

**Bubibura跑酷小游戏**

**项目开发总结报告**

|  |  |  |
| --- | --- | --- |
| 文件状态：  　[ ]草稿  　[ ]正式发布  　[√]正在修改 | 文件标识： | SE2019春-G12-Bubibura |
| 当前版本： | 1.0 |
| 作者： | 汪诗怡，马易安，王淑慧 |
| 完成日期： | 2019-06-15 |

# 版 本 历 史

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 版本/状态 | 作者 | 参与者 | 起止日期 | 备注 |
| 1.0 | 汪诗怡 | 王淑慧，马易安 | 2019/06/11-2019/06/15 | 第一版 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

目录

[版 本 历 史 2](#_Toc11522717)

[1引言 4](#_Toc11522718)

[1.1编写目的 4](#_Toc11522719)

[1.2背景 4](#_Toc11522720)

[1.3定义 4](#_Toc11522721)

[1.4参考资料 4](#_Toc11522722)

[2实际开发结果 4](#_Toc11522723)

[2.1产品 4](#_Toc11522724)

[2.2主要功能和性能 4](#_Toc11522725)

[2.3基本流程 5](#_Toc11522726)

[2.4进度 5](#_Toc11522727)

[2.5费用 5](#_Toc11522728)

[3开发工作评价 5](#_Toc11522729)

[3.1对生产效率的评价 5](#_Toc11522730)

[3.2对产品质量的评价 5](#_Toc11522731)

[3.3对技术方法的评价 6](#_Toc11522732)

[3.4出错原因的分析 6](#_Toc11522733)

[4经验与教训 6](#_Toc11522734)

# 1引言

## 1.1编写目的

写此项目开发总结报告，以方便我们在以后的项目开发中来更好的实施项目的订制开发；让我在今后的项目开发中有更多的有据的资料来规范我们的开发过程和提高我们的开发效率

## 1.2背景

项目名称：Bubibura!

客户：杨枨

开发人员：G12小组

## 1.3定义

列出本文件中用到的专门术语的定义和外文首字母组词的原词组。

## 1.4参考资料

张海藩 牟永敏《软件工程导论》 北京：清华大学出版社；ISBN：978-7-302-33098-1

# 2实际开发结果

## 2.1产品

Bubibura!安卓手机游戏APP

## 2.2主要功能和性能

玩家的基本操作为

点击一次屏幕进行普通跳跃

再次点击屏幕进行二段跳跃。

详见《需求分析报告》

## 2.3基本流程

用图给出本程序系统的实际的基本的处理流程。

## 2.4进度

基本与原计划符合

详见甘特图

## 2.5费用

原定计划费用与实际支出费用的对比

1. 工时，以人月为单位：小组三人每人4个人月
2. 这种支出：

购买与项目相关的书籍：500左右；

各种软件会员：100左右；

所需开发工具及开发环境：400左右；

团建：500左右；

其他费用（如电费...）：600左右；

# 3开发工作评价

## 3.1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，即每人月生产的行数；
2. 文件的平均生产效率，即每人月生产的千字数；

并列出原订计划数作为对比。

## 3.2对产品质量的评价

说明在测试中检查出来的程序编制中的错误发生率，即每干条指令（或语句）中的错误指令数（或语句数）。如果开发中制订过质量保证计划或配置管理计划，要同这些计划相比较。

## 3.3对技术方法的评价

本游戏主要由Unity2D制作完成：

Unity2D是由Unity Technologies开发的一个让玩家轻松创建诸如二维视频游戏、建筑可视化、实时二维动画等类型互动内容的多平台的综合型游戏开发工具，是一个全面整合的专业游戏引擎。利用Unity2D可以比较轻松的进行游戏编程，建模等操作。

同时Unity2D提供C#编程语言，C#是微软公司发布的一种面向对象的高级程序设计语言。

## 3.4出错原因的分析

由于初学Unity2D、VS2017等工具，对这些工具并不是很熟悉，造成操作不当导致错误。

# 4经验与教训

经验：团队合作尤其重要，需要制定合理的计划，严格遵守计划规则，才能做到事半功倍。

教训：前期动作太拖拉造成后面工作积压，加界面设计耗时耗力。