

# “一起滚铁环” 微信小游戏

## [软件设计说明书]

## 目录

1. 引言 .....	5
1.1 编写目的 .....	5
1.2 命名规范 .....	5
1.3 术语定义 .....	5
1.4 参考资料 .....	5
1.5 相关文档 .....	5
1.6 版本更新记录 .....	6
2. 总体设计 .....	6
2.1 硬件运行环境 .....	6
2.2 软件运行环境 .....	6
2.3 开发环境 .....	6
2.4 子系统清单 .....	7
3. 数据库设计 .....	7
3.1 数据库中表名列表 .....	7
3.2 表的详细清单 .....	7
4. 典型功能子系统设计 .....	7
4.1 主界面系统.....	8
4.1.1 活动图设计 .....	8

4.1.2 顺序图设计 .....	9
4.2 结束界面系统 .....	9
4.2.1 活动图设计 .....	9
4.2.2 顺序图设计 .....	10
4.3 游戏界面系统 .....	10
4.3.1 活动图设计 .....	10
4.3.2 顺序图设计 .....	11
4.4 排行榜系统 .....	11
4.4.1 活动图设计 .....	11
4.4.2 顺序图设计 .....	12
4.5 分享系统 .....	12
4.5.1 活动图设计 .....	12
4.5.2 顺序图设计 .....	13
<b>5. 功能模块设计 .....</b>	<b>13</b>
5.1 主界面设计 .....	13
5.2 结束界面设计 .....	14
5.3 游戏界面设计 .....	14
5.4 暂停界面设计 .....	15
5.5 排行榜界面设计 .....	15

5.6 分享界面设计 .....	15
6. 存储过程设计 .....	16
7. 接口设计 .....	16
7.1 用户接口 .....	16
7.2 外部接口 .....	16
7.3 内部接口 .....	16
8. 角色授权设计 .....	17
9. 系统错误处理 .....	17
9.1 出错信息 .....	17
9.2 故障预防与补救 .....	17
9.3 系统维护设计 .....	17
10. 测试计划 .....	18

## 1. 引言

### 1.1 编写目的

在完成“一起滚铁环”微信小游戏需求分析的基础上，我们依据《需求规格说明书》对该系统进行了设计。为了文档化对系统的设计，编写本文档。本文档主要供团队在具体实现“一起滚铁环”微信小游戏时参考使用。

### 1.2 命名规范

数据库表明、字段名用大写开头的英文命名。

私有变量用“\_”开头的小写英文命名。其他的变量用大写英文字母开头。

所有的函数用大写英文字母开头。

### 1.3 术语定义

总体结构：软件系统的总体逻辑结构，本系统采用面向对象的方法对系统进行设计。

概念数据模型（CDM）：关系数据库的概念设计模型，主要表现为 E-R 图。

物理数据模型（PDM）：关系数据库的物理设计模型。

### 1.4 参考资料

[1] 吕云翔. 软件工程实用教程. 北京：清华大学出版社，2015.

### 1.5 相关文档

“一起滚铁环”微信小游戏《软件开发说明书》

“一起滚铁环”微信小游戏《需求规格说明书》

“一起滚铁环”微信小游戏《测试分析报告》

## 1.6 版本更新记录

版本更新记录见表 1。

表 1 版本更新记录

版本号	创建者	创建日期	维护者	维护日期	维护纪要
V1.0	王煦阳	2019-5-5	王煦阳		

## 2. 总体设计

### 2.1 硬件运行环境

1. 安卓手机：系统要求：安卓 4.0 以上系统适用屏幕像素：通用。
2. iPhone：系统要求：iPhone6.1-iPhone8.0 以上系统适用屏幕像素：通用。

### 2.2 软件运行环境

微信 v6.6.1 以上版本（微信小游戏平台，使用微信底层接口，使用了 cocos 引擎，使用 HTML5，javascript 语言。）

### 2.3 开发环境

本小游戏采用 COCOS CREATOR 进行开发  
利用 github 进行项目管理

## 2.4 子系统清单

子系统清单见表 2.1

表2.1 子系统清单

## 3. 数据库设计

### 3.1 数据库中表名列表

表 3.1 数据库中的表名列表		
编号	表名	表功能说明
1	Ranking List	玩家得分信息表

### 3.2 表的详细清单

表 3.2.1 RankingList 表

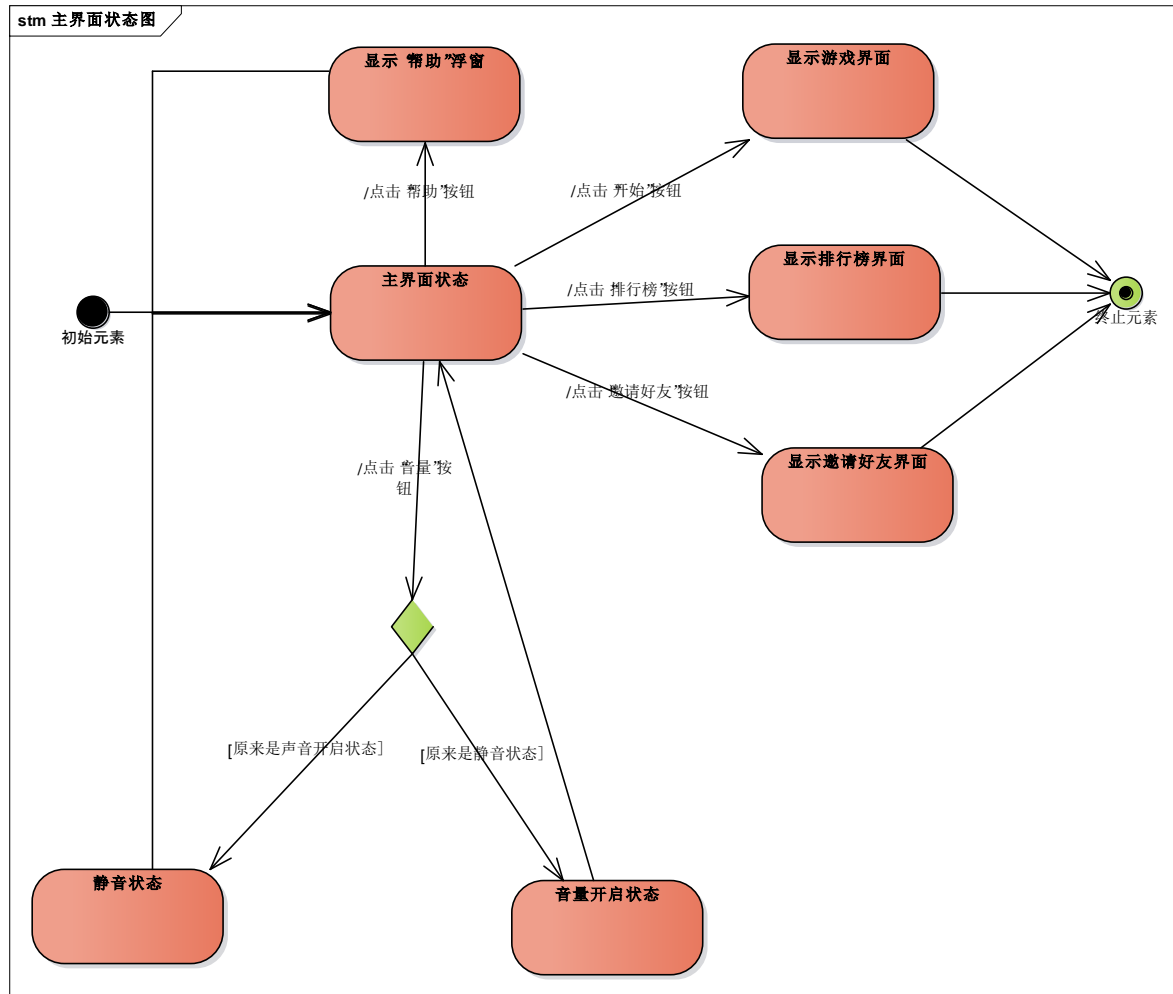
序号	英文字段名	中文字段名	数据类型	是否允许为空	主键/外键
1	ID	用户ID	int	否	主键
2	Username	用户昵称	varchar(20)	否	
3	Score	用户分数	int	否	
4	Date	记录时间	date	否	

## 4. 典型功能子系统设计

本节以一起滚铁环游戏系统为例，从主界面系统、结束界面系统、游戏界面系统、排行榜系统以及分享系统五个层面进行详细设计描述。

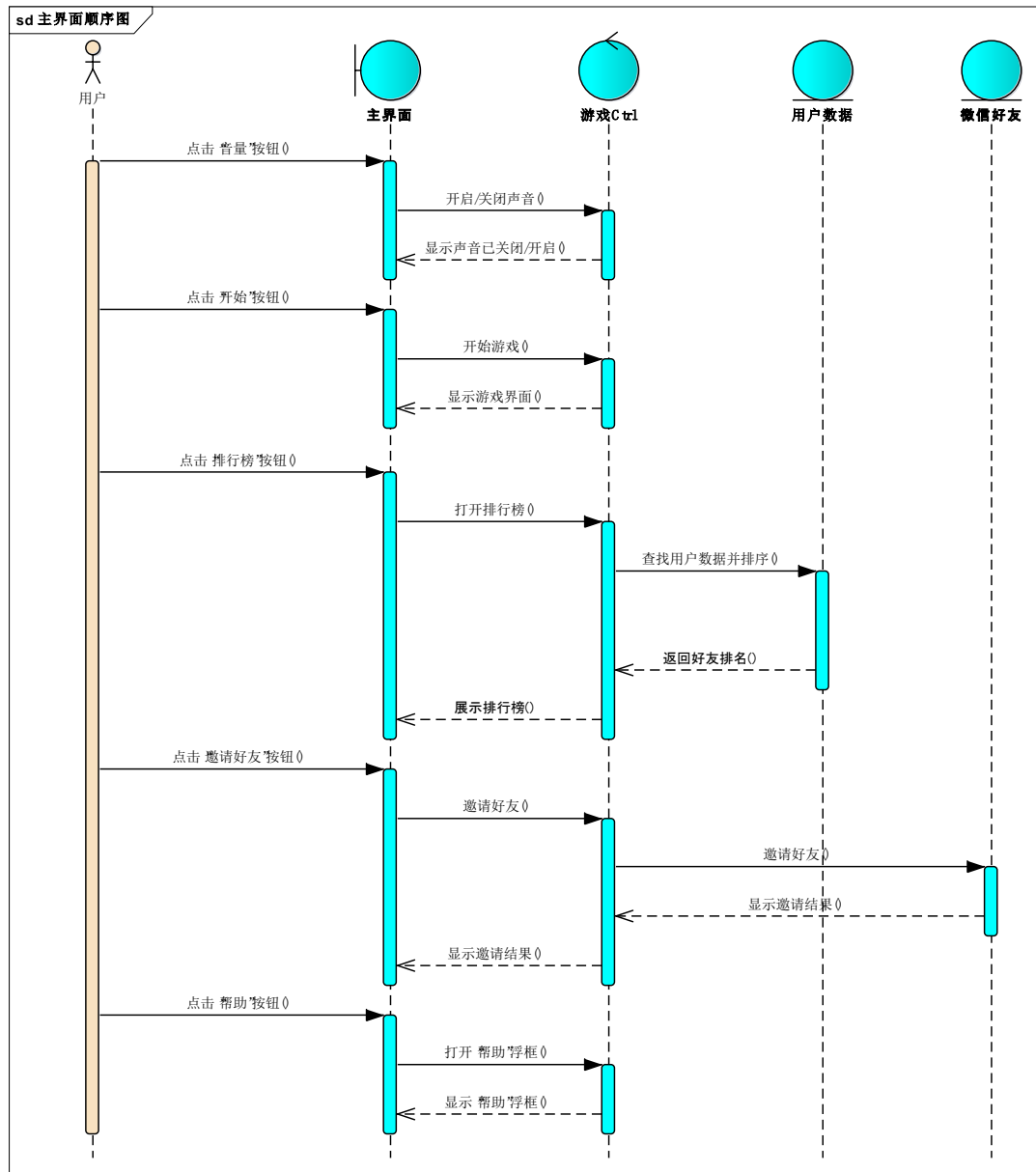
## 4.1 主界面系统

### 4.1.1 活动图设计



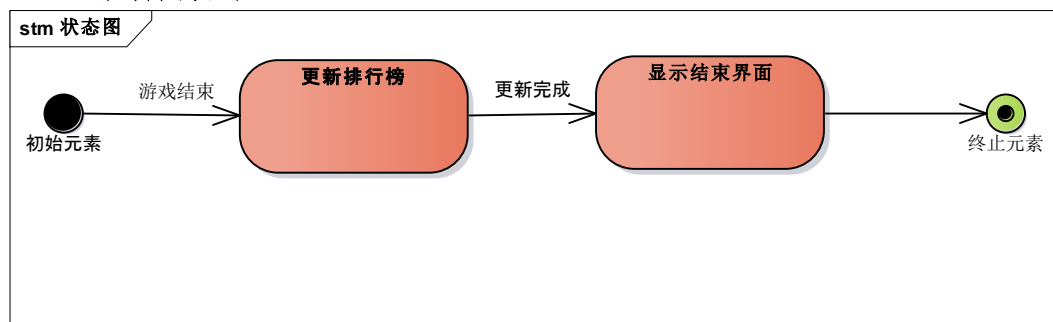


## 4.1.2 顺序图设计

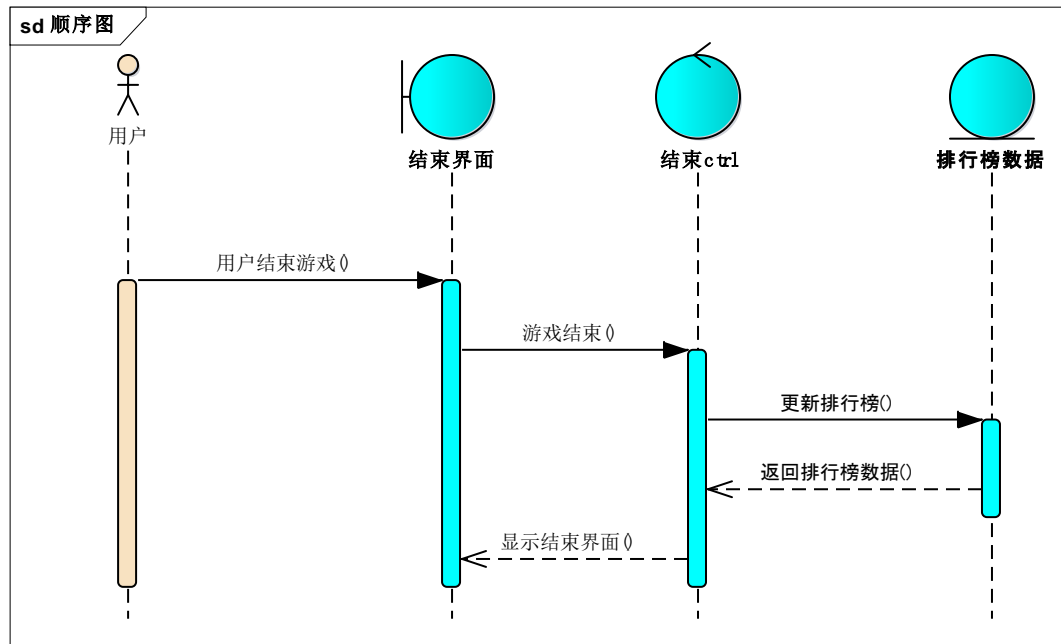


## 4.2 结束界面系统

### 4.2.1 活动图设计

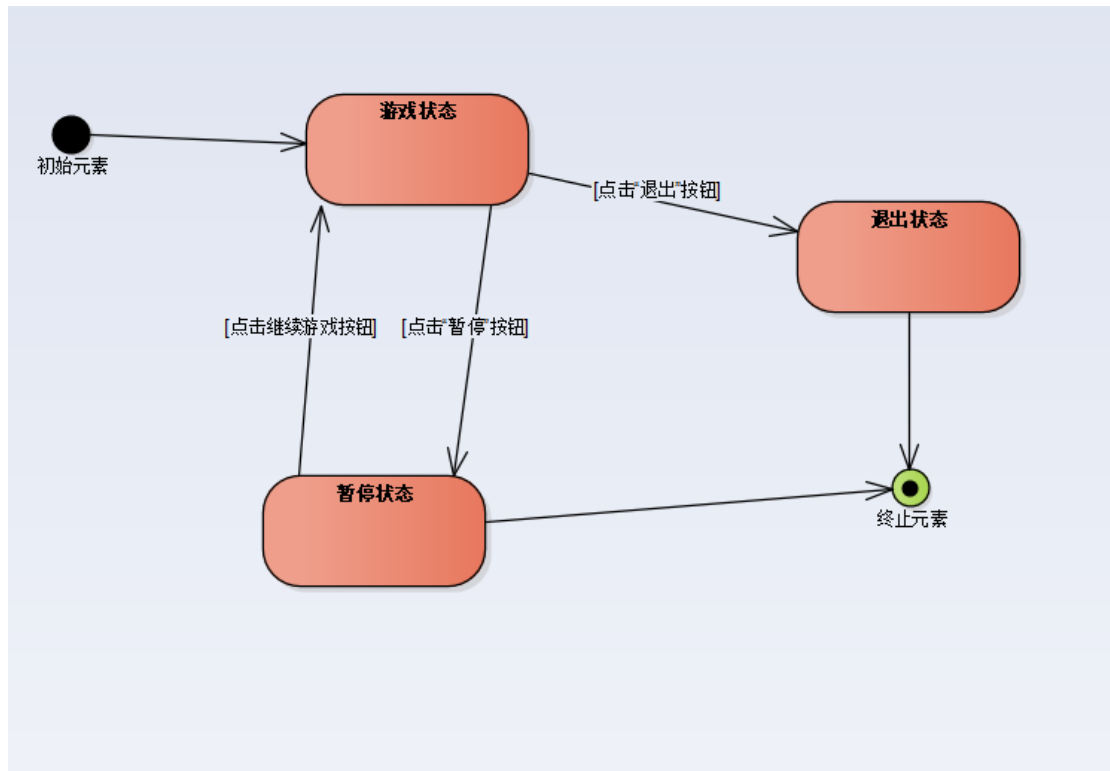


## 4.2.2 顺序图设计

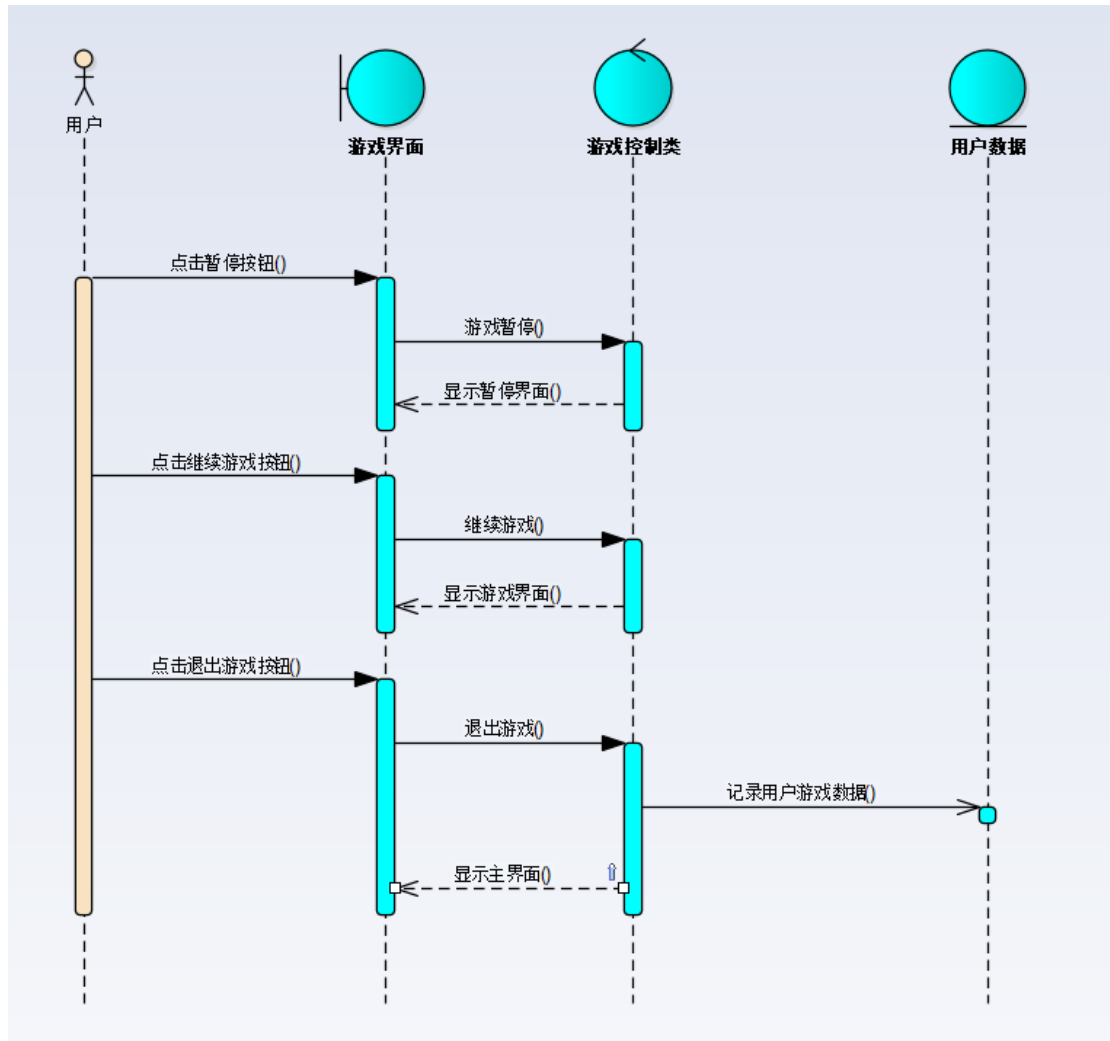


## 4.3 游戏界面系统

### 4.3.1 活动图设计

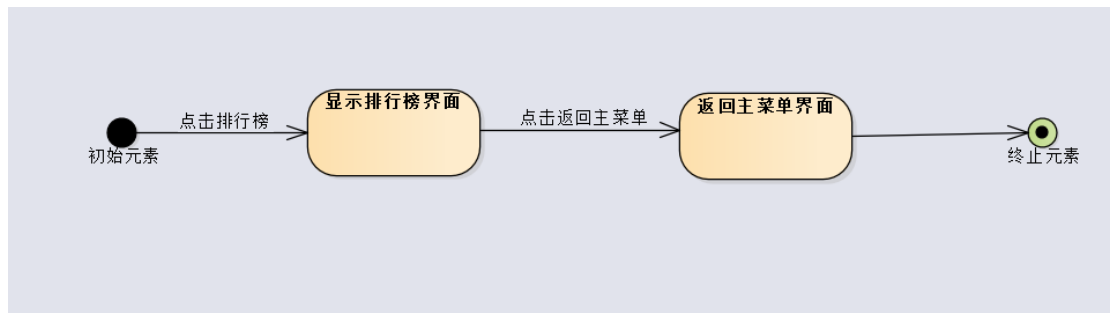


### 4.3.2 顺序图设计

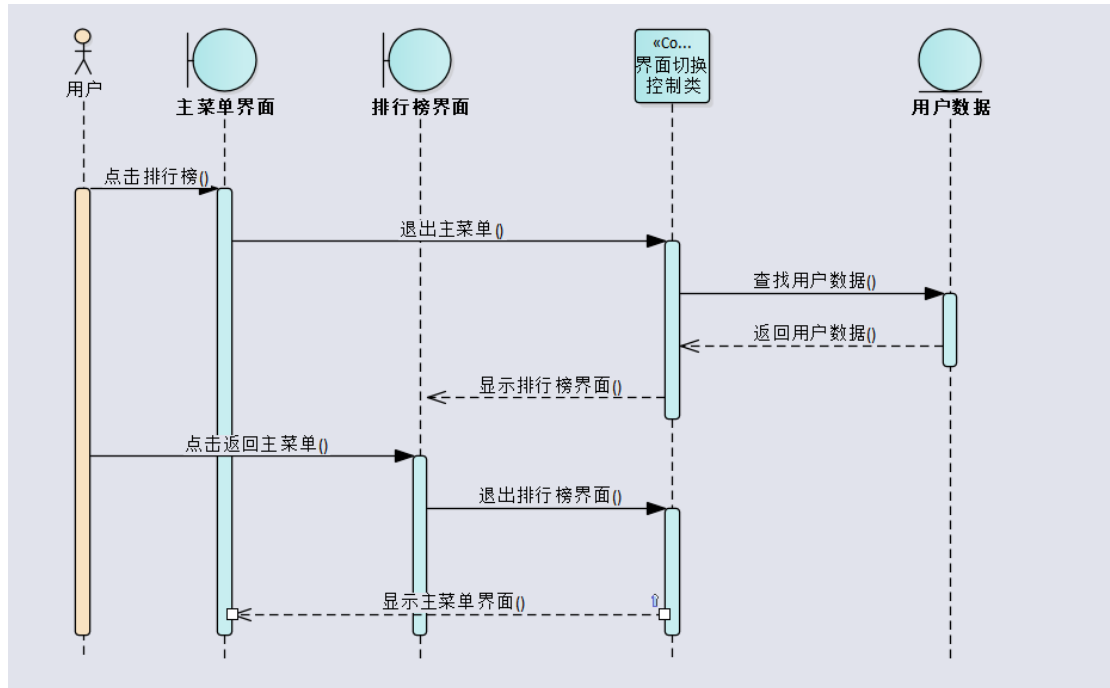


## 4.4 排行榜系统

### 4.4.1 活动图设计

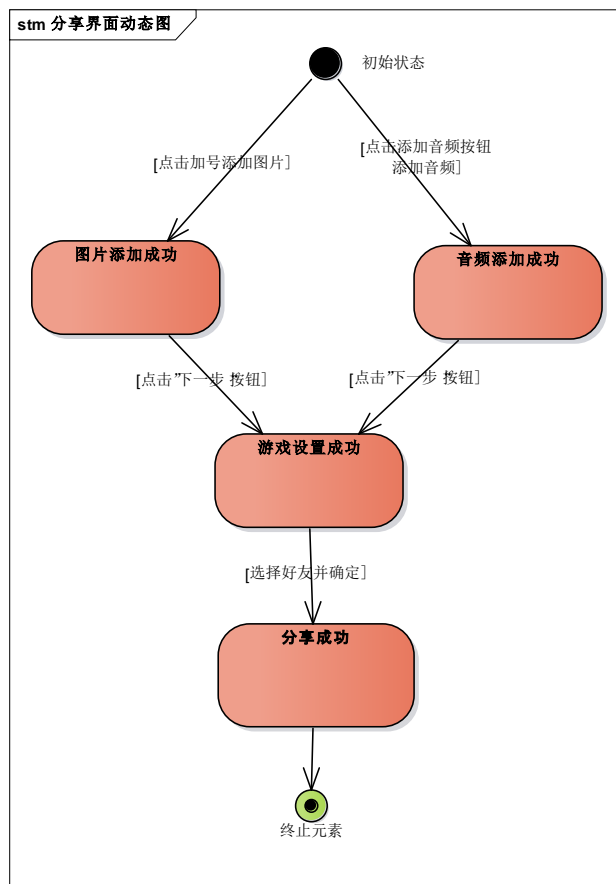


#### 4.4.2 顺序图设计

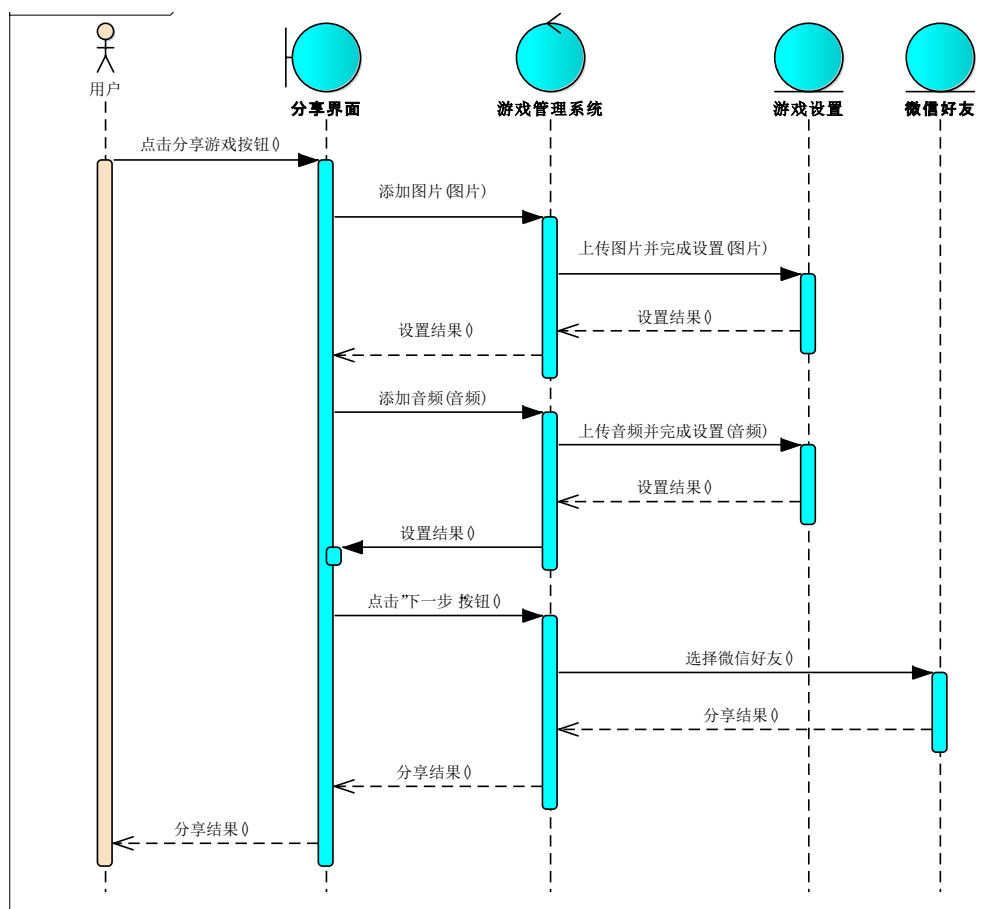


#### 4.5 分享系统

##### 4.5.1 活动图设计



## 4.5.2 顺序图设计



## 5. 功能模块设计

### 5.1 主界面设计

调用背景：

- (1) 小游戏启动时跳转至此界面。
- (2) 游戏暂停界面点击“返回至主菜单”，跳转至此界面。
- (3) 游戏结束时点击“返回至主菜单”，跳转至此界面。

页面组成：

主界面中上部显示“一起滚铁环”游戏名称。  
 游戏名称右方有“音量”按钮。  
 游戏名称下方有“开始”按钮，开始按钮下方有“邀请好友”按钮。  
 主界面左下角有“排行榜”按钮。  
 主界面右下角有“帮助”按钮。

调用描述:

当用户点击“音量”按钮时,若最初是开启声音,则点击后关闭声音;若最初是关闭声音,则点击后开启声音。

当用户点击“开始”按钮时,游戏跳转到游戏界面。

当用户点击“邀请”按钮时,游戏跳转到邀请好友界面。

当用户点击“排行榜”按钮时,游戏跳转到排行榜界面。

当用户点击“帮助”按钮时,游戏弹出“帮助”浮框。

## 5.2 结束界面设计:

调用背景:

(1) 当玩家处于游戏界面,在游戏结束时自动跳转。

页面组成:

正上方显示玩家走了多少米和玩家获得的分数的文本。中间是显示用户头像的图片框,下面是显示游戏排行榜的表格。最下方是一个“再次游戏”按钮

调用描述:

(1) 当玩家处于游戏界面且游戏结束时,程序自动跳转到该界面。

## 5.3 游戏界面设计

调用背景:

(1) 在主界面点击“开始游戏“,跳转到该页面。

(2) 在结束界面点击“再次挑战”,跳转到该页面。

页面组成:

左上角有路程显示。

右上角有暂停按钮和退出按钮。

中间部分有三条跑道以及代表用户的小铁环。在旁边会有动态显示的背景。

调用描述:

当用户倾斜手机时,游戏界面中的小铁环会向对应方向移动来改变跑道。

当用户点击暂停按钮时,小铁环停止移动,背景停止改变,并跳转到暂停界面。

当用户点击退出按钮时,跳转到主界面。

## 5.4 暂停界面设计

调用背景：

在游戏界面点击暂停按钮，跳转到该界面

页面组成：

在页面中央显示用户名和头像

下面是已走过的路程及用户得分

最下面有继续游戏按钮。

调用描述：

当用户点击继续游戏按钮时，跳转回游戏界面，并恢复用户刚才的游戏过程。

## 5.5 排行榜界面设计

调用背景：

玩家处于主菜单界面，点击“排行榜”，跳转到该页面。

页面组成：

上方“排行榜”三个字，中间是每个玩家的用户名和得分，最下面是一个“返回主菜单”

调用描述：

玩家处于主菜单界面，点击“排行榜”，跳转到该页面。

## 5.6 分享界面设计

调用背景：

在主界面点击“分享游戏”，跳转到该页面。

页面组成：

页面中间有三个中间有加号的空矩形图形。

页面下方有下一步按钮。

调用描述：

当用户点击空矩形中心加号时，能够打开手机相册选取游戏图片。

当用户点击下一步按钮时，跳转到微信好友界面，选择好友并确定则分享成功。

## 6. 存储过程设计

### 1. AddScore()

功能描述：添加得分记录

入口参数：@UserID(int, 无默认值)

          @Username(varchar(20), 无默认值)

          @Score(int, 无默认值)

出口参数：无伪语言实现：

```
create procedure AddScore (@UserID, @Username, @Score) as declare
```

```
if @UserID is not null
```

```
begin
```

```
insert into RankingList
```

```
values(@UserID, @Username, @Score, @Date,
```

```
(select convert(char(10),getdate(),120))
```

```
end return
```

## 7. 接口设计

### 7.1 用户接口

基于 Java Script 搭建，提供清晰简洁易用美观的用户界面。

### 7.2 外部接口

数据存储：通过小游戏云开发平台上的一个既可在小程序前端操作，也能在云函数中读写的 JSON 数据库

### 7.3 内部接口

基于腾讯的小程序云开发平台，连接前后端



## 8. 角色授权设计

角色授权见表 8.1

表 8.1 角色授权表

子系统名称	管理员	玩家
主界面系统	●	○
结束界面系统	●	○
游戏界面系统	●	○
排行榜系统	●	○
分享系统	●	○

●表示有全部权限，○表示享有权限但并非全部权限。

玩家可以访问这些界面系统，但对于一些后台信息的处理只有管理员能做到。

## 9. 系统错误处理

### 9.1 出错信息

(1) 对玩家输入的各项内容进行有效性、安全性检查、减少错误发生的机率。

(2) 对程序运行中的异常均进行捕获，按统一的方式将出错提示信息提供给玩家。

### 9.2 故障预防与补救

以统一的机制进行网站权限的控制，对程序中用到的数据尽量进行加密，以减少黑客攻击的危害。定期对数据库中的数据进行海量备份及增量备份。

### 9.3 系统维护设计

(1) 编码实现时应采用模块化和分层的思想，提高模块内部的内聚，减少模块间的耦合。使系统逻辑结构清晰，从而增强可读性和可维护性。

(2) 面向数据与面向对象相结合，模块划分符合面向对象思想。

(3) 在编码过程中注意标识符命名的意义，添加适量注释。

## 10. 测试计划

测试计划作为本系统的测试指导，测试成员按计划和规定进行测试。本系统的测试将从单元测试、集成测试、系统测试 3 个方面来设计。