项目开发总结报告

# 引言

**此文档致力于介绍java程序开发2022年春季学期本实践小组的项目“植物大战僵尸”纯java版本。编撰于实践项目的收尾之际，本报告涵括本学期的实际开发结果，包括产品具体介绍、主要功能和性能、开发进度、项目总结和评价与经验教训。**

# 实际开发结果

## 产品

说明最终制成的产品，包括：

1. 程序系统中各个程序的名字，它们之间的层次关系；

|  |  |
| --- | --- |
| 程序名 | 使用到的组件 |
| 开始界面程序 | 上部已选植物列表，下部植物选择卡片栏，植物卡片部分 |
| 游戏界面程序 | 上部已选植物列表，下部游戏主界面画面，游戏进度条，各个对象等 |
| 植物程序 | 各个植物对象及其功能和函数 |
| 僵尸程序 | 各个僵尸对象及其功能和函数 |

# 开发工作评价

## 对生产效率的评价

给出实际生产效率，包括：

1. 程序的平均生产效率，即每人月（或人日）生产的行数；
2. 文件的平均生产效率，即每人月（或人日）生产的页数或字数；
3. 测试的平均生产效率，即每人日执行的测试用例数目

并列出原订计划数作为对比。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **项目** | **实际生产效率** | **计划生产效率** | **比较结果** | **原因分析** |
| 程序 | 40行/人\*日 | 25行/人\*日 | 增加 | 1. 代码习惯不够好导致代码逻辑有些冗余 2. 没有使用组件减少前端复杂度 3. 本身功能难度体量超出预估 |
| 文档 | 23字/人\*日 | 30字/人\*日 | 减少 | 1. 文档工作不够重视 2. 敏捷开发过程中简化了部分文档 |
| 测试 | 10行/人\*月 | 100行/人\*月 | 减少 | 1. 测试形式和预估出现区别，只有后端在进行单元测试，前端主要选择了手动测试 2. 集成测试和功能测试直接选择了运行测试 |
| 图片及文件 | 7页/人\*月 | 4页/人\*月 | 增加 | 1. 不了解需要进行的图片绘制 2. 每一次报告的日渐优化 3. OOA和OOD所需要的内容 |

## 对产品质量的评价

我们的基础功能调试中发现各个对象间交互过程出现了问题。整体而言，执行情况良好，目的达成，通过测试结果的分析改进平台设计和实现细节。本次迭代的测试工作完成过程中，工作组织采用了集体工作的模式，大部分组员选定合适的时间、地点集中工作进行测试。实践看来，这种组织模式能有效提高工作效率：测试中的问题能被负责这一部分的组员迅速发现，并得到高效解决。

**存在的问题：**

1. 时间不足：迭代时间不足，本身是使用敏捷开发的模式，并没有比较明确的一个开发的限定流程，导致并不能很好的估计开发时间和配合当中会出现的问题
2. 缺少交互经验，一开始的各对象分离开发，导致后期的合并调试出现比较严重的问题，虽然后期都有完美解决，下一次需要提前做好相应的规划。

所以总体上来说，本产品的质量较为良好可运行。

## 对技术方法的评价

我们在这一小节，评价在开发中使用的技术、方法、工具、手段。

**技术**：我们在开发中采用了采用了面向对象和结构化的方法

**优点**：面向对象：从对象入手规划我们的需求和需要实现的效果，使得目标和任务更加清晰，避免了很多的中间转换环节和多余劳动，加快了系统开发的进程，使得开发过程更有效率更易分工。同时，面向对象方法使得我们的功能更易扩展，也更加容易进行测试和维护。同时注重各个对象和函数间的重用性，可以减少很多不必要的代码和工作。

**缺点**：在需要对对象进行更改时非常难以维护，因为所有的信息和操作基于对象，所以调整的时候需要非常好的迁移方法。且类和继承等方式需要维护更多虚拟表来达成对应的操作，增加了程序出错的可能性。结构化自顶向下设计，但是在开发过程中会出现很多的调整，所以不能再一开始做出一个扩展性足够好的方案。

**方法：**我们在开发中，采用了敏捷开发的开发模式。

**优点：**更新迭代整体而言是速度较快，效率较高的。在一次迭代的过程中，我们小组在很短时间内完善功能若干，开发新功能若干，集成测试若干，很适合短期合作项目的进行。

**缺点：**速度较快，效率较高的开发模式能够让初次合作的团队尽快推进项目进展，但是局限于软件不成熟的开发者自身的能力，往往在敏捷开发的过程中出现一些手忙脚乱的问题。比如，在迭代任务的重压下，本组的程序员们往往急于实现一些功能，可能对产出代码的质量缺乏管控导致易读性差，其他队友上手难度大，或者不同组件代码之间的兼容性不好，导致在后面开发中反复占用开发者的时间，这些问题是可以通过一些手段在敏捷开发中得以避免的，但局限于组员们的自身能力和项目经验，在开发初期，这些问题规避得不是很好。

# 经验与教训

列出从这项开发工作中所得到的最主要的经验与教训及对今后的项目开发工作的建议。