

**G16组号：G16**

**代 码 清 单**



课 程 名 称 软件工程基础

项 目 名 称 模拟三国

小 组 名 称 明德1-518

指 导 老 师 杨枨



|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 代码版本号 | 更新内容 | 更新日期 |
| 1 | V0.0.0 | 最初的代码框架 | 2019.5.14 |
| 2 | V0.0.1 | 框架细化 | 2019.5.24 |
| 3 | V0.0.2 | 初版代码清单编写完成 | 2019.5.26 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

注：红色标注为当前文档版本。

目录

[1 引言 2](#_Toc9894940)

[1.1 编写目的 2](#_Toc9894941)

[1.2 阅读对象 2](#_Toc9894942)

[1.3 代码规范 2](#_Toc9894943)

[1.4 注意事项 2](#_Toc9894944)

[1.5 参考模板 2](#_Toc9894945)

[2 系统简述 3](#_Toc9894946)

[2.1 系统流程 3](#_Toc9894947)

[2.2 功能模块 4](#_Toc9894948)

[2.3 代码分工 5](#_Toc9894949)

[2.4 功能关系 6](#_Toc9894950)

[3 详细代码 7](#_Toc9894951)

[3.1 前端部分： 7](#_Toc9894952)

[3.2 后端部分： 17](#_Toc9894953)

# 引言

## 编写目的

1. 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。
2. 测试时，帮助测试人员确认实际代码是否按照设计来编写。
3. 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：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。

## 参考模板

<https://wenku.baidu.com/view/853c9869453610661ed9f463.html>

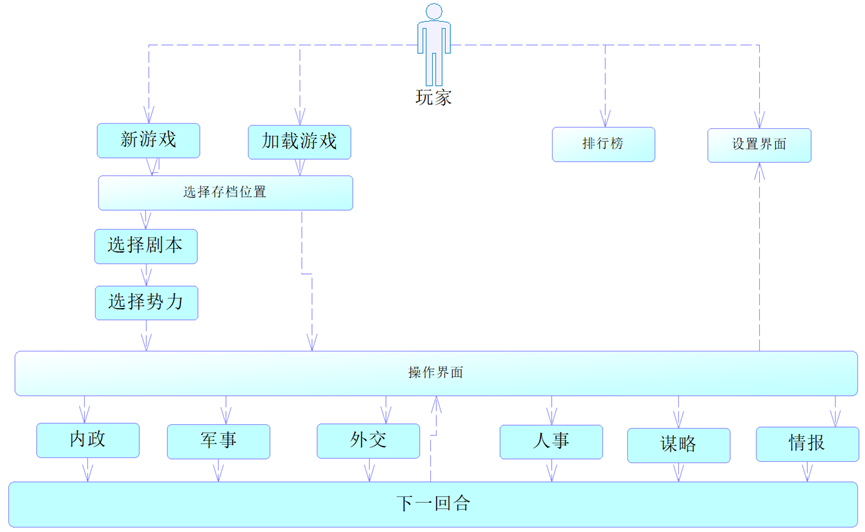
百度文库.源代码清单模板.2018-07-01

<https://wenku.baidu.com/view/fd41a46cf5335a8102d2207d.html>

百度文库.源代码清单模板.2018-06-30

# 系统简述

## 系统流程



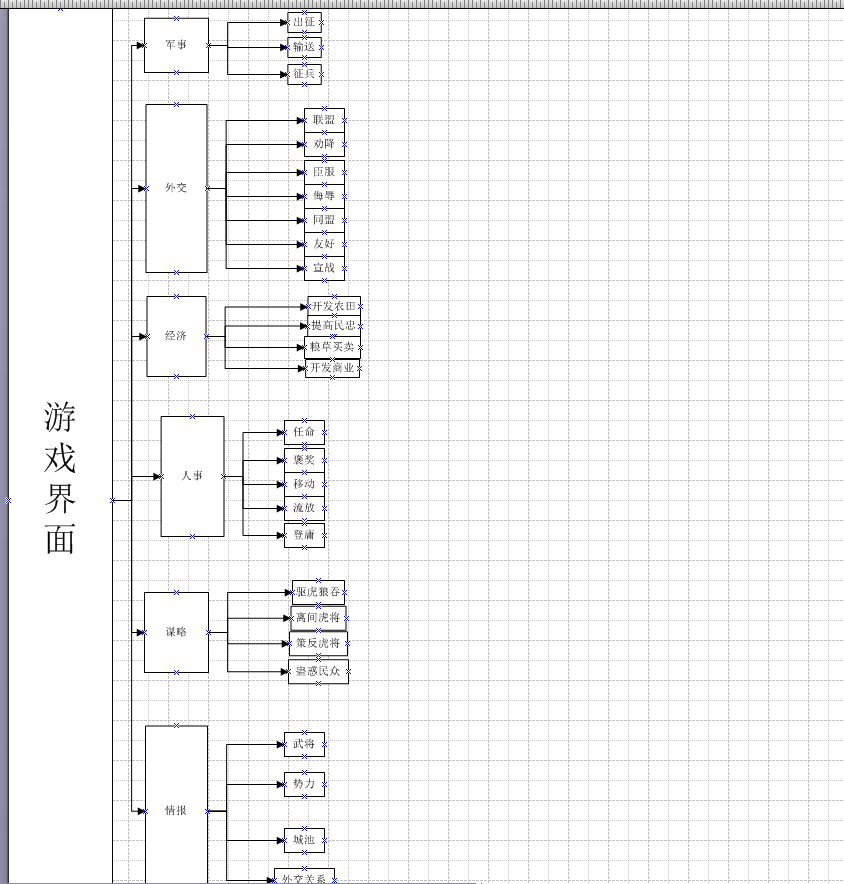
## 功能模块

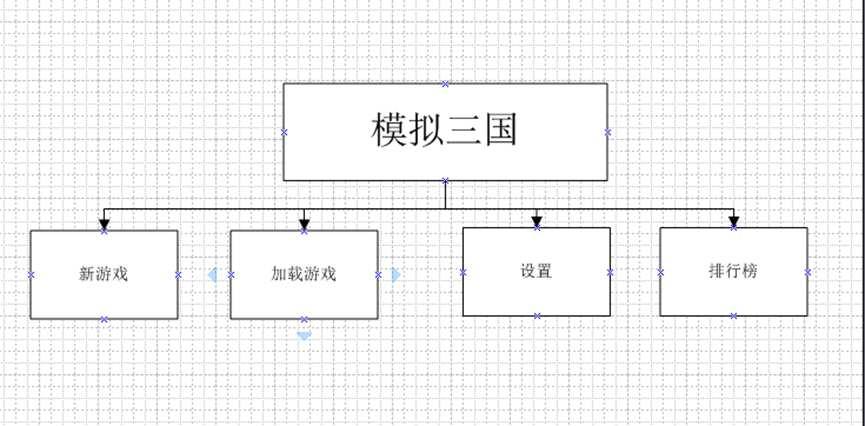
|  |  |  |  |
| --- | --- | --- | --- |
| **一级模块** | **二级模块** | **三级模块** | **模块描述** |
| 新游戏 | 存档选择 | 空白新增 | 在未有存档的地方新增存档 |
| 覆盖原有 | 在已有存档的地方覆盖存档 |
| 剧本选择 | | 选择游玩的剧本 |
| 势力选择 | | 选择游玩的势力 |
| 加载游戏 | 存档选择 | 存档读取 | 读取选择的存档 |
| 存档删除 | 删除选择的存档 |
| 游戏界面 | 军事 | | 出征，输送，征兵 |
| 经济 | | 开发农田，开发商业，粮草买卖，提高民忠 |
| 人事 | | 任命，褒奖，移动，流放，登庸 |
| 谋略 | | 策反武将，蛊惑民众,降低商业，降低农业 |
| 情报 | | 显示总体的情报，例如武将、城池、势力、等 |
| 设置 | | 同下【设置】模块 |
| 下一回合 | | 进行下一回合的推演 |
| 排行榜 | | | 查看本人在好友和全局的排名和分数 |
| 设置 | | | 对音量，字体等进行设置 |

## 代码分工

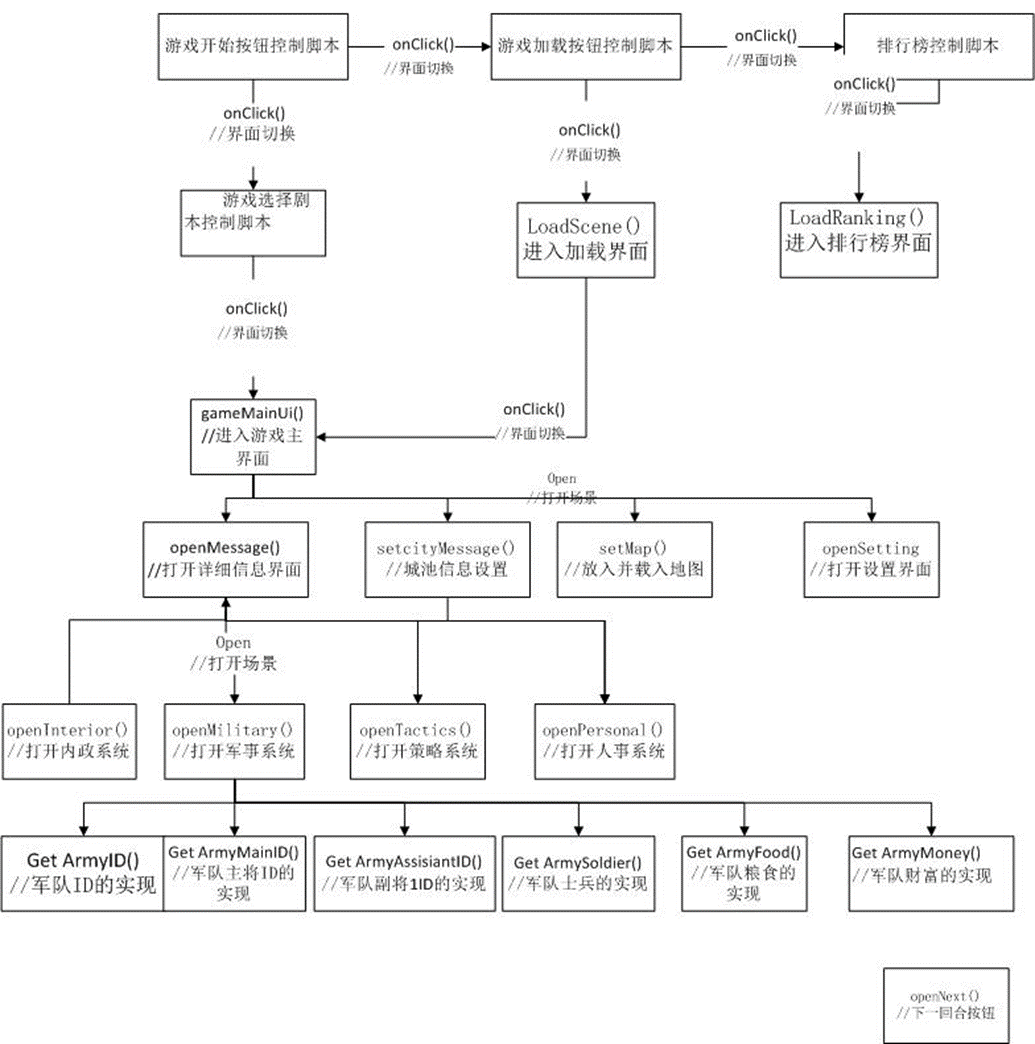
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 一级分类 | 二级分类 | 三级分类 | 设计 | 实现 | 测试 |
| 前端 | UI用户界面 | 界面 | 王华怿 | 王仕杰 | 吴帅毅 |
| 王仕杰 |
| 逻辑 | 王华怿 |
| 王仕杰 |
| 美工 | 王华怿 |
| 王仕杰 |
| 美工素材 | 可自制素材 | 王仕杰 |
| 不可自制素材 | 王华怿 |
| 王仕杰 |
| 后端 | 数据库交互 | 本地数据库 | 王华怿 | 吴帅毅 | 王仕杰 |
| 吴帅毅 |
| 云服务器数据库 | 王华怿 |
|
| 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 |
| 核心运行 | 王华怿 |
|
| 类设计及实现 | 常量 |
|
| 函数 |
|

## 功能关系





# 详细代码



## 前端部分：

游戏开始按钮控制脚本

export default class Game\_con extends Laya.Script {}

游戏选择剧本控制按钮脚本

export default class Gamechose extends Laya.Script {}

游戏加载按钮控制脚本

export default class gameLoad extends Laya.Script {}

游戏势力选择进入游戏控制脚本

export default class gameStartchose extends Laya.Script {}

返回按钮控制脚本

export default class gameReturn extends Laya.Script {}

游戏主界面逻辑代码

export default class gameMainUi extends Laya.Scene {}

openMessage(){}//打开详细信息界面

openSetting(){}//打开设置界面

militaryCon(){}//打开军事系统

interiorCon(){}//打开内政系统

tricksCon(){}//打开策略系统

personnelCon(){}//打开人事系统

openNext(){}//下一回合按钮控制

setCityMessage(){}//城池信息设置

setMessage(){}//详细信息界面控制

setMap(){}//放入地图并控制地图

游戏界面加载并转换游戏界面

export default class gameLoading extends Laya.Scene{}

loadScene(){}//加载进入场景

open(){}//打开场景

close(){}//关闭完成后，调用此方法

}

排行榜设置脚本

export default class setRanking extends Laya.Script{}

## 后端部分：

1. 类常量设计及实现：

军队类属性的定义及实现

class Army{

constructor(armyID,armyMainID,armyAssisiant1ID,armyAssisiant2ID,armySoldier,armyFood,armyMoney){

this.armyID=armyID;

this.armyMainID=armyMainID;

this.armyAssisiant1ID=armyAssisiant1ID;

this.armyAssisiant2ID=armyAssisiant2ID;

this.armySoldier=armySoldier;

this.armyFood=armyFood;

this.armyMoney=armyMoney;

}

} //军队类属性的定义

城池类属性的定义

class City{

constructor(cityID,cityName,cityFood,cityFarm,cityMoney,cityBussiness,cityDefense,citySoldier,cityLoyal,cityBelongFactionID){

this.cityID=cityID;

this.cityName=cityName;

this.cityFood=cityFood;

this.cityFarm=cityFarm;

this.cityMoney=cityMoney;

this.cityBussiness=cityBussiness;

this.cityDefense=cityDefense;

this.citySoldier=citySoldier;

this.cityLoyal=cityLoyal;

this.cityBelongFactionID=cityBelongFactionID;

}

势力类属性的定义

class Faction{

constructor(FactionID,FactionName,FactionLeaderPersonID,FactionCapitalID){

this.FactionID=FactionID;

this.FactionName=FactionName; this.FactionLeaderPersonID=FactionLeaderPersonID;

this.FactionCapitalID=FactionCapitalID;

}

}

武将类属性的定义

class Persons{

constructor(personID,personName,personBirthYear,personCommand,personMilitary,personPolitics,personTrick,personLoyal,personCityID,personState){

this.personID=personID;

this.personName=personName;

this.personBirthYear=personBirthYear;

this.personCommand=personCommand;

this.personMilitary=personMilitary;

this.personPolitics=personPolitics;

this.personTrick=personTrick;

this.personLoyal=personLoyal;

this.personCityID=personCityID;

this.personState=personState;

}

}

对战类属性的定义

class Battle{

export default class Battle

constructor(BattleID,BattleName,BattleAttackArmyID,BattleAttackCityName,BattleDefendArmyID,BattleFieldCityID,BattleResult){

this.BattleID=BattleID;

this.BattleName=BattleName;

this.BattleAttackArmyID=BattleAttackArmyID;

this.BattleDefendArmyID=BattleDefendArmyID;

this.BattleFieldCityID=BattleFieldCityID;

this.BattleResult=BattleResult; this.BattleAttackCityName=BattleAttackCityName;

}

}

时间类属性的定义及实现

class Time {

constructor(year,season){

this.year=year;

this.season=season;

}

}

2各模块函数实现

**军事模块函数的实现**

warOpen(e,index) //出征

solAdd(e,index) //征兵

sendPerOpen(e,index) //输送

**内政模块函数的实现**

armAdd(e,index) //农田开发

bussinessAdd(e,index) //商业开发

foodChange(e,index) //粮草买卖

lovalAdd(e,index) //增加民忠

defenseAdd(e,index) //提高城防

**人事模块函数的实现**

praisel(e,index) //褒奖

DengYong(e,index) //登庸

moveStar(e,index) //移动

banish(e,index) //流放

计策模块函数的实现

pLodecOpen1(e,index) //离间武将

loDecOpen(e,index) //蛊惑民众

defDecPersonOpen() //降低城防

busDecOpen(e,index) //降低商业

farDecOpen(e,index) //降低农业

时间函数的实现

nextRound() //时间信息变化