

# 沙盒游戏应用

# 可行性分析报告



课程名称： 软件工程基础

组 号： SE-G02

专业班级： 软件工程 1602

所在学院： 计算与计算科学学院

报告日期： 2018 年 4 月 16 日

GB/T8567-2006

目录

沙盒游戏应用 1

可行性分析报告 1

目录 2

版本控制 4

1 引言 4

1.1编写目的 4

1.2背景 5

1.2.1 项目名称及开发成员 5

1.2.2可行性研究分析员 5

1.2.3项目的用户 5

1.2.4 项目主要承担部门 5

1.2.5项目建设背景 5

1.3定义 6

2 引用文件 6

3可行性研究的前提 7

3.1要求 7

3.2目标 7

3.3条件、假定和限制 7

3.3.1条件： 7

3.3.2假定： 8

3.3.3限制 8

3.4进行可行性研究的方法 8

3.5评价尺度 8

4对现有系统的研究 9

4.1现有系统概述 9

4.2现有系统的基本功能 9

4.3现有系统的总体系统流程图 12

4.4经济费用 13

4.5现有系统的缺点和局限性 13

5 可选的方案  13

5.1 原有方案的优缺点、局限性及存在的问题 13

5.2 可重用的系统，与要求之间的差距  13

5.3 可选择的系统方案 1  13

5.4 可选择的系统方案 2 13

5.5可选择的系统方案 3 14

5.6 选择最终方案的准则  14

6 所建议的系统  14

6.1 对所建议的系统的说明  14

6.2 数据流程和处理流程  14

6.3 与原系统的比较(若有原系统)  19

6.4 影响(或要求)  20

6.4.1 设备 20

6.4.2 软件  20

6.4.3 运行  20

6.4.4 开发  20

6.4.5 环境  20

6.4.6 经费  20

6.5 局限性  21

7 经济可行性(成本----效益分析)  21

7 经济可行性 21

7.1支出 21

7.1.1基本建设投资 21

7.1.2其他一次性支出 21

7.1.3代码行估计 22

7.2收益 22

8 技术可行性 22

8.1代码语言的选择 22

8.2数据库的选择 24

9操作可行性 25

10 法律可行性 27

11 用户使用可行性 27

12 其他与项目有关的问题 27

12.1项目总结 27

13 注解 28

附录 28

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 版本控制 | | | | | |
| 版本 | 修改日期 | 修改问题 | 跟踪情况 | 修改人 | 审核人 |
| V0.1 | 2018.4.13 | 完成可行性研究报告 | 已跟踪 | 胡方正 | 刘雨霏 |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

# 

# 1 引言

## 1.1编写目的

编写该可行性研究报告的目的在于用最小的代价在尽可能短的时间内确定问题是否能够解决，给出各种可能实施的方案，从技术、操作、经济方面说明并论证所选定实施方案的理由。

## 1.2背景

### 1.2.1 项目名称及开发成员

|  |  |
| --- | --- |
| 项目名称 | Android沙盒游戏应用 |
| 提 出 者 | 刘雨霏 |
| 开 发 者 | 刘雨霏 胡方正 杨智麟 |
| 实现中心 | Android平台 |

### 1.2.2可行性研究分析员

刘雨霏、胡方正、杨智麟

### 1.2.3项目的用户

全年龄段玩家

### 1.2.4 项目主要承担部门

浙江大学城市学院软工、信管16级 G02小组

小组人员：

组长：刘雨霏

组员：胡方正、杨志麟

### 1.2.5项目建设背景

目标用户：

一级用户:对沙盒游戏有浓厚兴趣，经常在游玩沙盒游戏的玩家群体。

二级用户:对沙盒游戏有兴趣，并想接触沙盒游戏的潜在玩家群体。

背景：

现行的沙盒游戏中，游戏模式基本固定，缺少新意。

因此希望通过开发一个游戏模式新颖的沙盒游戏，来进行一些改善。

### 1.3定义

|  |  |
| --- | --- |
| 软件 | 软件是计算机系统中与硬件相互依存的部分，它是包括程序、数据及相关文档的完整集合 |
| 软件工程 | 软件工程是研究和应用如何以系统化的、规范的、可度量的方法去开发、运行和维护软件，即把工程化应用到软件上 |
| 软件生存周期 | 软件生存周期是指软件产品从考虑其概念开始到该软件产品交付使用，直至最终退役为止的整个过程，一般包括计划、分析、设计、实现、测试、 集成、交付、维护等阶段 |
| 软件质量 | 软件质量是软件与明确声明的功能和性能需求、文档中明确描述的开发标准以及任何专业开发的软件产品都应该具有的隐含特征相一致的程度 |
| 软件需求 | 软件需求是指  （1）用户解决问题或达到目标所需的条件或能力；  （2）系统或系统部件要满足合同、标准、规范或其它正式规定文档所需具有的条件或能力；  （3）一种反映上面（1）或（2）所描述的条件或能力的文档说明。 |
| 项目管理 | 项目管理就是通过合理地组织和利用一切可以利用的资源，按照计划 的成本和计划的进度，完成一个计划的目标，它包含团队管理、风险管理、采购管理、流程管理、时间管理、成本管理和质量管理等。 |

### 

# 2 引用文件

* 《软件项目管理》 Rajeev T Shandilya 编著 科学出版社 ISBN

9787302218708

* 软件工程国家标准文档
* 软件工程项目开发文档范例
* 《我所理解的Cocos2d-x》 秦春林 电子工业出版社 ISBN

978-7-121-24625-8

* 《写给大家看的设计书》[美]罗宾·威廉姆斯(RobinWilliams) 人民邮电出版社
* Programming in Lua Roberto Ierusalimschy ISBN 859037985X
* 《数据库系统概念》Abraham Silberschatz Henry F.Korth S.Sudarshan 机械工业出版社 ISBN: 9787111375296
* 《Cocos2d-x实战 Lua卷》关东升 清华大学出版社 ISBN：9787302457305

# 3可行性研究的前提

## 3.1要求

开发一款Android沙盒手机游戏应用。

具体要求：

1. 有较为完整的美术设计和背景音乐
2. 将地图等分为正六边型
3. 游戏中的角色具备2种行动方式：移动、攻击
4. 具备蓝牙联机功能，实现双方玩家同时游戏
5. 游戏胜利条件：

模式1:全歼敌人

模式2:杀死国王

1. 游戏分兵种，不同兵种都有不同的能力。
2. 士气因素：士气高时增加单位的属性值，低时减少一定的属性值

## 3.2目标

给用户提供良好的游戏环境和游戏体验，有一定的创新性、可玩性和吸引力，不至于使得游戏过于单调。

## 3.3条件、假定和限制

### 3.3.1条件：

·开发系统：

Windows 工作环境，Windows 7 虚拟机工具使用环境；

·应用系统：

Dev C++编程软件，Mysql数据库，cocos2D游戏引擎。

### 3.3.2假定：

·开发的软件运行的最短寿命为一年；

·游戏设计符合负载与欲求；

·开发时间与试运行时间预定为一个月；

·对比与现有系统新的形式被接受；

### 3.3.3限制

· 开发人员处于边学习边开发的过程，经验不足与学习时间安排都可能对开发进程会有阻碍；

·系统投入使用的最晚时间:2018年6月10日，系统开发时间紧迫

·设想的软件与能力水平经验差值太大，由于瀑布模型每个阶段输出为抽象的文档，最终可能会有出入。

·服务器配置不是过高。

## 3.4进行可行性研究的方法

1复查系统的规模和目标

2研究目前正在使用的系统

3导出待开发系统的雏形，提供数据流程和处理流程

## 3.5评价尺度

首先功能的完整性，功能完整是评价系统优劣的重要指标，该有的功能必须要有；其次功能要简洁而易用；游戏要有一定的创新点和可玩性，提高用户的游戏体验。

# 4对现有系统的研究

## 4.1现有系统概述

参考系统：《火焰之纹章：烈火之剑》（ファイアーエムブレム 烈火の剣）

开发商：Intelligent Systems

发行日期：（日本）2003年4月25日



## 4.2现有系统的基本功能

剧情完备，将地图切分为小正方形供角色移动，拥有多样的兵种和武器。

艺术设计：有非常完备的美术设计和背景音乐

每个角色有相应的：HP、力量、技术、速度、幸运、物理防御、魔法防御、移动、体格、武器熟练度等属性。

角色成长率：根据每个角色的性格特征、所属职业，会形成一个玩家不可见的角色成长率，用以表示角色每次升级时增加相应属性值的概率。

武器克制关系：剑克斧、斧克枪、枪克剑等克制关系。

武器的属性：（数据来源：<http://www.fireemblem.net/fe/fe7/fe8--data/wuqi_0.htm>）

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| てつの剣 | E | 1 | 5 | 5 | 90 | 0 | 46 |  |
| ほそみの剣 | E | 1 | 2 | 3 | 100 | 5 | 30 |  |
| はがねの剣 | D | 1 | 10 | 8 | 75 | 0 | 30 |  |
| ぎんの剣 | A | 1 | 8 | 13 | 80 | 0 | 20 |  |
| てつの大剣 | D | 1 | 12 | 9 | 70 | 0 | 35 |  |
| はがねの大剣 | C | 1 | 14 | 11 | 65 | 0 | 25 |  |
| ぎんの大剣 | A | 1 | 13 | 14 | 60 | 0 | 15 |  |
| どくの剣 | D | 1 | 6 | 3 | 70 | 0 | 40 | 中毒效果 |
| レイピア | ★ | 1 | 5 | 7 | 95 | 10 | 40 | 对骑兵、重步兵有特效 |
| マーニ·カティ | ★ | 1 | 3 | 8 | 80 | 20 | 45 | 对骑兵、重步兵有特效 |
| ゆうしゃの剣 | B | 1 | 12 | 9 | 75 | 0 | 30 | 两次攻击效果 |
| 倭刀 | D | 1 | 5 | 8 | 75 | 35 | 20 | 必杀系武器 |
| キルソード | C | 1 | 7 | 9 | 75 | 30 | 20 | 必杀系武器 |
| アーマーキラー | D | 1 | 11 | 8 | 80 | 0 | 18 | 对重步兵有特效 |
| ドラゴンキラー | C | 1 | 5 | 7 | 75 | 0 | 20 | 对龙系有特效 |
| ひかりの剣 | C | 1～2 | 9 | 9 | 70 | 0 | 25 | 间接攻击系，附加光魔法 |
| ルーンソード | A | 1～2 | 11 | 12 | 65 | 0 | 15 | 间接攻击系，附加暗魔法 |
| ランスバスター | C | 1 | 9 | 9 | 75 | 5 | 15 | 反克系，枪强斧弱 |
| 長柄刀 | D | 1 | 11 | 6 | 85 | 0 | 18 | 对骑兵有特效 |
| かぜの剣 | B | 1～2 | 9 | 9 | 70 | 0 | 40 | 间接攻击系，附加理魔法 |
| デュランダル | ★ | 1 | 16 | 17 | 90 | 0 | 20 | 『神将器』　烈火の剣　力+5 |
| ソールカティ | ★ | 1 | 14 | 12 | 95 | 25 | 30 | 对龙系有特效 |
| リガルブレイド | S | 1 | 9 | 20 | 85 | 0 | 25 | 追求极限之道的人所使用的剑 |

等级制度：每个角色拥有等级，下级职业为1～20级，上级职业1～20级，下级职业转职后变回Lv1，角色升级时会根据职业的特性，以一定的几率增加相应的属性值。

兵种：佣兵🡪勇者、剑士🡪剑圣、战士🡪勇士、弓箭手🡪狙击手、魔道士🡪贤者、海贼🡪狂战士......等多样兵种，其中🡪左边的职业在达到lv10时可转职为箭头右边的职业

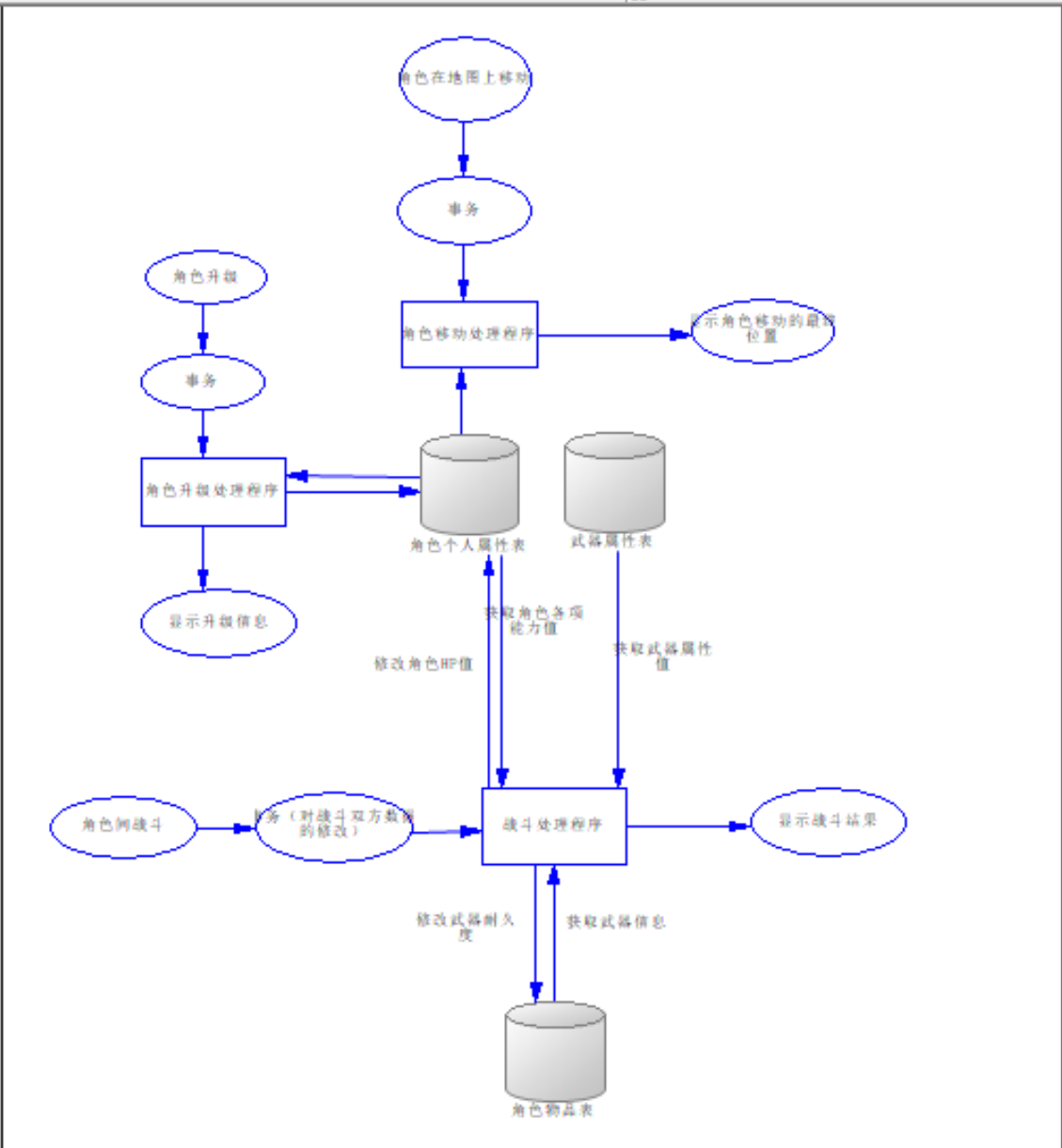
游戏中的计算公式：（数据来源：<http://www.fireemblem.net/fe/fe7/fe8--data/gongshi.htm>）

|  |  |
| --- | --- |
| 攻击力 | 力 + 【武器攻击力 + 3种修正】 3修正：『武器特效』 + ファーラの力效果 + 支援効果修正值 |
| 魔法攻击力 | 魔力 + 【魔道书攻击力 + 3种修正】 3修正：『武器特效』 + ファーラの力效果 + 支援効果修正值 |
| 防御力 | 守 + ニニスの守護的效果 + 支援効果修正值 + 地形効果修正值 |
| 魔法防御力 | 魔防 + せいすい的効果 + Mシールド的効果 + ニニスの守護的効果 + 支援効果修正值 + 地形効果修正值 |
| 物理实际伤害 | 【攻撃力 - 防御力】 『必杀时』X3 |
| 魔法实际伤害 | 【魔法攻撃力 - 魔法防御力】 『必杀时』X3 |
| 攻击速度 | 速 - 【武器、魔道书重量 - 体格】 体格大于武器重量 攻速＝速 |
| 2回攻击 | 攻速 - 对手攻速度≧ 4 |
| 命中率 | 武器、魔道书的命中率 + 技 X 2 + 幸 / 2 + 武器LvS修正值『+5』 + 军师同属性的修正值『+星％』 + 支援効果修正值 + 武器相克修正（+\-15 反克为+\-30） |
| 必殺率 | 武器、魔道书的必杀率 + 技 / 2 + 武器LvS修正值『+5』+ トォルの怒り的効果 + 支援効果修正值 + 职业修正值『＋15』 |
| 回避率 | 攻击速度 X 2 + 运 + 军师同属性修正值『+星％』 + セチの祈り的効果 + 支援効果修正值 + 地形効果修正值 |
| 必杀回避率 | 运+ 支援効果修正值+军师修正值『+星％』 |
| 最终命中率 | 命中率 - 对手的回避率 |
| 最终必杀率 | 必杀率 - 对手的必杀回避率 |
| 杖的命中率 | 30 + 魔力 X 5 + 技 - 对手的魔防 X 5 - 距離 X 2 |

角色间的支援效果：根据故事情节，不同的角色之间有相应的支援效果，战斗时，当相邻角色站在一起时，会对角色的命中率、回避率等造成影响。

每个角色都有物品栏

## 4.3现有系统的总体系统流程图



## 4.4经济费用

## 4.5现有系统的缺点和局限性

现有系统对战斗的影响因素较多，不太适合我们组3个没有开发经验的人来完成。因此现有系统中：支援关系对战斗的影响、特殊道具对战斗的影响等可以不与考虑。

# 5 可选的方案

## 5.1 原有方案的优缺点、局限性及存在的问题

原有方案的相关设定：武器属性、角色成长等与新系统的设计不相符合。同时，限于人数，新系统不能过于复杂。

新系统希望将地图以正六边形的形式分割地图。

## 5.2 可重用的系统，与要求之间的差距

## 5.3 可选择的系统方案 1

平台：WEB

数据库：MongoDB

代码语言：HTML5

## 5.4 可选择的系统方案 2

平台：PC

数据库：SQLserver

代码语言：C#

## 5.5可选择的系统方案 3

平台：Android

数据库：MySQL

代码语言C++/java

## 5.6 选择最终方案的准则

1、所选游戏平台应该具有一定的编写性

2、所选数据库应当免费

3、代码语言应选择游戏引擎Unity支持的语言

# 6 所建议的系统

## 6.1 对所建议的系统的说明

在PC上实现,为玩家提供联机对战功能,使可玩度提高,并方便与玩家进行1对1的对战.

对战系统通过Photon网络引擎,给面对面游戏的玩家带来更多的便利性.

游戏对战模式打破传统的沙盒游戏,采用职业与士气系统,有新颖的游戏规则增强玩家的可玩性与游戏的丰富性.

游戏的原画与音乐都由本组自行制作,避免版权问题与和其他游戏重复的问题.

## 6.2 数据流程和处理流程

数据字典:

-----------------------------------------------------------

玩家信息

名称:玩家信息

描述:用来储存每一个用户与该游戏相关行为需要的数据集合

定义:玩家信息=玩家编号+玩家昵称+玩家战绩

位置:注册

-----------------------------------------------------------

角色信息

名称:角色信息

描述:用来储存每一个角色与该系统相关行为需要的数据集合

定义:角色信息=角色编号+角色名称+角色属性

位置:角色操作

-----------------------------------------------------------

区域信息

名称:地图信息

描述:用来储存每一个区域与该区域包含特性的数据集合

定义:区域信息=区域编号+区域名称+区域属性

位置:地图标识

-----------------------------------------------------------

名称:玩家编号

描述:唯一地标识玩家信息中一个特定玩家的关键域

定义:用户编号=1{数字}5

位置:玩家信息

-----------------------------------------------------------

名称:角色编号

描述: 唯一地标识角色信息中一个特定角色的关键域

定义:角色编号=1{数字}5

位置:角色信息

-----------------------------------------------------------

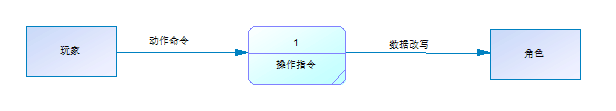
名称:区域编号

描述: 唯一地标识区域信息中一个特定区域的关键域

定义:订单编号:1{数字}5

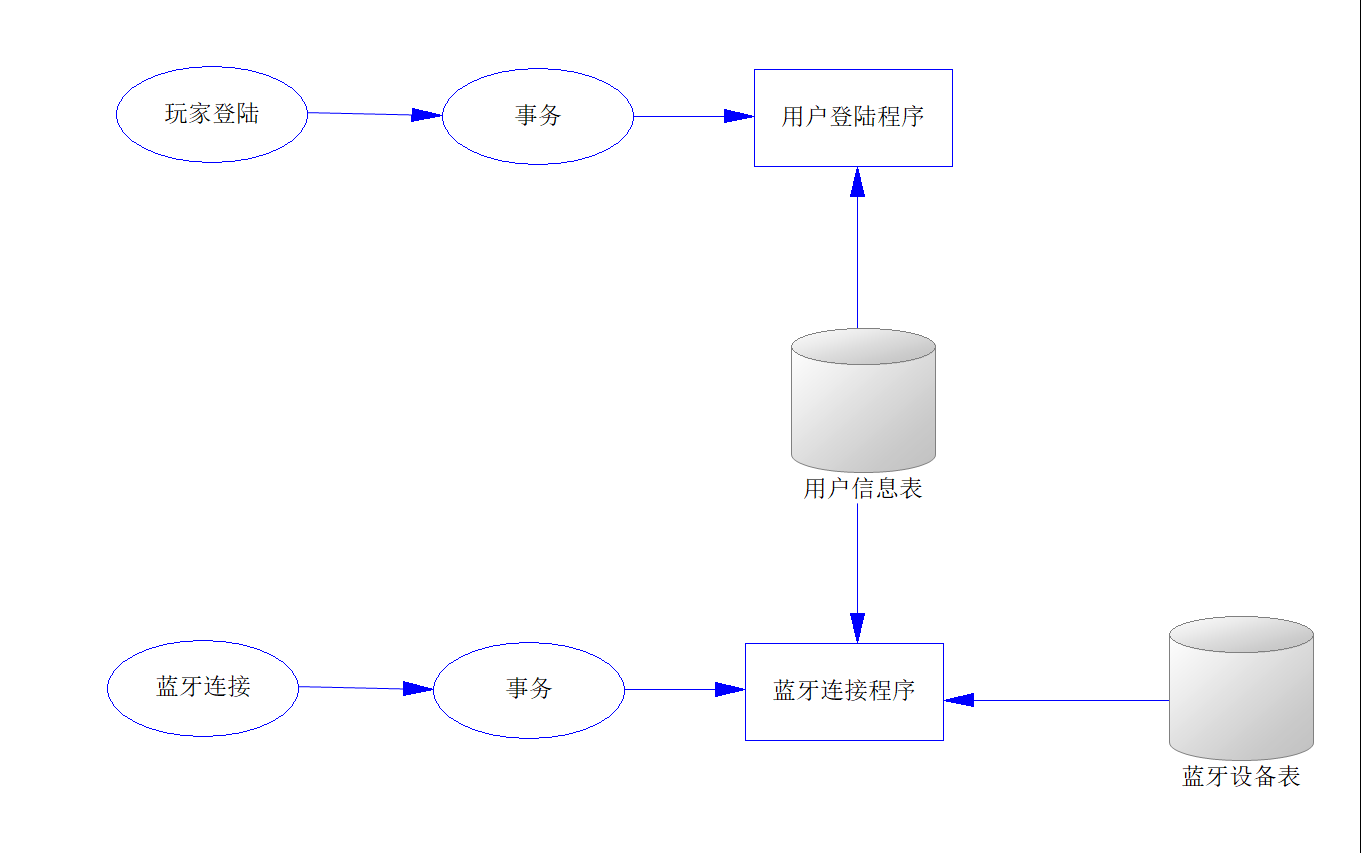
位置:区域信息

**基本系统模型:**

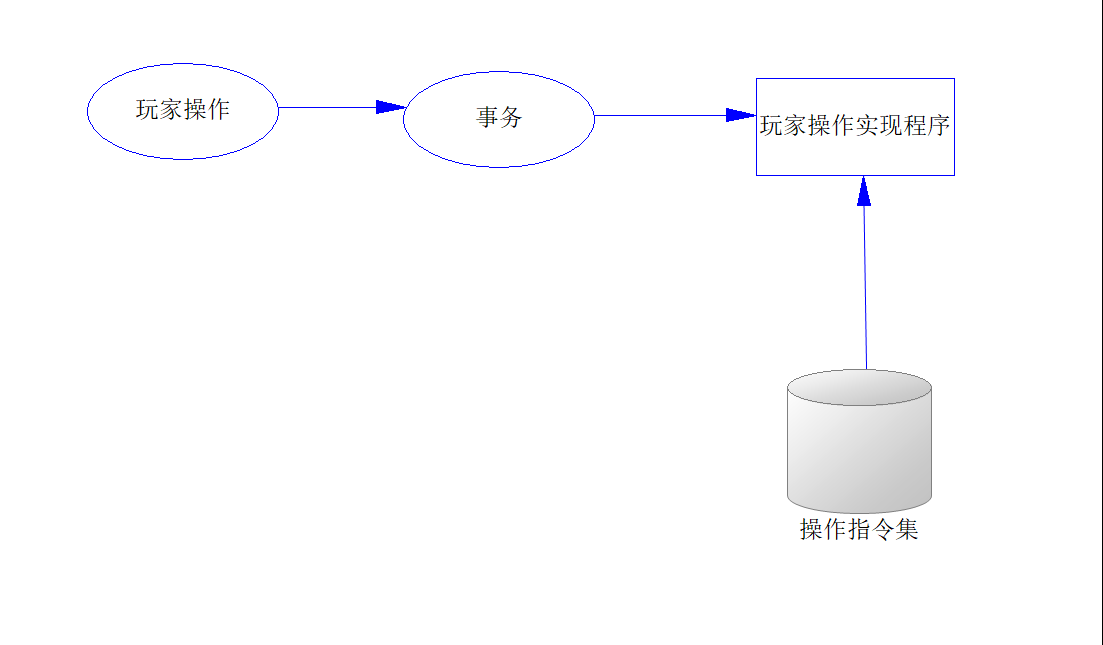


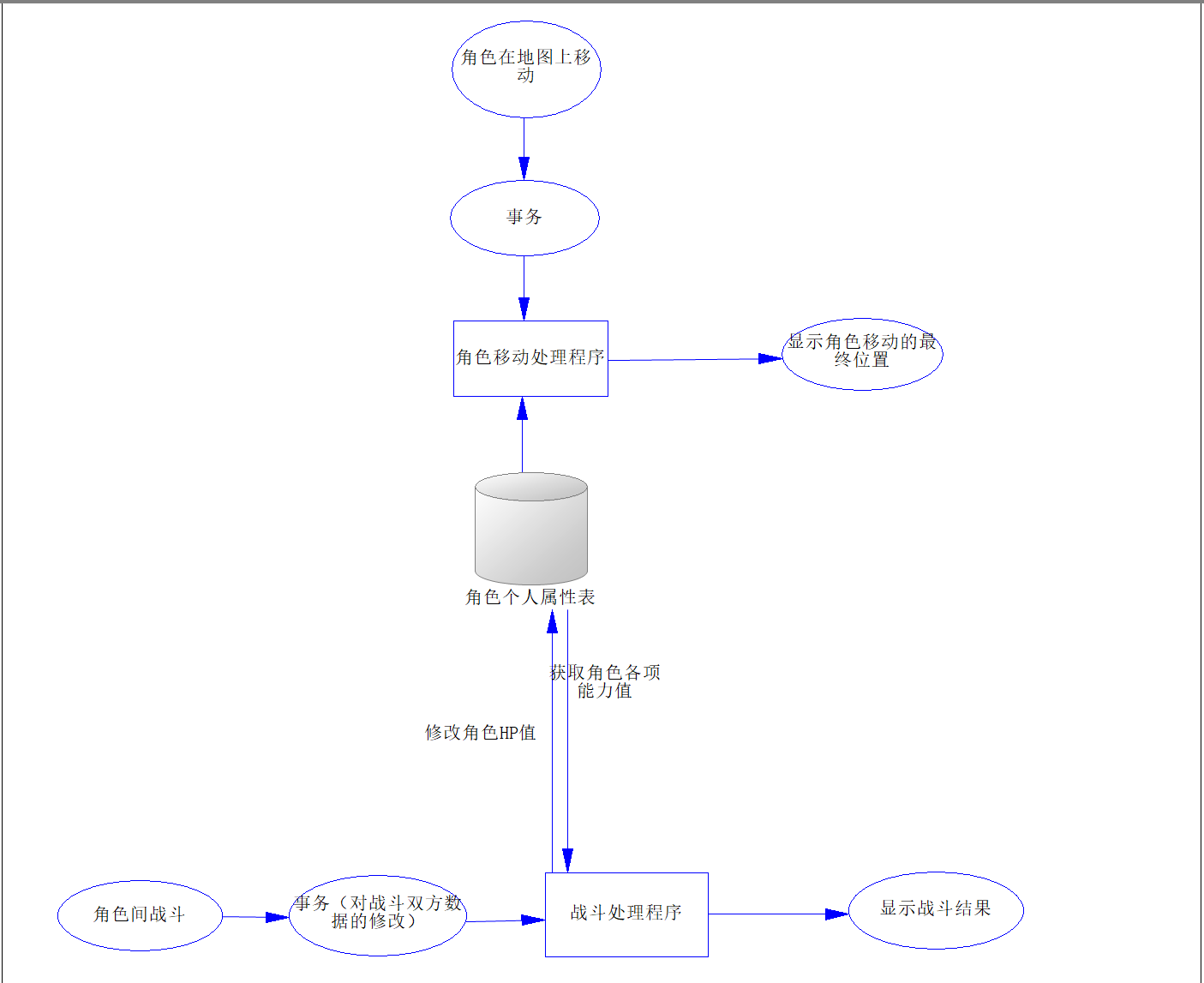
**系统流程图:**

**用户模块:**

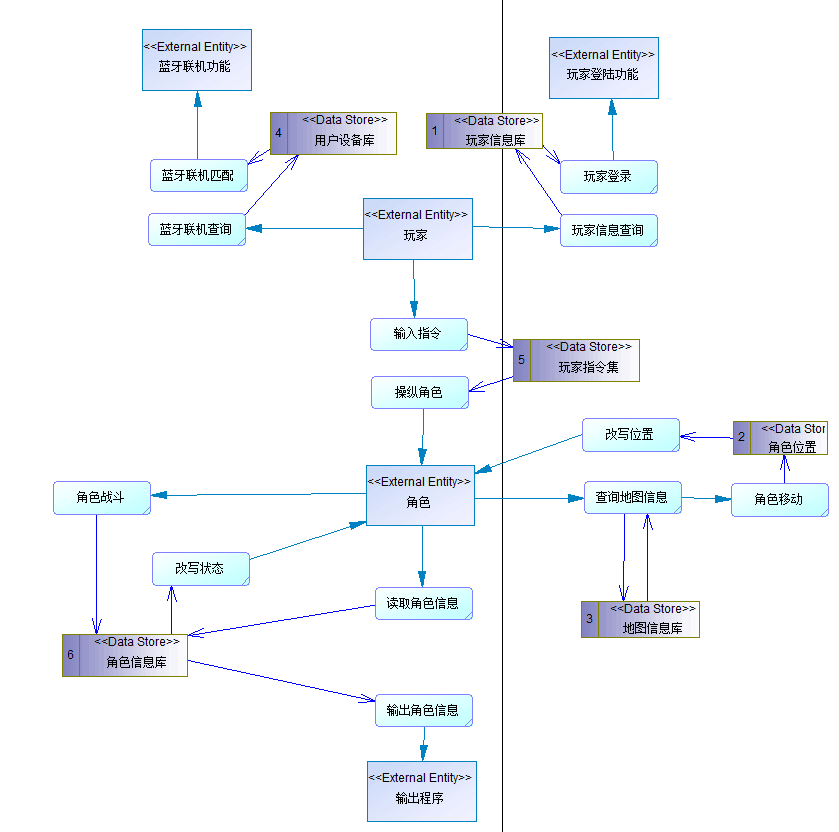
****

**操作模块:**





**数据流图:**



## 6.3 与原系统的比较(若有原系统)





## 6.4 影响(或要求)

### 6.4.1 设备

需要有搭载蓝牙功能的安卓手机,以支持联机功能.

### 6.4.2 软件

为了适应不同的开发人员使用的环境，我们小组统一使用了相同的环境配置以适应所有的软件开发过程.如:虚拟机统一适配WIN10操作系统,办公软件统一OFFICE2016版本,项目管理版本控制工具统一使用GIT进行管理.

### 6.4.3 运行

### 6.4.4 开发

Unity3D游戏引擎,MySQL

### 6.4.5 环境

Win10

### 6.4.6 经费

|  |  |  |
| --- | --- | --- |
| 资源 | 支出（单位：元） | 备注 |
| 开发者工资 | 0 | 本课程中没有工资 |
| Unity3D游戏引擎 | 0 |  |
| MySQL数据库 | 0 |  |
| xcode | 0 |  |
| Visual Studio | 0 |  |
| Data Grip | 0 | 学生免费 |
| photoshop | 0 |  |
| Axure RP | 870 | 3个月 |
| pro git v2.1.8 | 0 | git书籍，免费 |
| sublime text3 | 480 |  |

### 6.5 局限性

因作画与音乐都是小组自行完成,在品质上无法与专业的比较.

作为战棋类游戏,没有在操作便利的安卓端上实现

# 7 经济可行性(成本----效益分析)

# 7 经济可行性

## 7.1支出

### 7.1.1基本建设投资

按照每人每周工作11.5小时，以2017年杭州市最低时薪为标准，人工成本共计为7344元。

### 7.1.2其他一次性支出

租用服务器 60元，使用的开发框架都是开源的因此不存在费用的问题，使用的IDE是学生教育版也是免费的，因此除租用服务器外其他支出为0。

### 7.1.3代码行估计

根据现有的类似项目的代码行量，我们估计总的代码行应当在4k-5k左右。

## 7.2收益

因为是课程作业，故此预计的项目收益值为0，不加以考量。

# 8 技术可行性

## 8.1代码语言的选择

框架或游戏引擎选择：

可选框架或游戏引擎：

J2EE(WEB方向)  
Java+Apache+Hibernate+MySQL  
优点：  
1.功能强大  
2.开发人员熟悉,有一定技术储备  
3.有一些可选控件资源  
4.数量级用户量的扩展  
  
缺点：  
1.需要整体配置和构架  
2.UI设计、对Ajax的支持、Web页面风格  
3.需要统一个配置、功能模块独立性差,当功能增加或关联度增加时对开发人员的压力较大  
4.模版代码、数据库代码、配置代码和模块编码量大

Unity3D引擎

优点:

1.开发速度优势

2.编辑器开发功能全面

3.有丰富的学习资料与教程

缺点：

1.需要学习C#语言,小组成员相对不熟悉.

2.基于Mono的运行环境,使得初始化时间过长。对于本组的小型游戏来说，无法体现优势。

Cocos2D-X引擎(与主流游戏引擎Unity3D做对比)

优点:

1.与Unity3D相比Cocos2D-X的APK包要小了近10倍,更符合本游戏作为小型游戏的定位

2、Cocos2D-X是采用纯C++实现,比Unity3D每次启动都需加载Mono框架和Unity3D引擎，再加载.Net Assembly需要快很多,在小型游戏的层面,加载过长时间会使用户体验变差

3.中国游戏多出自Cocos2D-X,C2D对中国用户有优势。在社区上，能更加获得想要的资料与解答.   
4.Unity3D要收费,而Cocos2D-X免费、开源,遇到问题可以研究源码或在社区寻找答案.

缺点:

2.无法制作3D游戏

2.与Unity的参数可视化不同, Cocos2D-X引擎对程序的能力要求相对高一些

语言选择：

对于Web项目的开发 Python、Java是两个主流的语言方向选择.

若采取游戏引擎,Unity3D需要掌握C#语言, Cocos2D-X需要掌握C++、JavaScript或Lua语言.

|  |  |
| --- | --- |
| C++ | 可以快速调用封装好的功能,熟练应用上手慢,学习难度大 |
| JavaScript | 与Lua相比,可以更好的做好面向对象,易于后期维护更新 |
| Lua | 易与C/C++进行交互,在运行效率上优于JS,学习起来比较容易上手,适合短平快的小型游戏 |

框架或引擎开发必然能够加快开发的进度和效率已经准确度.

根据小组人力,成本,语言熟悉度与游戏类型方面综合考虑,决定在安卓平台上开发,并采用Cocos2D-X游戏引擎,将Lua语言作为开发语言.

因后期cocos2d-x无法实现,则改用Unity3D引擎,将C#语言作为开发语言

## 8.2数据库的选择

SQLserver

优点：

1、易用性、适合分布式组织的可伸缩性、用于决策支持的数据仓库功能、与许多其他服务器软件紧密关联的集成性、良好的性价比等。

2、为数据管理与分析带来了灵活性，允许单位在快速变化的环境中从容响应，从而获得竞争优势。从数据管理和分析角度看，将原始数据转化为商业智能和充分利用Web带来的机会非常重要。

缺点：

1、SQLserver只能在windows系统上使用

2、伸缩性并行性 ：SQL server 并行实施和共存模型并成熟难处理日益增多用户数和数据卷伸缩性有限。

3、安全性：没有获得任何安全证书。

4、性能 ：SQL Server 多用户时性能佳 。

MySql  
优点：  
1、体积小、速度快、总体拥有成本低，开源。  
2、支持多种操作系统。  
是开源数据库，提供的接口支持多种语言连接操作  
3、MySql的核心程序采用完全的多线程编程。线程是轻量级的进程，它可以灵活地为用户提供服务，而不过多的系统资源。用多线程和C语言实现的MySql能很容易充分利用CPU。  
4、MySql有一个非常灵活而且安全的权限和口令系统。当客户与MySql服务器连接时，他们之间所有的口令传送被加密，而且MySql支持主机认证；  
支持ODBC for Windows， 支持所有的ODBC 2.5函数和其他许多函数， 可以用Access连接MySql服务器， 使得应用被扩展。  
支持大型的数据库， 可以方便地支持上千万条记录的数据库。作为一个开放源代码的数据库，可以针对不同的应用进行相应的修改。  
拥有一个非常快速而且稳定的基于线程的内存分配系统，可以持续使用面不必担心其稳定性。   
5、MySQL同时提供高度多样性，能够提供很多不同的使用者介面，包括命令行客户端操作，网页浏览器，以及各式各样的程序语言介面，例如C+，Perl，Java，PHP，以及Python。你可以使用事先包装好的客户端，或者干脆自己写一个合适的应用程序。MySQL可用于Unix，Windows，以及OS/2等平台，因此它可以用在个人电脑或者是服务器上；  
  
缺点：  
1、不支持热备份；  
MySQL最大的缺点是其安全系统，主要是复杂而非标准，另外只有到调用mysqladmin来重读用户权限时才发生改变。

Oracle  
优点：  
1、开放性：Oracle 能所有主流平台上运行（包括 windows）完全支持所有工业标准采用完全开放策略使客户选择适合解决方案对开发商全力支持。  
2、可伸缩性,并行性：Oracle 并行服务器通过使组结点共享同簇工作来扩展windownt能力提供高用性和高伸缩性簇解决方案windowsNT能满足需要用户把数据库移UNIXOracle并行服务器对各种UNIX平台集群机制都有着相当高集成度。  
3、安全性：获得最高认证级别的ISO标准认证。    
性能：Oracle 性能高 保持开放平台下TPC-D和TPC-C世界记录。  
客户端支持及应用模式：Oracle 多层次网络计算支持多种工业标准用ODBC、JDBC、OCI等网络客户连接   
使用风险：Oracle 长时间开发经验完全向下兼容得广泛应用地风险低   
  
缺点：  
对硬件的要求很高。  
价格比较昂贵。  
管理维护麻烦一些。  
操作比较复杂，需要技术含量较高。

# 9操作可行性

在分析实际的操作可行性中，我们考虑了很多具体的实现方案。

游戏用户最早的简略游戏流程为：

打开游戏->主菜单

* 进入游戏：创建战局->邀请好友->选择对战类型->进入对战
* 个人资料：记录当前玩家各个游戏类型战局的历史记录和积分。

对战开始->选择个体

* IF->在攻击范围内有可以攻击的目标->不操作或者进行攻击/移动操作
* IF->在攻击范围内没有可攻击的目标->不操作或者进行移动操作

做完本回合部署点击按钮结束回合。

当一方完成全部获胜条件则游戏结束，游戏总结，更新双方战局记录和评分。

方案一：

目标软件为PC网页游戏，用户在网页上申请账号进行随机匹配对局。

解决了的问题：

1. 玩家打开游戏更加方便，可以随时在网上随机匹配对局。
2. 在网络上可以方便联系客服进行询问。
3. 网上可以根据不同玩家的分值进行全服的排名。
4. 可以智能将分值相近的玩家优先匹配。

衍生出的问题：

1. 网络存储不甚稳定，容易造成数据流失。
2. 大量一次性账号占据存储空间。

方案二：

目标软件为PC客户端游戏，继承方案一所有功能。

解决了的问题：

1. 数据存储较为稳定，可以进行添加好友和好友匹配对局等操作。
2. 可以按时清理久未在线的马甲号。

衍生出的问题：

1. 沙盘类游戏在PC端上操作较为繁琐，不能简便的拖拽地图，了解战局全貌。
2. 沙盘类游戏用时较长，需要对战时精神集中，不能一心二用，对玩家日常行动限制较大，在现今快节奏社会，长时间固定座位游戏对玩家限制较大。

方案三：

将游戏做为手机安卓端游戏，利用cocos2D引擎进行开发。

实施方案：

针对上述方案，我们提出了新的实施方案，将游戏定位为PC游戏.

游戏设定为联网对战模式，用户可以自主选择对战对象，方便进行相关沟通。游戏将实现对战双方数据同步，使得游戏拥有更强的连贯性，提高用户的游戏体验。

针对上述新方案的确定，分析其技术可行性所谈及到的关键内容，我们小组对其还处于陌生阶段，而要完成项目所涉及到的关键点，就需要我们对这些技术有很好的掌握以及能够熟练地运用这些框架和技术。

因此我们在项目开展的过程中，就应当自己抽出额外的时间去学习相应的技术以及应用框架，这部分的时间是不算在项目工时中的。当真正进行开发时，再加以强化这些技术的实现和增强个人熟练度。

同时经过小组内的一致讨论，将这些技术掌握的任务进行了合理的分配。项目经理刘雨霏作为统筹人，需要掌握上述技术的基本实现，胡方正掌握Unity3D的网络连接和数据库的连接存储问题，同时管理好服务器的内容。杨智麟需要了解Unity3D在UI设计上的应用效果，使其与基本操作连接。

经过上述分析，我们认为虽然使用这些较为先进的框架以及技术会在前期学习的时候耗费较多的精力以及时间，但是长远来看，对整个项目的开发以及后期扩展维护是极为方便有利的，并且我小组的三名成员在代码能力上都具有较好的基础，学习能力较强，故此操作上应该可行。

# 10 法律可行性

本项目作为课程作业，并不实际投入使用，也不存在法律方面的问题。

# 11 用户使用可行性

界面简洁明了，拥有基础教学关卡，游戏用户容易上手操作。

1.游戏用户可以通过教学关卡进行学习。

2.游戏用户可以在设置中寻找相关帮助。

3. 游戏用户可以与客服沟通。

# 12 其他与项目有关的问题

## 12.1项目总结

经过上述从技术、经济、以及可操作性这三个大方面的分析，我们G02小组一致认为项目具有可行性，建议立即开展项目计划并加以实施。

# 13 注解

# 附录