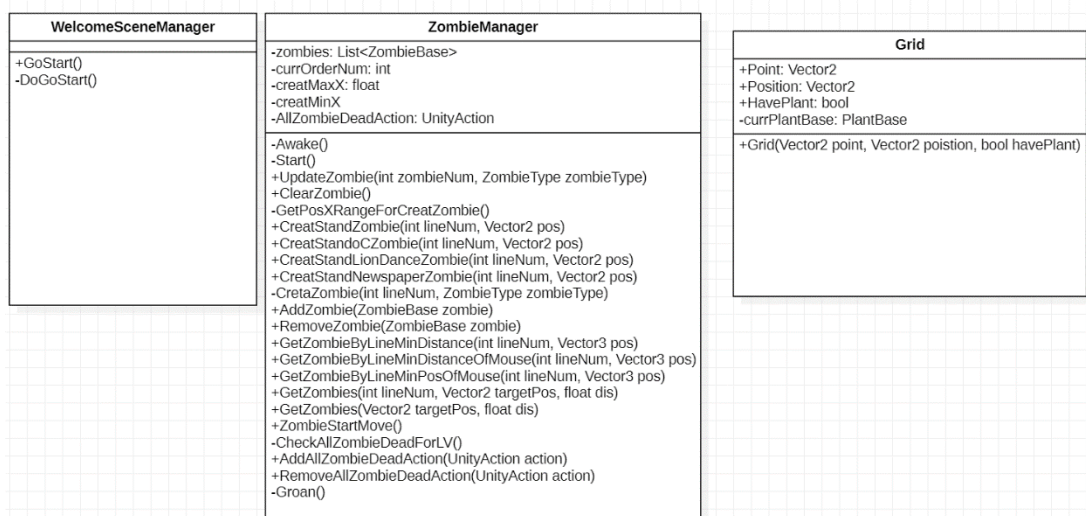
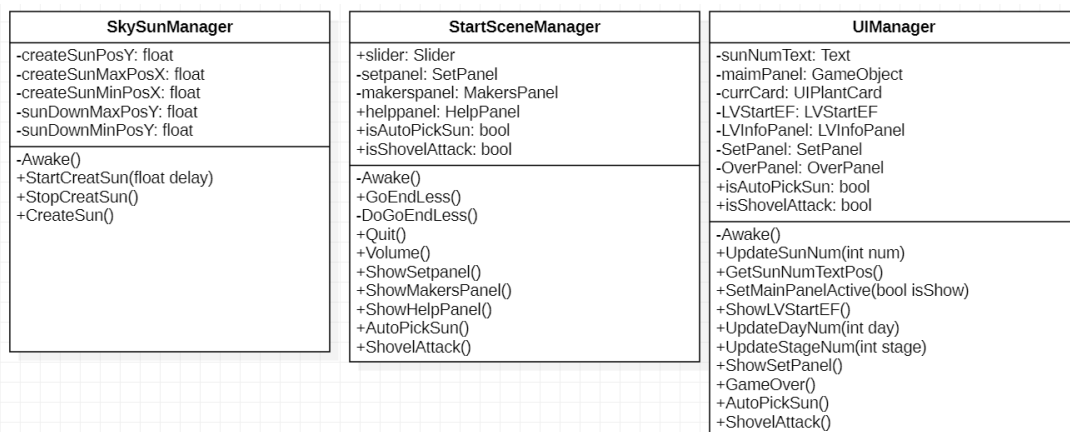
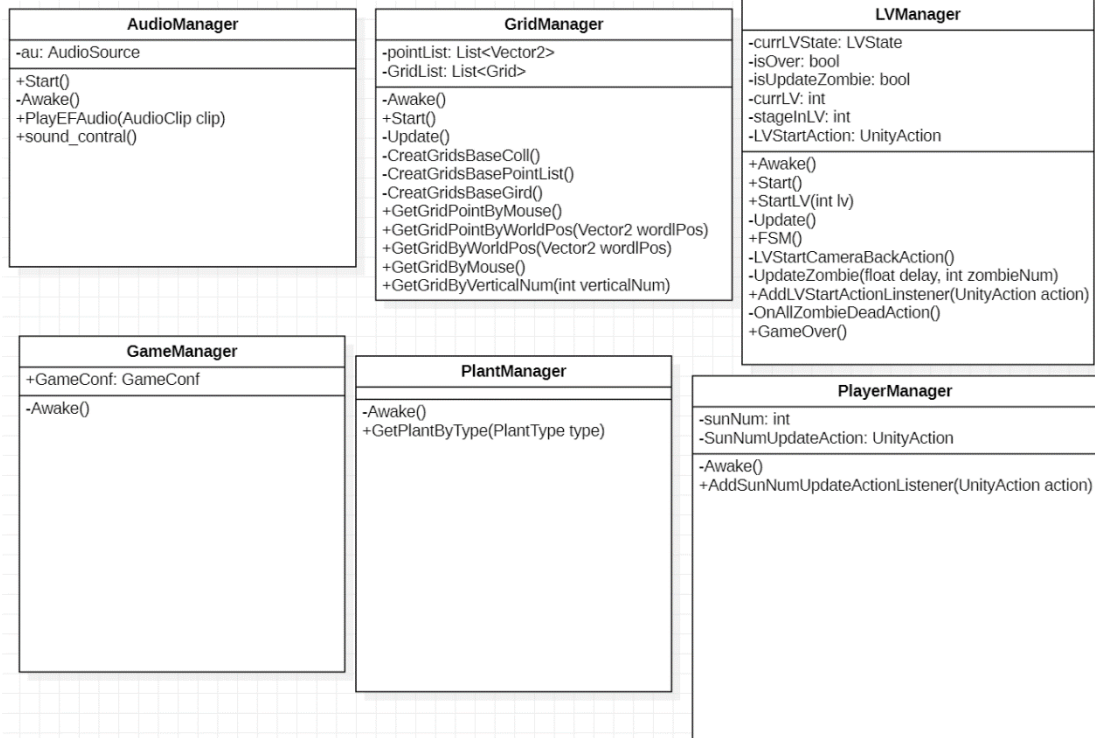


### 3.2.3 其他

UI 部分：



控制器部分：

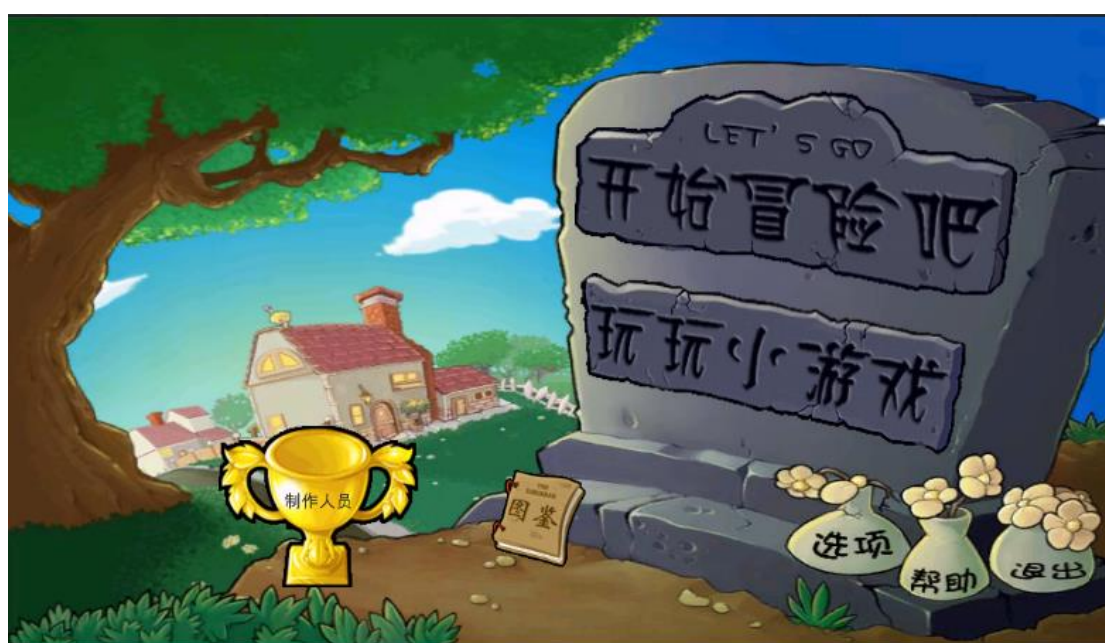


## 四、界面设计

### 4.1 欢迎界面



### 4.2 主菜单界面





#### 4.3 选关界面



#### 4.4 主菜单设置界面与游戏中设置界面





#### 4.5 游戏选植物界面





## 4.6 游戏画面

