G16

项目总结



**小组名称： 明德1-518小组**

**项目名称： 模拟三国**

**组长： 王华怿**

**组员： 王仕杰，吴帅毅**

目录

[1. 项目计划 1](#_Toc12109350)

[1.1 编写目的 1](#_Toc12109351)

[1.2 背景分析 1](#_Toc12109352)

[1.3 可行性分析 2](#_Toc12109353)

[1.4 目标功能概述 4](#_Toc12109354)

[1.5 项目进度 5](#_Toc12109355)

[1.6 关键问题 6](#_Toc12109356)

[1.7 计算机系统支持 6](#_Toc12109357)

[1.8 需要用户承担的工作 6](#_Toc12109358)

[1.9 需由外单位提供的条件 6](#_Toc12109359)

[2. 可行性分析 6](#_Toc12109360)

[2.1 编写目的 6](#_Toc12109361)

[2.2 要求 7](#_Toc12109362)

[2.3 目标 7](#_Toc12109363)

[2.4 条件、假定和限制 7](#_Toc12109364)

[2.5 进行可行性研究的方法 8](#_Toc12109365)

[2.6 评价尺度 8](#_Toc12109366)

[2.7 人员 8](#_Toc12109367)

[2.8 技术条件方面的可行性 8](#_Toc12109368)

[2.9 法律方面的可行性 9](#_Toc12109369)

[2.10 使用方面的可行性 9](#_Toc12109370)

[3. 需求分析 9](#_Toc12109371)

[3.1 编写目的 9](#_Toc12109372)

[3.2 项目功能 9](#_Toc12109373)

[3.3 用户类和特征 10](#_Toc12109374)

[3.4 设计和实现限制 10](#_Toc12109375)

[3.5 假设和依赖 10](#_Toc12109376)

[3.6 系统界面及介绍 11](#_Toc12109377)

[3.7 外部接口需求 13](#_Toc12109378)

[3.8 分析模型 14](#_Toc12109379)

[4. 总体设计 16](#_Toc12109380)

[4.1 编写目的 16](#_Toc12109381)

[4.2 设计思路 17](#_Toc12109382)

[4.3 外部接口需求 17](#_Toc12109383)

[4.4 模块设计 18](#_Toc12109384)

[5. 详细设计 19](#_Toc12109385)

[5.1 各模块Jackson图 19](#_Toc12109386)

[5.2 HIPO图 21](#_Toc12109387)

[5.3 PDL（关键算法） 21](#_Toc12109388)

[6. 代码清单 24](#_Toc12109389)

[6.1 编写目的 24](#_Toc12109390)

[6.2 阅读对象 24](#_Toc12109391)

[6.3 代码规范 24](#_Toc12109392)

[6.4 注意事项 25](#_Toc12109393)

[6.5 代码分工 25](#_Toc12109394)

[6.6 详细代码 26](#_Toc12109395)

[7. 测试用例 27](#_Toc12109396)

[7.1 编写目的 27](#_Toc12109397)

[7.2 阅读对象 28](#_Toc12109398)

[7.3 注意事项 28](#_Toc12109399)

[7.4 测试计划 28](#_Toc12109400)

[7.5 测试用例设计 30](#_Toc12109401)

[7.6 集成测试 31](#_Toc12109402)

[7.7 系统测试 31](#_Toc12109403)

[7.8 确认测试 32](#_Toc12109404)

[7.9 测试结果 33](#_Toc12109405)

[8. 项目总结 37](#_Toc12109406)

# 项目计划

## 编写目的

为战略策略游戏爱好者和三国迷提供的一款能在手机上简单操作的回合制策略模拟类游戏。且同时,我们致力于给每个玩家一个简单而又有趣的体验,不会因为许多策略类游戏复杂的游戏机制而放弃这类游戏。一款简单，有趣却也需要动脑的手机游戏。

预期读者：杨枨老师，助教，各位组长。

## 背景分析

软件名称：模拟三国

任务提出者：杨枨

开发者：G16，明德1-518小组

组长王华怿：项目经理、项目技术负责人

组员吴帅毅：开发经理、实施和测试工程师

组员王仕杰：开发工程师、实施和测试工程师

用户：杨枨老师，游戏主要面向战略策略游戏爱好者和三国迷

软件平台：微信小游戏

定义：三国题材，SLG ([Sim](https://www.baidu.com/s?wd=Sim&tn=SE_PcZhidaonwhc_ngpagmjz&rsv_dl=gh_pc_zhidao)ulation Game,模拟游戏)，策略，模拟，经营，战略，回合制，微信小游戏，跨平台，简单易上手。

## 可行性分析

①技术上：

1.2D回合制策略类游戏技术上较为容易实现。

2.同一题材的游戏众多,方便借鉴其中的精华。

3.使用游戏引擎可以大大提高开发的效率

②经济上：

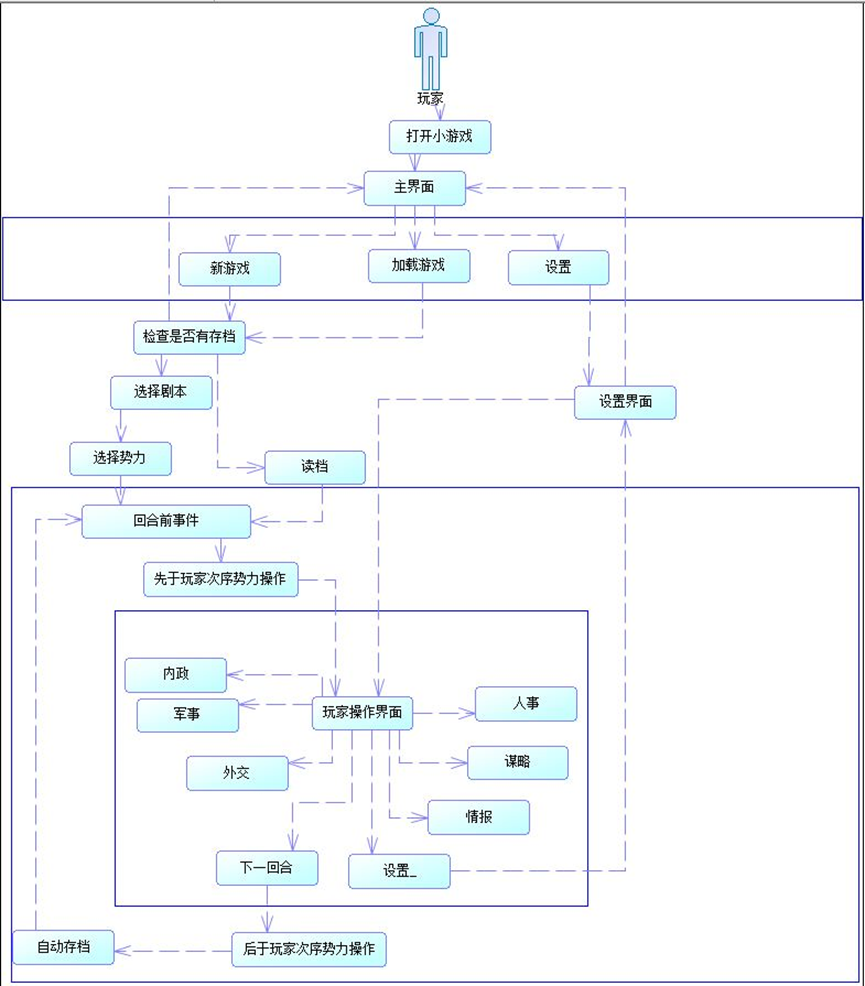
1.免费开源的游戏引擎（Layabox）

2.如果有必要可以使用免费云服务器

③操作上：

PC和微信小程序都是简单易懂的操作

**处理流程和数据流程**



**工作符合**

项目难度不大，但工作量比较大，因为三国设计的人物巨多，需要写出每个人物的特点，但在可接受的范围之内。

**费用支出**

|  |  |  |
| --- | --- | --- |
| **预算名称** | **预算内容** | **预算费用** |
| 劳务费 | 按当前杭州IT行业平均时薪计算，每周4小时，共16周 | 3\*70\*16\*4=13440元 |
| 硬件费 | 开发所用硬件设备，3台中等性能笔记本电脑 | 7000\*3=21000元 |
| 软件费 | 开发所用软件，每台电脑平均500元/年 | 500\*3=1500元 |
| 服务器费 | 微信小程序自带服务器，免费 | 0元 |
| 团建费 | 寝室奶茶、烧烤等 | 100元 |
| 应急费 | 应急资金 | 500元 |
| 总计 | | **32540元** |

**设备**

游戏引擎（Layabox）

配置管理工具（gittub）

**局限性**

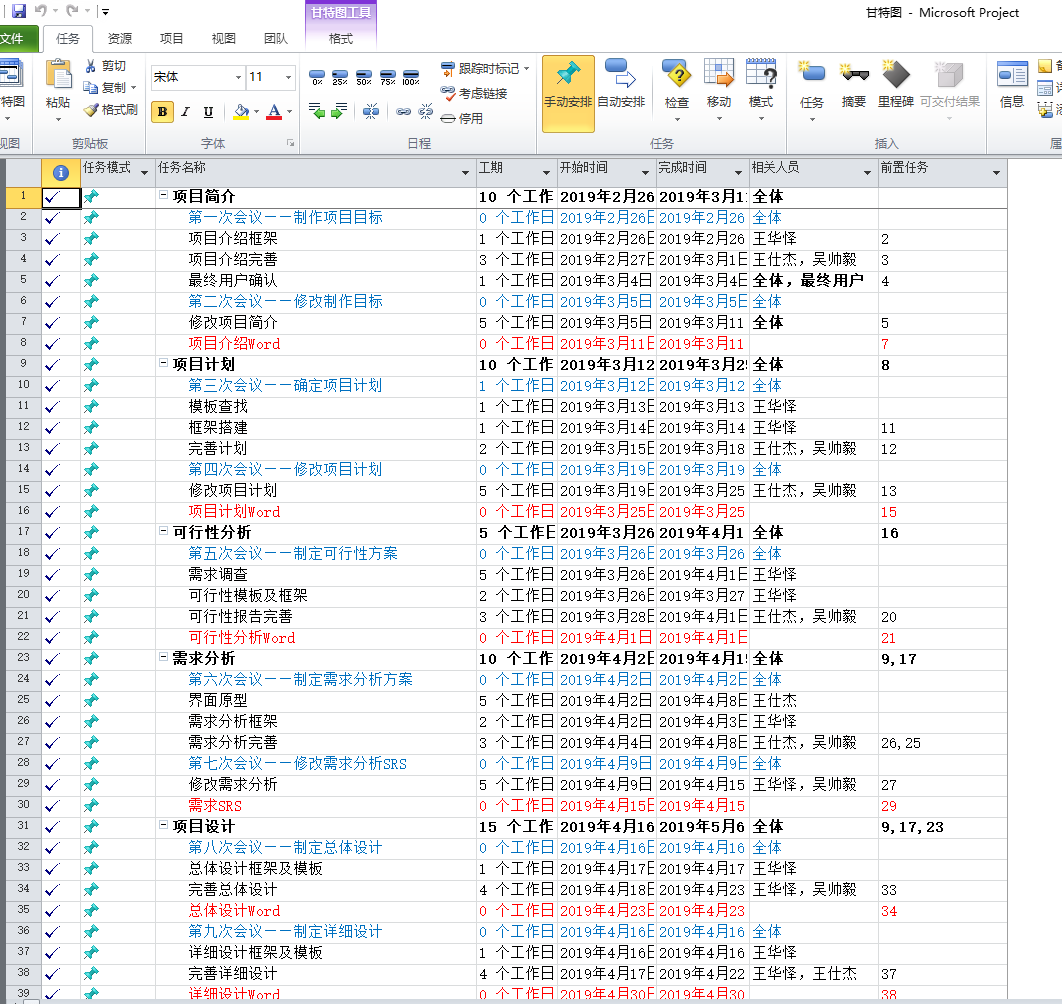
1.各成员对所需要的设备都不太熟悉，需要花费一定时间去学习

2.若算法设计不当，将影响整个游戏的继续

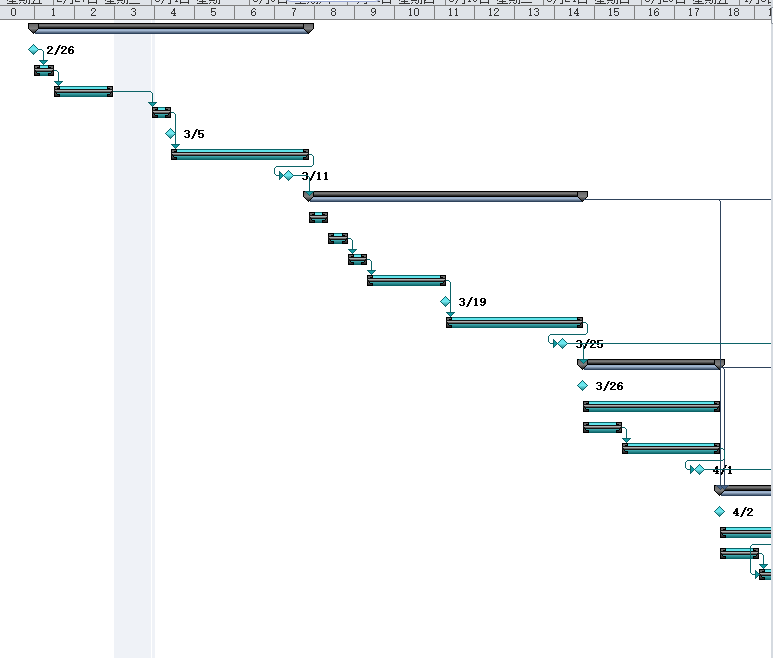
## 目标功能概述

|  |  |  |
| --- | --- | --- |
| **基础目标** | 地图—城池 | 地图只由城池构成 |
| 武将系统 | 每一位武将拥有自己的属性和简介 |
| 内政系统 | 一整套内政操作 |
| 军事系统 | 一整套军事操作 |
| 外交系统 | 一整套外交操作 |
| 计策系统 | 一整套计策操作 |
| 简单的AI | AI对手会简单分析并进行操作 |
| **中极目标** | 可操纵的战斗界面 | 可以对战斗进行操作，影响战局 |
| 头衔-官位系统 | 通过一定条件可以获得官位 |
| 多种胜利条件 | 不仅仅只能通过政府获得胜利，还可以复兴汉室，或者组成联合。 |
| 复杂的资源 | 除军粮，兵器以外，各类生活资源也会作为战略资源 |
| 中等的AI | AI会进行合纵连横等策略 |
| **高级目标** | 武将的特殊能力 | 部分武将可以在一定条件下使用特殊技能 |
| 地图——城池、野外、 | 地图中不仅仅只有城池，还有各类野地，极大扩展战略布置方法和战斗策略 |

## 项目进度







## 关键问题

关键问题：各类胜利条件算法的设计于实现

技术难点：游戏引擎使用，算法设计

若算法设计不当，将影响整个游戏的继续

## 计算机系统支持

计算机机型：PC电脑三台

模拟器：安卓、iOS手机系统模拟器

编译程序：Layabox自带编译器

操作系统：Windows10

## 需要用户承担的工作

触摸屏幕，体验游戏。

## 需由外单位提供的条件

微信团队提供微信小程序平台

# 可行性分析

## 编写目的

　编写可行性报告的目的是为了对此次G16小组的软件工程作业——“模拟三国”游戏的可行性进研究进行报告于总结

预期读者：杨枨老师，助教，各位组长

## 要求

功能：玩家可以在微信小程序上或者PC端、APP进行多人联网对战，游戏可以按照既定的算法结算和正常运行，并且加入排行榜功能，玩家可以与好友的积分进行对比。

性能：中

输出：屏幕展示

输入：触碰屏幕

在安全与保密方面的要求：对玩家的游戏账号进行保密，并且具有存档功能，

同本系统相连接的其他系统：微信

完成期限：第十六周

## 目标

可以进行多人实时对战，按照算法正常结算游戏，游戏可以正常进行，为各个不同三国历史人物设定不同属性，加入排行榜功能。

## 条件、假定和限制

经费、投资反面的来源和限制：寝室小组人员自掏，只能用在本产品的开发中和团队建设。

法律和政策反面的限制：需遵守信息法和微信小程序开发相关条款。

硬件、软件、运行环境和开发环境方面的条件限制：无法在较老机型和系统中运行。

可以利用的信息和资源：网络上有关本类型游戏的信息，Layabox游戏引擎或者unity游戏引擎内的资源。

## 进行可行性研究的方法

调查：使用问卷星在朋友圈中进行问卷调查，在查找各类有关资料。

（详见调研）

## 评价尺度

费用:不高

各项功能的优先次序：基础界面🡪地图人物🡪交互功能🡪高级功能

开发时间的长短：短

使用中的难易程度：不会太难，操作比较简单易懂

## 人员

吴帅毅：开发经理、实施和测试工程师

王华怿：项目经理、项目技术负责人

王仕杰：开发工程师、实施和测试工程师

## 技术条件方面的可行性

1. 2D回合制策略类游戏技术上较为容易实现。

2. 同一题材的游戏众多,方便借鉴其中的精华。

3. 使用游戏引擎可以大大提高开发的效率

## 法律方面的可行性

本软件并非用于商业用途，在专利权与版权等问题上不会构成侵犯，游戏内容也不存在涉黄，暴力血腥等

## 使用方面的可行性

现在人们使用微信的频率是非常高的，而且在前段时间腾讯制作的跳一跳小程序的带领下，人们也经常使用微信小程序玩各种类型的小游戏，使用小程序玩游戏也比较方便，但是策略类的游戏小程序并不存在，并且热爱策略类游戏的玩家的数量也是非常多的。

# 需求分析

## 编写目的

编写目的为理清目标用户所需要的基本功能和制作组想呈现的额外功能。试图从总体架构上给出游戏的轮廓，然后从功能需求，性能需求和其他方面的需求进行详细描述。

## 项目功能

本产品为模拟策略类游戏，可以快速简单地进行游戏，其主要功能有：游戏存档并加载、自己国家的资源管理与升级、与其他国家的对战、好友排行榜功能。

## 用户类和特征

1. 重要用户类

用户代表：杨老师、评审、同时热爱三国题材和喜爱策略游戏玩家用户。

用户需求：制作出一款三国题材的策略游戏，同时游戏能正常稳定运行,且完成度到达预计的目标。

b) 次要用户类

用户代表：喜欢三国或者喜欢战略策略的玩家用户。

用户需求：体现三国味，或者重视战略策略要数。

c）非目标用户类

用户代表：既不喜欢三国，也不喜欢战略游戏的玩家用户。

用户需求：简单、有趣。

## 设计和实现限制

A. 游戏有一定数量的玩家，可以在好友排行榜上进行比较

B. 游戏可以按照既定的规则和机制进行

C. 游戏存档与加载需同步

D. 不同的历史人物需有不同的属性，各个历史人物同各个不同的势力需匹配好

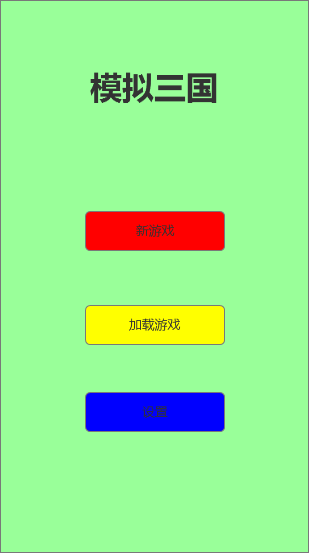
## 假设和依赖

建议开发和运行软件的寿命最短为2年，经费来源为小组G16，使用限制为手机微信小程序，符合法律和政策反面所有条件，运行环境与之前的“运行环境”相同，开发环境由开发方提供，可利用的信息来自问卷星调查数据与互联网上的信息。

1. 只有玩家进行存档，下次加载必须在存档处继续游戏
2. 玩家成功结算后，总分记入排行榜，其好友可以同时看见
3. 玩家在获得更好的分数后，排行榜上的数据应该及时跟新

## 系统界面及介绍

**原始界面**

**修改后的界面**

****

****

**系统功能及介绍**

|  |  |  |
| --- | --- | --- |
| **基础目标** | 地图—城池 | 地图只由城池构成 |
| 武将系统 | 每一位武将拥有自己的属性和简介 |
| 内政系统 | 一整套内政操作 |
| 军事系统 | 一整套军事操作 |
| 外交系统 | 一整套外交操作 |
| 计策系统 | 一整套计策操作 |
| 简单的AI | AI对手会简单分析并进行操作 |
| **中极目标** | 可操纵的战斗界面 | 可以对战斗进行操作，影响战局 |
| 头衔-官位系统 | 通过一定条件可以获得官位 |
| 多种胜利条件 | 不仅仅只能通过政府获得胜利，还可以复兴汉室，或者组成联合。 |
| 复杂的资源 | 除军粮，兵器以外，各类生活资源也会作为战略资源 |
| 中等的AI | AI会进行合纵连横等策略 |
| **高级目标** | 武将的特殊能力 | 部分武将可以在一定条件下使用特殊技能 |
| 地图——城池、野外、 | 地图中不仅仅只有城池，还有各类野地，极大扩展战略布置方法和战斗策略 |

## 外部接口需求

**用户界面**

简单干净的界面，让玩家对各个资源的数量的监控十分方便，操作简单方便，玩家可以轻松愉快的游戏，对新手或从未接触过策略类游戏的玩家十分友好。

**硬件接口**

本软件不需要特定的硬件或者硬件接口进行支撑，只需要可以运行微信的手机便可以运行这个游戏

**软件接口**

运行于微信，安卓或ios12.0以下版本都可以支持

**故障处理**

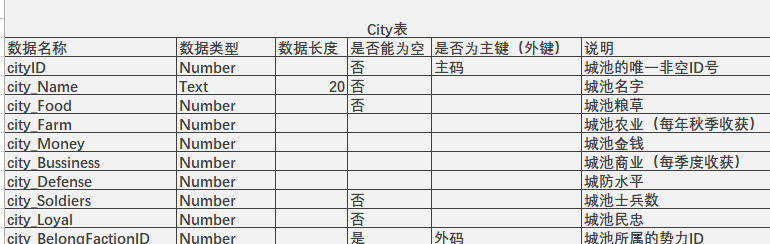
正常使用时不应出错，若运行时遇到不可恢复的系统错误也必须保证数据库的完好无损。

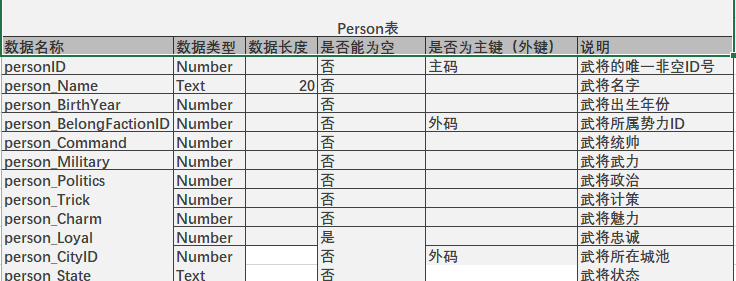
## 分析模型

**系统流程图**



**数据字典**

****

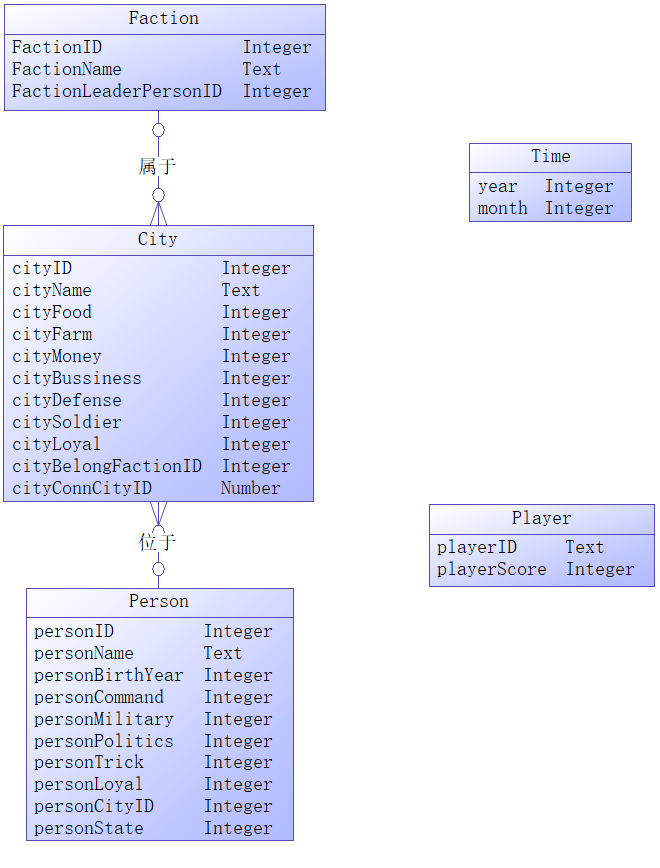
****

****

****

****

**ER图**

****

**数据流程图**

****

# 总体设计

## 编写目的

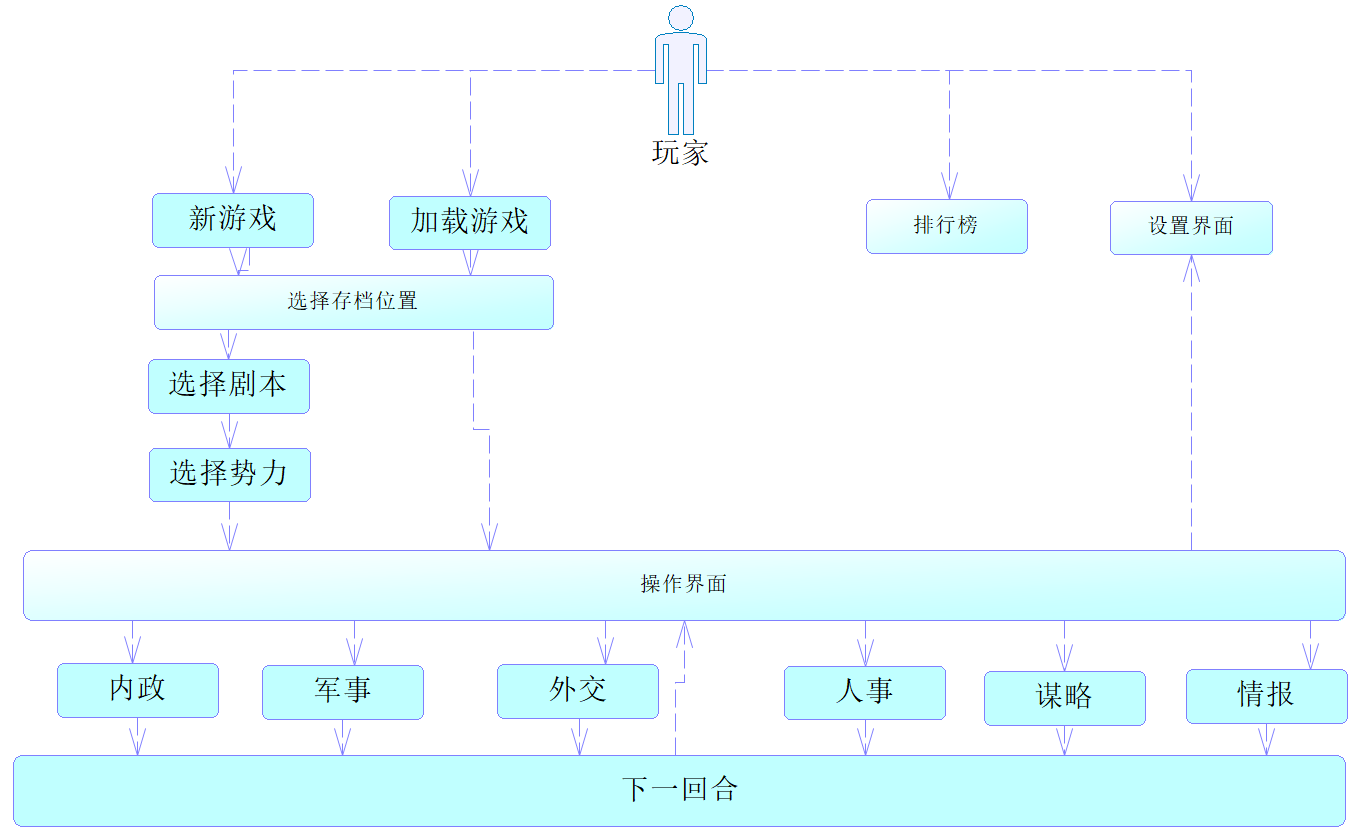
**本阶段要在系统需求的基础上，对项目做进一步的总体说明，它主要解决一下三方面的问题：**

**A．确定软件的结构---有哪几个模块组成,调用关系，各模块间的接口。**

**B．硬件端口分配设计---确定IO端口、和外设之间的资源分配。**

**C．文档的书写**

## 设计思路

****

## 外部接口需求

用户接口

|  |  |  |
| --- | --- | --- |
| 命令 | 语法 | 信息回复 |
| 开始游戏 | Click事件，点击 | 进入游戏 |
| 查询排行榜 | Click事件，点击 | 出现排行榜内容 |
| 返回 | Click事件，点击 | 返回上级菜单或主界面 |
| 设置 | Click事件，点击 | 对音量等进行设置 |
| 游戏内各类操作命令 | Click事件，点击 | 游戏内功能选择 |

**硬件接口**

无

**软件接口**

微信小程序

**外部接口**

无

**内部接口**

系统内部与数据库接口为localStorage

客户端通过配置数据源与服务器建立连接

## 模块设计

**各模块概述**

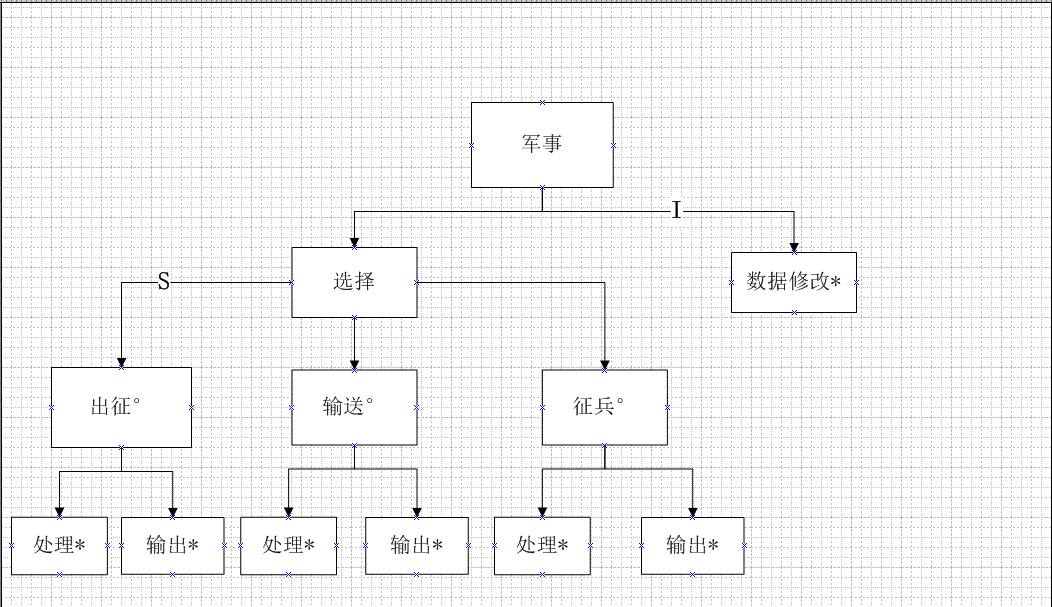


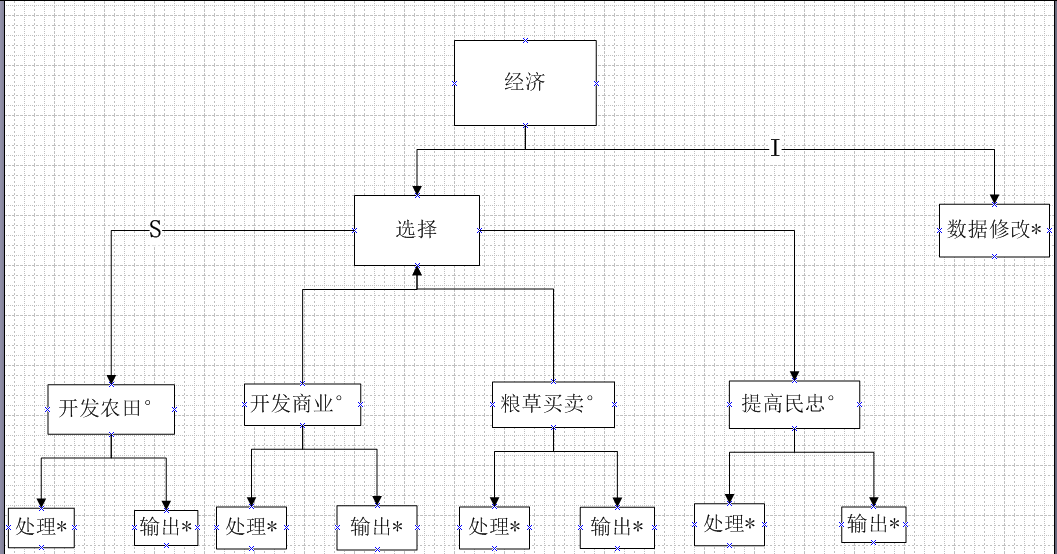
**运行模块的组成**

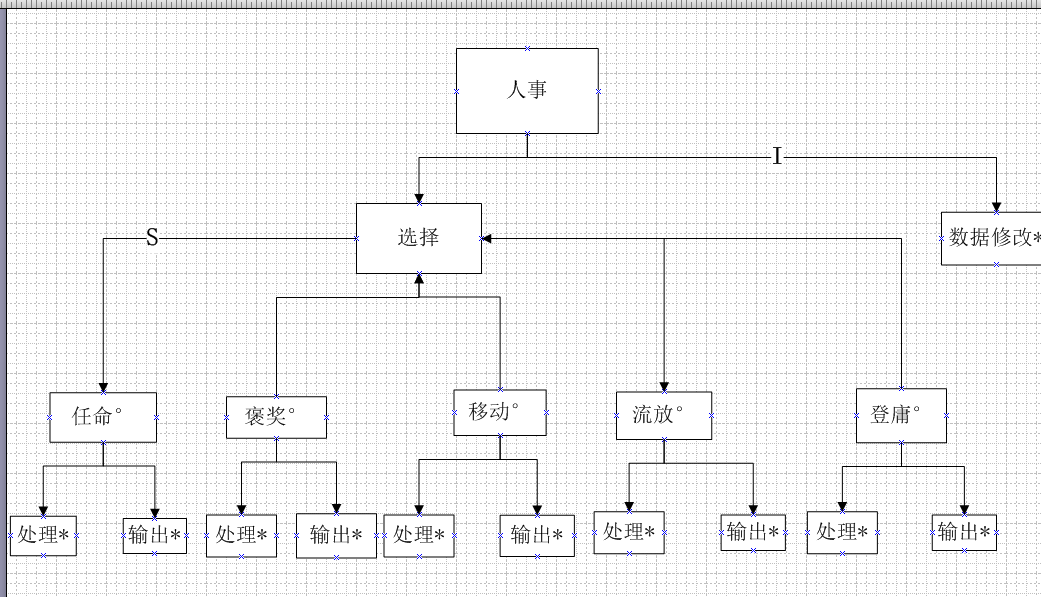
1. 系统分成几个相对独立的模块，但这些模块都进行集中式管理。
2. 分层的模块划程序设计思想，整个系统采用模块化结构设计，作为应用程序有较强的课操作性和扩展性。
3. 合理的数据流设计，在应用系统设计中，相对独立的模块间以数据流相互连接，使各模块间的耦合性较低，方便系统运行，提高系统的安全性。

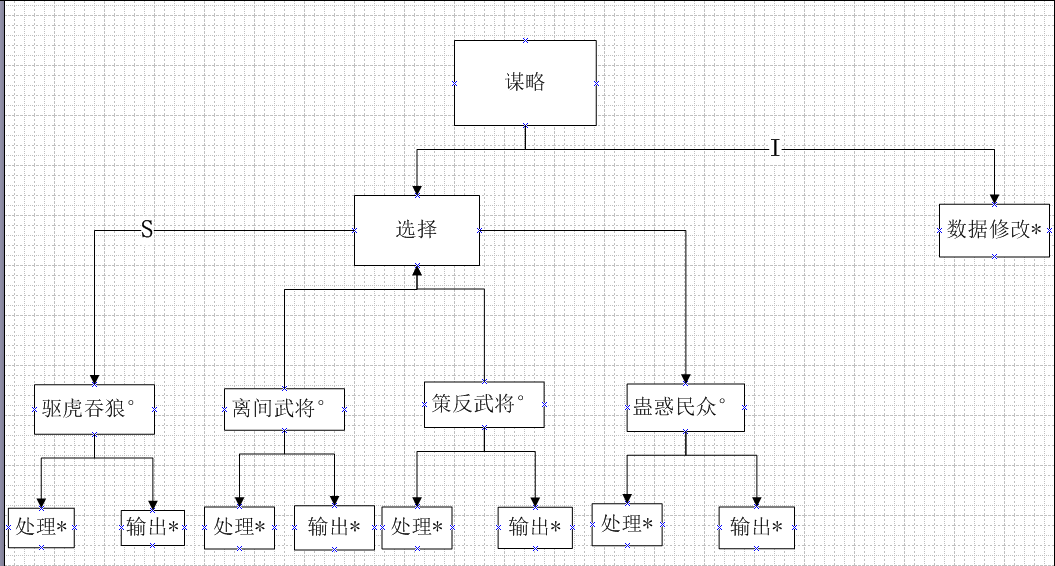
# 详细设计

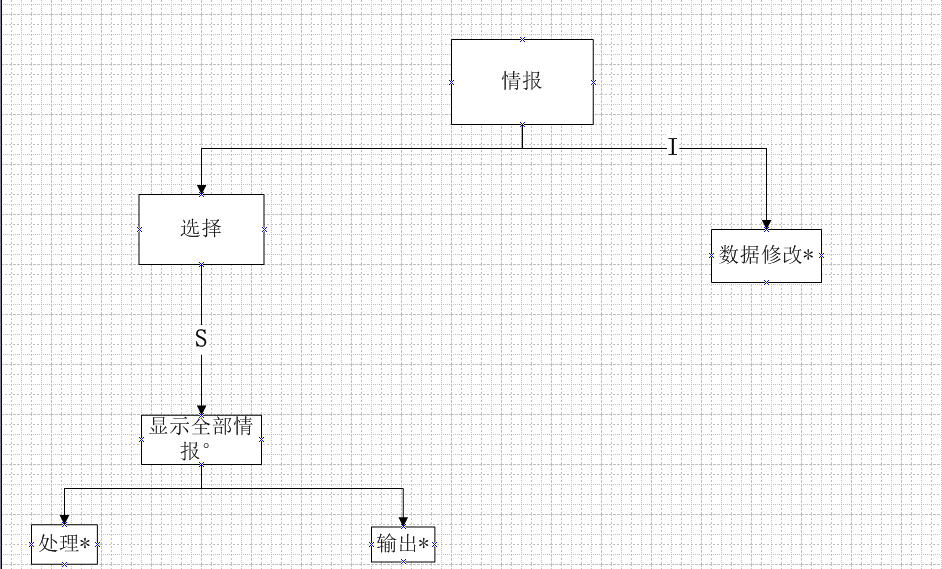
## 各模块Jackson图



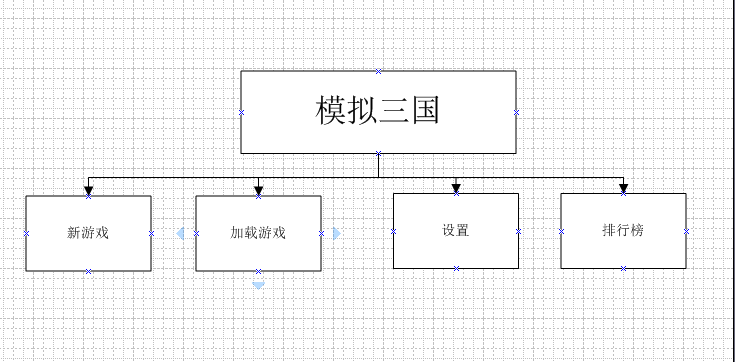


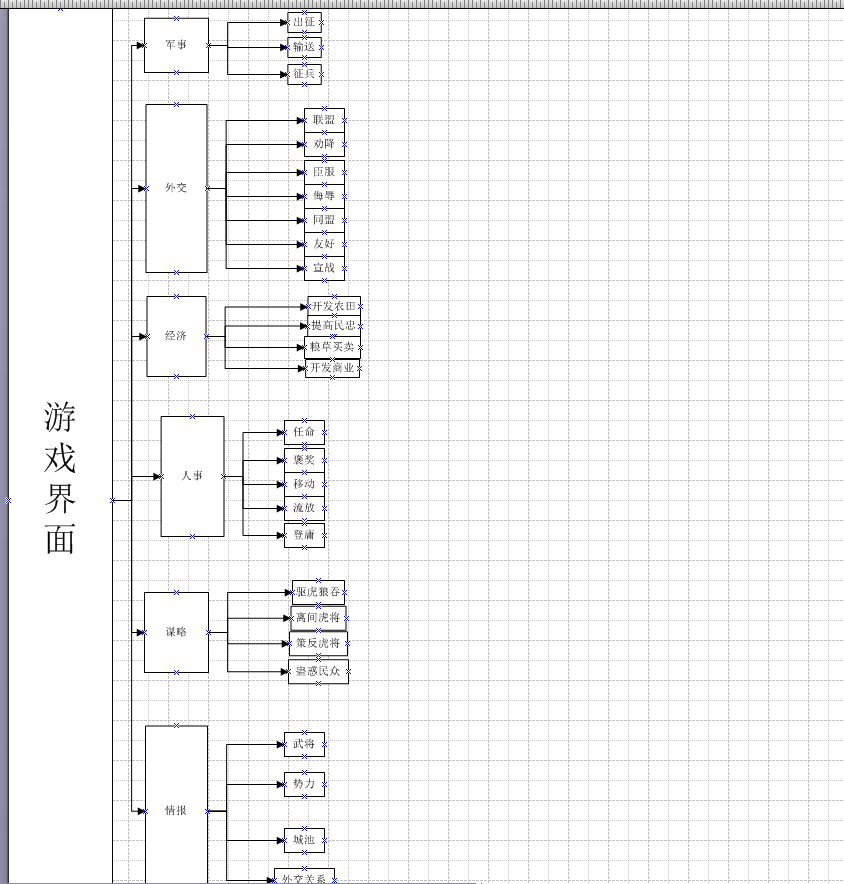






## HIPO图





## PDL（关键算法）

**主界面**

全局变量：用户ID;

Main(){

While（1）{

选择{

1.新游戏()

2.读取存档()

3.排行榜()

4.设置()

}

}

退出

}

**新游戏**

New(){

剧本号=剧本选择（）

势力号=势力选择（）

游戏数据库操作（剧本号，势力号）

进入游戏（）

}

**读取游戏**

Load(){

读取数据从服务器到本地（用户id）//包括所有存档数据

存档号=选择存档（）

加载存档到游戏数据库（存档号）

进入游戏()

}

**排行榜**

Phb(){

读取数据从服务器（）

列出玩家所在（用户ID）

}

**设置**

Setting(){

While(1)

选择{

1.

2.

3.

0.返回Main()

}

}

**游戏**

Game(){

While(1){

for(ID=1;ID<=MAXID;ID++){

if(AI)

AIAUTO();

Else if(Player)

Player();

}

Save();

}

}

# 代码清单

## 编写目的

开发时：帮助内部人员（指设计人员和实现人员）理清代码层次和关系，确认自身的工作完成情况，使工作按照设计完成。

测试时：帮助测试人员确认实际代码是否按照设计来编写。

评审时：为评审人员提供评审对象。

## 阅读对象

设计人员，实现人员，测试人员和评审人员。

## 代码规范

编程语言：JavaScript

命名方式：小驼峰（例：武将年龄 personAge）

注释要求：所有函数须在前面注释该函数作用，所有类须带有详细代码。

## 注意事项

注意1：本文档仅供内部人员查阅！！如给他人借阅，请通报全组成员！

注意2：每次更新时需要在文档第2页更新文档版本，记得改变红色字体颜色。

注意3：当出现代码清单和实际设计矛盾时，应先组内通报并讨论，再决定修改实际代码还是代码清单。

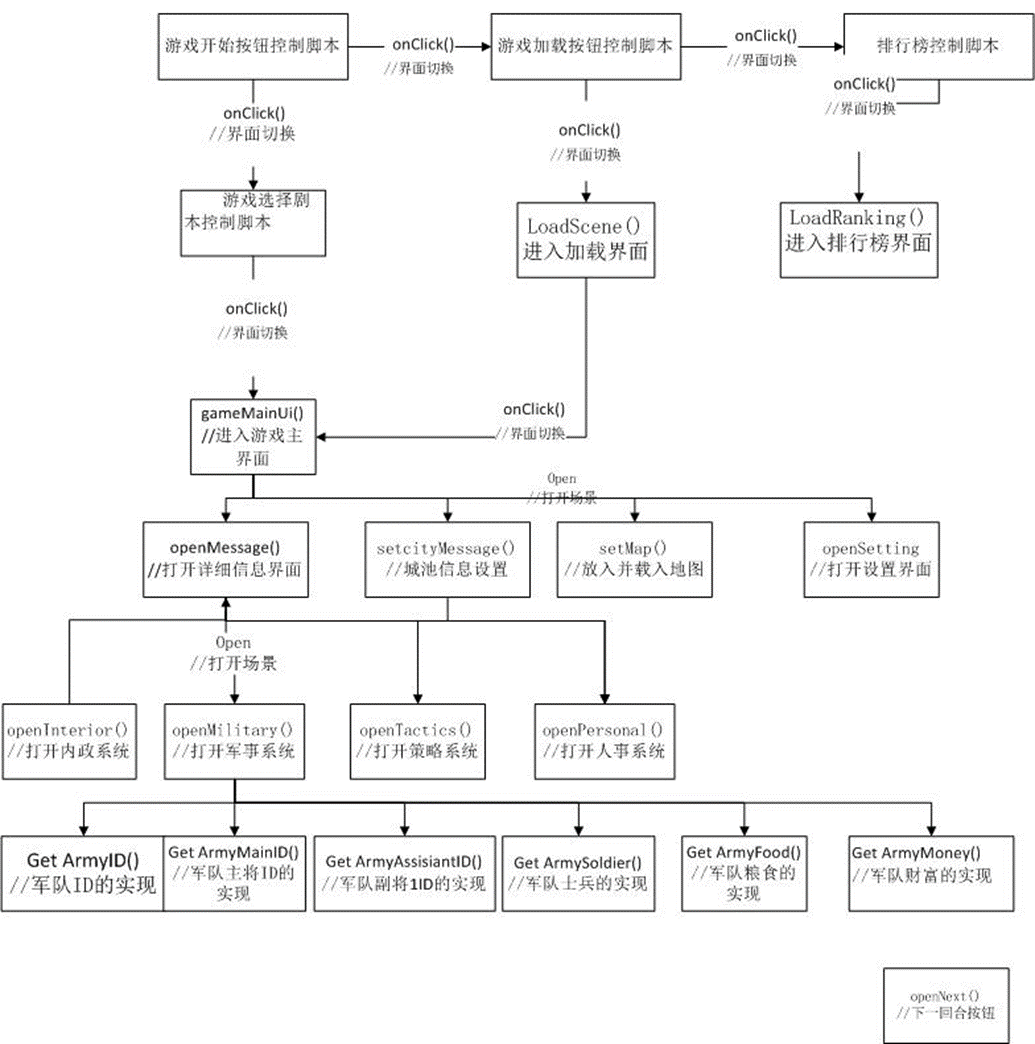
## 代码分工

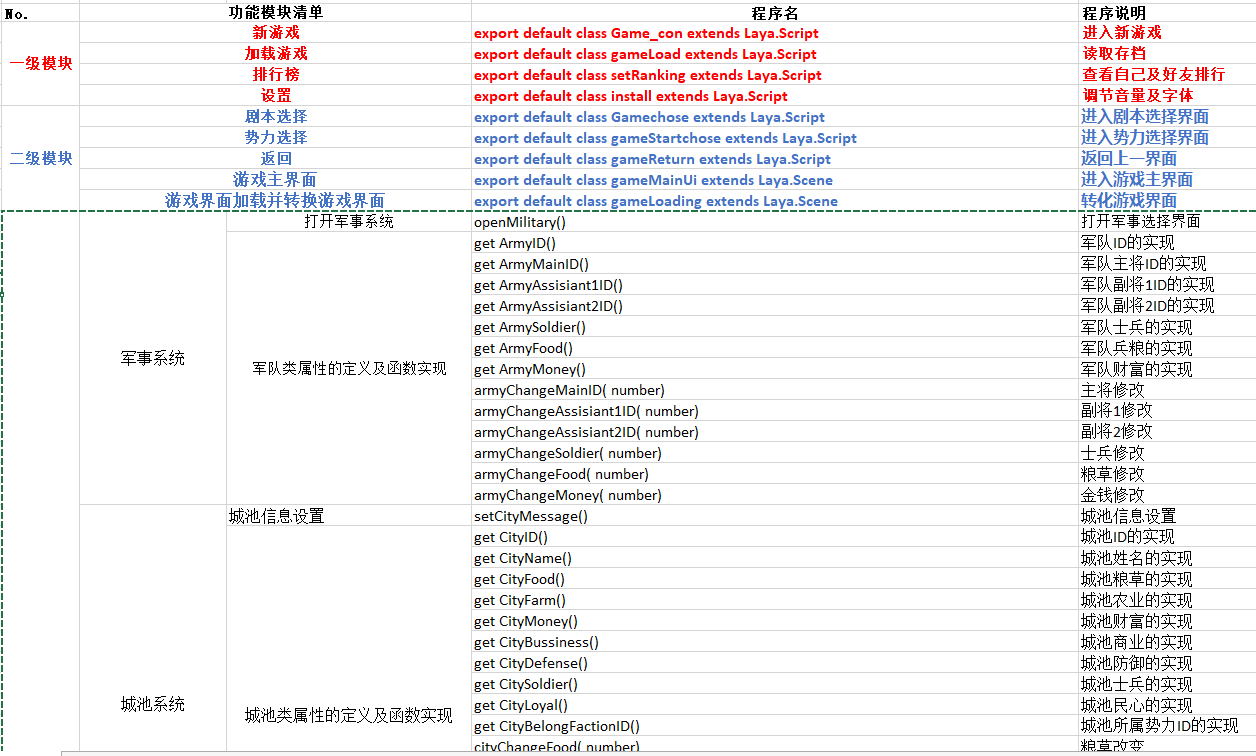
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 一级分类 | 二级分类 | 三级分类 | 设计 | 实现 | 测试 |
| 前端 | UI用户界面设计 | 界面 | 王华怿 | 王仕杰 | 吴帅毅 |
| 王仕杰 |
| 逻辑 | 王华怿 |
| 王仕杰 |
| 美工 | 王华怿 |
| 王仕杰 |
| 美工素材 | 可自制素材 | 王仕杰 |
| 不可自制素材 | 王华怿 |
| 王仕杰 |
| 后端 | 数据库交互 | 本地数据库 | 王华怿 | 吴帅毅 | 王仕杰 |
| 吴帅毅 |
| 云服务器数据库 | 王华怿 |
|
| 王华怿 |
| 游戏运行逻辑 | 前后端对接 | 王仕杰 | 王仕杰 | 王华怿 |
| 吴帅毅 | 王仕杰 |

## 详细代码

**代码关系简图**

（详细代码请看代码清单文案）







# 测试用例

## 编写目的

测试时，帮助测试人员进行测试。

评审时，为评审人员提供评审对象。

## 阅读对象

设计人员，测试人员和评审人员。

## 注意事项

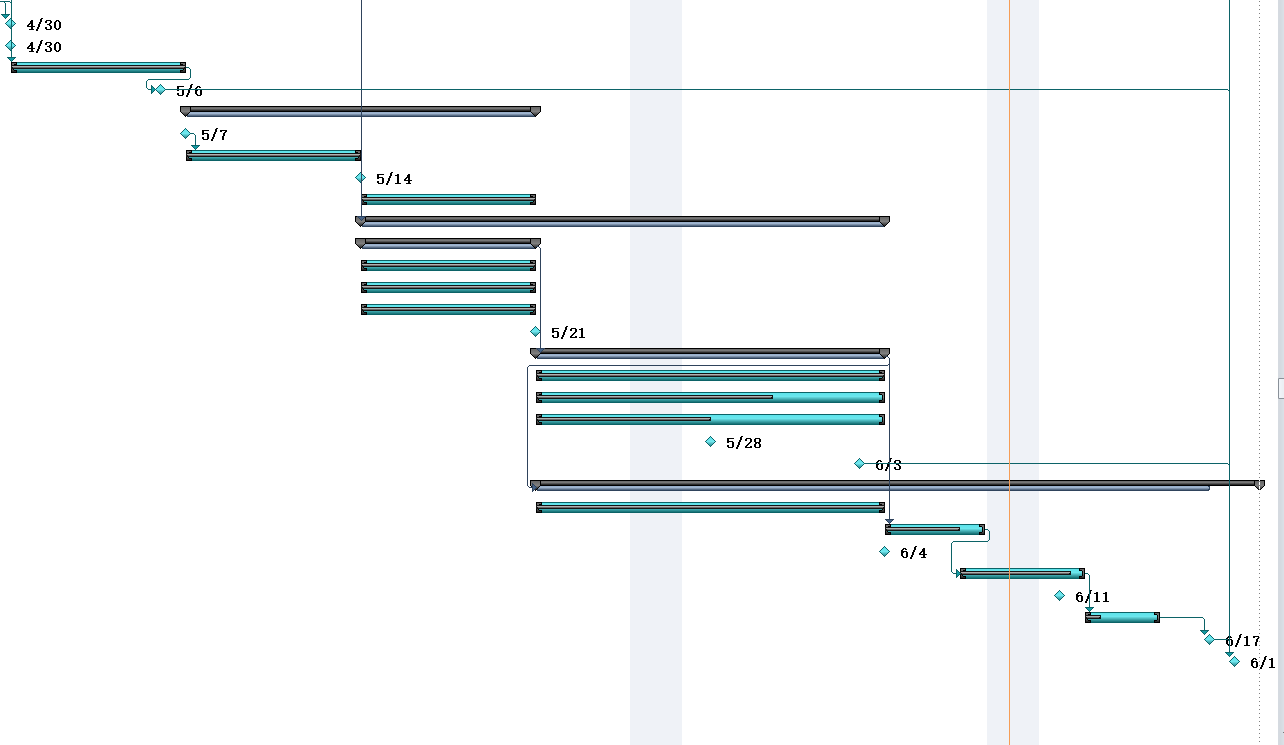
测试需要按照测试用例和相关方法范例，不可以盲目测试。

## 测试计划

**进度安排**

****

****

****

起始时间：5月21日。

结束时间：6月18日。

实际时长：21天。

**测试范围**

由于本软件规模较小，因此所有UI界面和游戏数据都需要进行测试。

**测试方法**

静态测试方法：代码审查，代码走查和静态分析。在下面第4大章会报告静态测试的结果。

动态测试方法：采用白盒测试和黑盒测试。

**测试工具**

测试工具名称：DebugPanel工具、LayaAir DebugTool调试工具。

工具简介：由于本游戏完全基于laya开发，因此laya自带的调试工具与本项目最为适配。这两个工具可以帮助开发者轻松调试并理清前端的关系和后端的数据，提高测试效率。



## 测试用例设计

**单元测试**

参照文件：详细设计文档

测试技术：白盒测试技术

测试说明：测试设计员需要按照详细设计的模块设计，划分相对应的模块测试用例。并且以白盒测试作为主要测试技术，对每个功能模块当中的所有选项进行测试。

//详情请看G16小组测试用例

**各模块测试设计：**

**军事**

设计思路：边缘值测试（0，1，25536），出错处理通路，重要执行通路。

逻辑覆盖：判断/条件覆盖。

**谋略**

设计思路：边缘值测试（0，1，25536），出错处理通路，重要执行通路。

逻辑覆盖：判断/条件覆盖。

**内政**

设计思路：边缘值测试（0，1，25536），出错处理通路，重要执行通路。

逻辑覆盖：判断/条件覆盖。

**人事**

**设计思路：边缘值测试（0，1，25536），出错处理通路，重要执行通路。**

**逻辑覆盖：判断/条件覆盖。**

## 集成测试

参照文件：总体设计文档

测试方法：渐增式测试方法,自顶向下

具体实现：事实上，最简单且有效的方法就是玩家从第一次进入小游戏开始，新游戏，剧本选择，势力选择，军事，内政，谋略，人事，详细信息，下一回合，退出，读取游戏，再一轮操作后退出，查看排行榜。既满足自顶向下，也满足渐增式的要求。

//详情请看G16小组测试用例

## 系统测试

|  |  |  |
| --- | --- | --- |
| 测试编号 | 测试名称 | 测试内容 |
| 1 | 功能测试 | 因与下方确认测试的功能检验相同，不与测试。 |
| 2 | 健壮性测试 | 1. 网络发生波动时，数据是否能够存储在本地和云端 2. 机器发生卡顿时，能否保证数据不出错 3. 微信发生闪退等程序问题时，数据是否能够存储在本地和云端 |
| 3 | 恢复测试 | 是否可以每回合自动保存数据 |
| 4 | 安全测试 | 因与下方确认测试的安全检验相同，不予测试。 |
| 5 | 压力测试 | 使用微信开发者工具进行测试 |

## 确认测试

参照文件：需求分析文档，游戏设计目标文档

参与人员：测试人员、预期用户

测试方法：黑盒测试方法

|  |  |  |
| --- | --- | --- |
| 编号 | 测试名称 | 测试内容 |
| 1 | 功能检验 | 确认游戏的功能是否满足用户的需求 |
| 2 | 性能检验 | 确认游戏的性能用户是否满意 |
| 3 | UI检验 | 确认游戏的UI界面用户是否满意 |
| 4 | 安全性检验 | 确认游戏的安全性用户是否满意 |
| 5 | 兼容性检验 | 确认游戏的兼容性用户是否满意 |
| 6 | 可维护性检验 | 确认游戏的可维护性用户是否满意 |
| 7 | 游戏性校验 | 确认游戏的游戏性用户是否满意 |
| 8 | 软件配置复查 | 自查软件配置的成分是否齐全，质量符合要求 |

## 测试结果

**单元测试测试结果**

|  |  |  |
| --- | --- | --- |
| 模块编号 | 模块名称 | 测试结果 |
| 1 | 新游戏 | 测试完成，满足设计要求。 |
| 2 | 读取游戏 | 测试完成，能正确读取游戏 |
| 3 | 情报 | 测试完成，满足设计要求。 |
| 4 | 下一回合 | 测试完成，能正确存档。 |
| 5 | 排行榜 | 测试完成，满足设计要求。 |
| 6 | 军事 | 测试完成，能准确进行军事的各项操作 |
| 7 | 内政 | 测试完成，能准确进行内政的各项操作 |
| 8 | 谋略 | 测试完成，能准确进行谋略的各项操作 |
| 9 | 人事 | 测试完成，能准确进行人事的各项操作 |

集成测试结果

|  |  |  |
| --- | --- | --- |
| 编号 | 模块排序 | 测试结果 |
| 1 | 新游-剧本-势力 | 所有情况下，游戏都可以按照选择的存档位置，剧本和势力开始。 |
| 2 | 新游-剧本-势力-军事 | 1.出征的话赢得一方得到敌方所有资源，输的一方资源减半。  2.输送则输送方减少资源，受益方获得资源。  3.征兵则是增加兵力 |
| 3 | 新游-剧本-势力-内政 | 1. 得到军事模块的相应功能 2. 得到人事模块的相应功能   得到经济模块的相应功能 |
| 4 | 新游-剧本-势力-人事 | 1.任命则某个武将获得相应的头协。  2.褒奖则相应武将获得相应奖励。3.移动则武将离开城池进入其他城池。  4.流放则是民忠低到某个值把相应武将驱逐出城。  5.登庸则是民忠达到最大值相应武将进行登基。 |
| 5 | 新游-剧本-势力-谋略 | 所有结果与测试操作的预期结果相同。  1.降低商业则相应势力金钱降低，民忠减少。  2.降低农业则相应粮草减少，民忠减少。  3.离间武将则，武将之间的信任度降低。  4.驱虎吞狼则，制定两城进行战争，并自己取得渔翁之利。 |
| 6 | 新游-剧本-势力-情报 | 所有结果与测试操作的预期结果相同。   1. 显示武将属性等信息 2. 显示城池等信息 3. 显示相应势力等信息 |
| 7 | 新游-剧本-势力-下一回合 | 成功完成AI行动，回合推演和数据变化 |
| 8 | 旧游-读取-军事 | 有正确的存档并读取，打开相对应的游戏记录 |
| 9 | 旧游-读取-内政 | 有正确的存档并读取，打开相对应的游戏记录 |
| 10 | 旧游-读取-人事 | 有正确的存档并读取，打开相对应的游戏记录 |
| 11 | 旧游-读取-谋略 | 有正确的存档并读取，打开相对应的游戏记录 |
| 12 | 旧游-读取-情报 | 有正确的存档并读取，打开相对应的游戏记录 |

系统测试结果

|  |  |  |
| --- | --- | --- |
| 测试编号 | 测试名称 | 测试结果 |
| 1 | 功能测试 | 与确认测试的功能检验相同，各项功能的实现以准确进行。 |
| 2 | 健壮性测试 | 1. 网络发生波动时，数据能够存储在本地和云端 2. 机器发生卡顿时，能保证数据不出错 3. 微信发生闪退等程序问题时，数据能够存储在本地和云端 |
| 3 | 恢复测试 | 每回合自动保存数据 |
| 4 | 安全测试 | 与下方确认测试的安全检验相同。 |
| 5 | 压力测试 |  |

确认测试结果

|  |  |  |
| --- | --- | --- |
| 编号 | 测试名称 | 测试结果 |
| 1 | 功能检验 | 游戏的功能大致满足用户的需求 |
| 2 | 性能检验 | 游戏的性能用户满意 |
| 3 | UI检验 | 游戏的UI界面用户满意 |
| 4 | 安全性检验 | 游戏的安全性用户满意 |
| 5 | 兼容性检验 | 游戏的兼容性用户满意 |
| 6 | 可维护性检验 | 游戏的可维护性用户满意 |
| 7 | 游戏性校验 | 游戏的游戏性用户满意 |
| 8 | 软件配置复查 | 自查软件配置的成分齐全，质量符合要求 |

# 项目总结

**需求分析阶段:** 1.需求人员分析需求，应该考虑到用户的真实使用场景，如果需求不切合实际，这样会导致过多的需求变更。设计开发人员应该对需求持以怀疑态度，需求三问，问清楚需求来源，问清楚需求内容，问清楚需求使用场景，如果需求人员无法解答，请需求人员找客户找客户，当然这不是针对需求细节，而是一些大方向以及需求要点。

**设计阶段:** 1.设计分为数据库设计和功能详细设计，数据库设计为了让每个人能会设计数据库，所以需要开发设计自己模块，但是开发自己设计的怎么样，谁都不知道，有些可能开发过程中暴露出来，有的甚至到项目后期或项目上线，后续数据库更改引发的工作量和问题都是很多。所以为了设计好数据库，首先要遵循数据库设计规范，其次有公共模块和同组其他人员去审核db设计，审核有明显问题，就得去更改，保证数据库设计尽量完整、稳定，后续尽可能不出现大的改动。2.功能详细设计：首先功能设计开发自己有个设计蓝图和需求理解，这个需要和设计对需求人员进行反串讲，避免出现需求背离现象。详细功能设计完成后需要交给用户审核。

**开发阶段：**

1.这个阶段存在问题是最多的，一千个人眼中有一千个哈姆雷特，每个人实现代码的方式和思想都是不一样的。

2.先说代码规范，代码的缩进方式、和对齐方式没有一个统一的格式，导致代码结构层次不清晰

3.有的java、js代码全篇没有一行注释，自己写的代码不光是给自己看的，其他人也会看，就算自己看懂，时间长了也不知道写的什么鬼。

4.还有类似于超长方法的，一个方法随便几百行代码（再加上没有注释），出了问题很难定位，让阅读者很难理解。

5.代码模块缺乏提取共性意识，有的地方每增加或者改动一个小点，都会增加一大块代码块，最终会导致整个业务类或者脚本过度复杂，给后期维护和修改带来不少困难。

6.还有一点，实现功能不要为了实现而实现，要求要做1你就做1，殊不知根本不符合需求。要想着这个功能是怎么来的，这个功能要解决什么问题，要实现举一反三。

**测试阶段：**

1.测试人员应该尽可能多的覆盖测试用例，不能为了测试功能而测试，举一反三去测试，考虑正常情况外，一定要考虑异常情况。

2.测试提问题时，一些模棱两可的问题，有些自身感觉是问题，开发说不是，那就找需求或者用户，不能开发说不是问题就不是问题。

3.测试尽可能的将测试环境模拟正式环境，有些能做到的就做到，做不到的就申请，或者从现场导出数据，将测试环境当成正式环境使用。总之为了尽可能早的将问题暴露出来。

4.测试环境不符合正式环境要求的，尽可能找开发模拟，将测试流程的终端和调试现场紧密结合起来。

**上线前：**

1.系统以及功能是做给用户的，对开发测试而言 ，用户就是需求人员。一定要得到需求人员的认可才可以上线，所以就需要需求人员将所有功能和流程全部走完，确保所做的东西是客户需要的。

2.鉴于开发人员和需求人员代沟比较大，需求人员一定要理解开发人员实现想法，请需求人员一定要有耐心去走完全流程，一些小的问题先记下来，可以不用立刻改，类似于流程或者阻塞问题必须改。

**上线运行时：**

1.现场人员一定要分清楚哪些是紧急问题、bug、优化、新需求等，这样会给开发人员留给更多时间解决紧急问题，处理问题有个优先级。

2.现场人员一定遇到问题不要谎，是问题就接反馈，不是问题新需求就一定不要接。

3.对于客户现场反馈的问题，不能用开发的方式回答用户，尽可能贴近用户的想法满足用户，或者让用户等待立刻修复，或者说用户操作失误。给用户提示一个正确的思路。

**个人总结**

**王华怿个人总结：**我在为期一个学期的软件工程课程当中，担任G16小组长，顺利带领组员完成绝大部分任务，通过之前的评审并达到最终评审。虽然过程是十分的坎坷，但是我和我的组员依然不折不饶地坚持了下来，并完成大部分的需求。我很感激我的队员的付出和努力，也十分认可他们的表现。第一次担任PM，说实话，经验相当不足。在整个课程过程当中犯了很多的失误，例如：对工作量的估计，对分工不够明细，没有使用正确Git，文档更新混乱等。当我相信，在今后我定能吸取教训，做得更好！

**吴帅毅个人总结：**自3月份开始，我加入了G16小组，到现在将近四个月的时间里，我从项目设计，可行性分析，在到需求规格说明，详细设计，总体设计，最后到项目测试，项目总结。从开始到项目结束，一步步的走了过来。在这个过程中我遇到了很多困难，像最开始的不会用很多软件，不怎么会写文档，到最后不怎么会用JS语言。我也犯了很多错误，比如各种图画的不够准确，各种文档写的不达标，各种PPT做的不怎么好。但我很幸运有组员帮我一起修改，一起进步。所以我在这里会为自己之前的错误做总结，会在以后的项目里加以改进。最后在这个过程中我懂得了团队合作的重要性，这些工作都不是一个人能完成过来了，我也懂得了分工的重要性，每个人都应该发挥自己的长处，这样会加快做事效率也可以减少犯错。

**王仕杰个人总结：**在学期初的课上，我加入了G16小组，在一个学期的学习中，我们按照老师的要求和课程的要求，学习了如何系统规范的制作一个软件，不仅仅是课本上的知识，还有许多课本上没有的东西，例如我们自己选择的LayaBox游戏引擎，虽然说并没有达到精通他的程度，但是现在还是可以比较好的去使用他了，还有许多许多需要实践才会知道，想要做出一个软件并不仅仅是编程这么简单。但是在这个过程中，我也做错了许多事，文档使用没有规范，不是按照标准文档，导致最后其他组员需要重新查找使用规范的文档，浪费了时间，所以，在之后的工作中还是每一步都要按照标准进行。还有在实现阶段解决问题能力还是不行，需要百度或者组员的帮助，所以在个人编程能力上还是需要提升。

**小组总结**

**编程语言可以选择更好**

JS语言是弱类型语言，所谓万物皆是var，在较小项目中比较方便，但是到了大项目就是噩梦了。而且对于习惯于使用强类型语言的我们来说，数值运算的时候都需要进行麻烦的转换。

**游戏类型与小游戏的理念不是很符合**

微信小游戏的理念基本上“打开即玩，快速游戏”，当我们一开始的设计并未按照这一理念进行设计，因此游戏相对繁琐。再后来进行修改耗费了大量的时间和精力。

**Laya游戏引擎的开发者生态不好，文档也不详细**

一开始选择Laya，是因为被它的虚假宣传，相对简洁的界面和齐全的功能所骗。但是，事实上，Laya的帮助文档相当简陋，而且缺少大部分的demo，使用的人少也不愿意分享例子或经验。因此对于初学者来说相当痛苦，开发的大部分时间都是在攻坚难点。

**参考文献**

[1]聂明.游戏开发导论[M].西安电子科技大学出版社,2009.

[2]张海藩.软件工程导论[M].清华大学出版社,1996:1-73.

[3]雷磊.微信小程序开发入门与实践.清华大学出版社,2017.

[4]项目计划说明书

[5]可行性研究报告

[6]需求规格说明书

[7]总体设计说明书

[8]详细设计说明书

[9]代码清单说明文档

[10]测试用例说明文档

[11]国标.计算机软件文档编制规范. GB/T 7358-2007

[12]国标.计算机软件需求规格说明规范. GB/T 9385-2008

[13]国标.计算机软件文档编制规范. GB/T 8567-2006

[14]国标.计算机软件文档编制规范. GB/T 8567-88

[15]国标.计算机软件文档编制规范.GB/T 9386-2008.

[16]国标.计算机软件测试规范..GB/T 15532-2008