# 基于funcode平台的植物大战僵尸

概

要

设

计

说

明

书

**题 目 植物大战僵尸\_\_\_\_\_\_\_\_\_\_\_**

**学 校 山东理工大学\_\_\_ \_\_\_\_**

**组 员 王青海 张国辉 黄海 张广琦 刘媛媛**

**撰写日期 2020.9.10\_\_\_\_\_\_\_\_\_\_\_\_**

目录

[背景及目的](#header-n4)  
 [运行环境](#header-n18)  
 [主要功能](#header-n27)  
 [项目结构](#header-n39)  
 [总体设计](#header-n44)  
 [冒险模式模块设计](#header-n47)  
 [迷你游戏模块设计](#header-n51)  
 [益智游戏模块设计](#header-n54)  
 [关卡设计及信息](#header-n58)  
 [游戏成绩持久化](#header-n266)  
 [项目总结](#header-n278)

## 背景及目的

* 名称：植物大战僵尸特别版
* 开发语言：c++
* 开发成员：王青海 张国辉 黄海 张广琦 刘媛媛
* 用户群体：所有喜欢玩这个游戏的玩家
* 目的：
* 软件开发过程中的概要设计，主要解决实现该游戏需求的程序模块设计问题。包括如何把该游戏划分成若干个模块、决定各个模块之间的接口、模块之间传递的信息，以及模块构造的设计等

## 运行环境

* 硬件要求：对计算机的硬件要求不高，普通计算机即可。
* 操作系统：Windows XP及以上任何版本
* 开发环境：c++、funcode平台

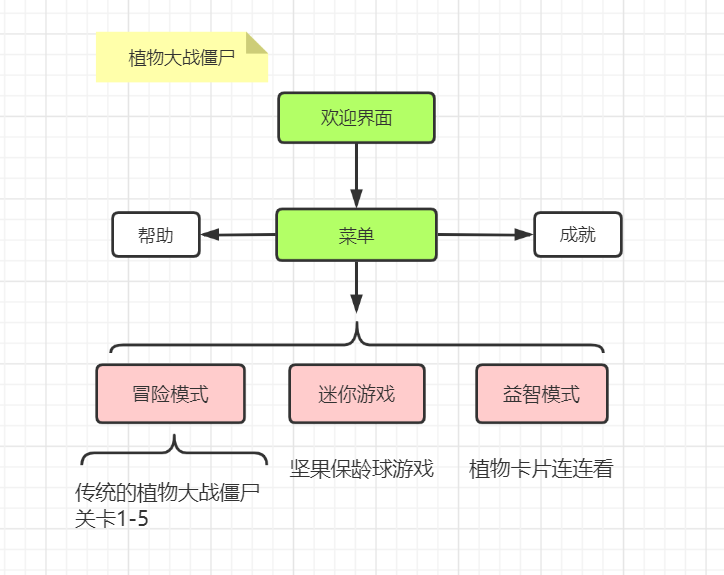
## 主要功能

* 冒险模式（共设计5关）
* 植物大战僵尸，每局游戏的开始给予一定的阳光作为启动资金，消耗阳光购买种子种植植物，向日葵等植物可以产生阳光，同时植物也有一定的种植冷却时间。种植攻击性的植物可以杀死僵尸。当僵尸进入家中，意味着游戏走向失败。整局游戏的用时将会被记录作为得分评定。只有打败所有的僵尸才能取得游戏的胜利。
* 益智游戏
* 基于植物卡片的连连看游戏，只有消掉所有的卡片才能取得游戏的胜利。
* 迷你游戏
* 坚果保龄球，种植坚果滚动撞倒僵尸，其中坚果还会进行随机角度区间内的反弹去撞其他方向的僵尸。撞倒所有的僵尸，获得游戏胜利。

## 项目结构

├─Bin  
│ ├─common  
│ ├─game  
│ │ ├─data  
│ │ │ ├─audio  
│ │ │ ├─images # 图片素材  
│ │ │ │ └─static  
│ │ │ └─levels  
│ │ ├─gameScripts  
│ │ ├─gui  
│ │ └─managed  
│ └─sound # 音乐  
│ ├─bgm  
│ ├─sem  
│ └─ui  
├─doc  
├─SourceCode # 源码  
│ ├─Header  
│ ├─Src   
│ └─VCProject2015  
└─tools # 图片处理工具类

## 总体设计



### 冒险模式模块设计

植物大战僵尸，每局游戏的开始给予一定的阳光作为启动资金，消耗阳光购买种子种植植物，向日葵等植物可以产生阳光，同时植物也有一定的种植冷却时间。种植攻击性的植物可以杀死僵尸。当僵尸进入家中，意味着游戏走向失败。整局游戏的用时将会被记录作为得分评定。只有打败所有的僵尸才能取得游戏的胜利。

### 迷你游戏模块设计

从滚动的卡片选择坚果种下，当坚果种下时，坚果会滚动撞倒路上的僵尸，撞倒的瞬间会由于碰撞发生反弹，再次随机撞向其他位置的僵尸。

### 益智游戏模块设计

基于植物卡片的连连看游戏。

### 关卡设计及信息

**关卡信息**

|  |  |  |  |
| --- | --- | --- | --- |
| 关卡名字 | 背景图片名称 | 植物信息 | 僵尸信息 |
| 冒险模式关卡一 | day*yard*base.jpg | 向日葵 豌豆射手 三管豌豆 墙果 樱桃炸弹 土豆地雷 | 普通僵尸\*1 足球僵尸\*1 报纸僵尸\*1 路障僵尸\*1 水桶僵尸\*1 |
| 冒险模式关卡二 | day*yard*bg.jpg |  |  |
| 冒险模式关卡三 | day*yard*bg.jpg |  |  |
| 冒险模式关卡四 | night\_bg.jpg |  |  |
| 冒险模式关卡五 | night\_bg.jpg |  |  |
| 迷你游戏 | 保龄球 | 规定时间内，安放坚果，撞倒僵尸取得胜利 | - |
| 益智游戏 | 连连看 | 58张植物卡片随机抽取n张组成40张卡片，只有消去所有卡片才能取得胜利 | - |
|  |  |  |  |

**植物信息表**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 名字 | 费用 | 伤害/产能/强度 | 攻击力 | 血量 | 恢复 | 范围 |
| 喷射蘑菇 | 0 | 伤害：普通 | - | - | 快 |  |
| 樱桃炸弹 | 150 | 伤害：极大 | - | - | 非常慢 | 中型区域内的所有僵尸 |
| 胡椒 | 150 | 伤害：极大 | - | - | 非常慢 | 一条道上的所有僵尸 |
| 土豆地雷 | 25 | 伤害：极大 | 8s | 500 | 慢 | 小型区域内的所有僵尸 |
|  |  |  |  |  |  |  |
| 豌豆射手 | 100 | 伤害：普通 | 25 | 500 | 快 | 一条道上的所有僵尸 |
| 三管豌豆 | 325 | 伤害：普通（每颗豌豆） | 25x3 | 500 | 快 |  |
| 机枪豌豆 | 250 | 射速：四倍 |  |  | 非常慢 | 机枪豌豆一次发射四颗豌豆 |
|  |  |  |  |  |  |  |
| 向日葵 | 50 | 产能：普通 | - | 300 | 快 |  |
| 阳光蘑菇 | 25 | 产能：少 | - | 500 | 快 |  |
| 墙果 | 50 | 强度：高 | - | 4000 | 慢腾腾 |  |
| 注： | **“非常慢”=50s“慢”=30s“快”=7.5s** | | | | |  |

**僵尸信息表**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名字 | 强壮度 | 血量 | 攻击力 | 速度 |
| 普通僵尸 | 低 | 200 | 50 | 2 |
| 报纸僵尸 | 低 | 450 | 50 | 2 速度：低，然后高（失去报纸之后） |
| 塑料僵尸（路障僵尸） | 中等 | 640 | 50 | 2 |
| 水桶僵尸 | 高 | 1370 | 50 | 2 |
| 足球僵尸 | 很高 | 1400 | 50 | 3.5 速度：快 |
|  |  |  |  |  |

## 游戏成绩持久化

**音乐配置pvzConfig.ini**

结合**MCI媒体库** 手动封装的音乐类，整合代码播放游戏全局音效音乐

|  |
| --- |
| #背景音乐 - 菜单 open-menu=open sound\\bgm\\menu.mp3 alias menu play-menu=play menu repeat  #特效音乐 - 豌豆打到僵尸身上 open-hit=open sound\\sem\\pea-hit.mp3 alias hit play-hit=play hit close-hit=close hit  ...  # ui点击音乐1 open-click1=open sound\\ui\\click1.wav alias click1 play-click1=play click1 close-click1=close click1  #关闭所有 close-all=close all  #示例：SuperSound::closeAndPlay("open-shoot", "play-shoot", "close-shoot"); |

**冒险模式关卡信息adventureConfig.ini**

|  |
| --- |
| [level\_1] BackgroundName=day\_yard\_base.jpg OrdinaryZombie=10 BarricadeZombie=10 BucketheadZombie=10 NewspaperZombie=10 FootballZombie=1 comeTimes=3  [level\_2] ....  [other] Name=PVZ version=1.0.0 author=Roll-over Team |

**关卡得分记录score.ini**

采用系统自带的ini文件读写接口进行读写文件

GetPrivateProfileInt() 和 WritePrivateProfileString()

|  |
| --- |
| # 记录保存 [level\_score] level\_1=1 level\_2=1 ...  [level\_time] level\_time\_1=100 level\_time\_2=161 level\_time\_3=0 level\_time\_4=0 level\_time\_5=0  [other\_time] bowling\_time=36 mini\_time=0 |

## 项目总结

* 对c++项目整体架构编程和开发模式更加熟悉。
* 通过使用github多端协作编程，对commit，merge，pull，push等常用的git操作更加熟悉。
* 自己动手封装音乐播放类、ini配置文件读取写入等工具类，对封装，可扩展性等概念有了更深层次的理解。
* 对僵尸、植物等一些精灵进行单例创建，容器化管理。对降低内存消耗，提高性能有了更好的认识。
* 实现连连看的寻路算法，编程思维得到进一步提升。