|  |
| --- |
| **浙江大学城市学院** |
| **用户操作手册** |
| **H5套圈小游戏** |

|  |
| --- |
| 项目名称 基于html5的休闲小游戏  组长 陈帆  组员 张荣阳、赵伟宏  小组号 G07  专业班级 软件工程1601  指导老师 杨枨 |



**修订历史**

|  |  |  |  |
| --- | --- | --- | --- |
| **日期** | **版本** | **说明** | **作者** |
| 2018-6-13 | 0.1 | 最后更改 | 张荣阳 |
| 2018-6-28 | 1.0 | 第一次正式修订 | 陈帆 |

目录

[1．引言 1](#_Toc506976439)

[1.1编写目的 1](#_Toc506976440)

[1.2项目背景 2](#_Toc506976441)

[1.3定义 2](#_Toc506976442)

[1.4参考资料 2](#_Toc506976443)

[2．软件概述 2](#_Toc506976444)

[2.1目标 2](#_Toc506976445)

[2.2功能 2](#_Toc506976446)

[2.3性能 2](#_Toc506976447)

[3．运行环境 2](#_Toc506976448)

[3.1硬件 2](#_Toc506976449)

[3.2支持软件 3](#_Toc506976450)

[4.使用说明 3](#_Toc506976451)

[4.1安装和初始化 3](#_Toc506976452)

[4.2输入 3](#_Toc506976453)

[4.3输出 3](#_Toc506976454)

[4.4出错和恢复 3](#_Toc506976455)

[4.5求助查询 3](#_Toc506976456)

[5.运行说明 3](#_Toc506976457)

[5.1运行表 3](#_Toc506976458)

[5.2运行步骤 3](#_Toc506976459)

[6.非常规过程 3](#_Toc506976460)

[7.操作命令一览表 3](#_Toc506976461)

[8.程序文件（或命令文件）和数据文件一览表 3](#_Toc506976462)

[9.用户操作举例 3](#_Toc506976463)

# 1．引言

## 1.1编写目的

帮助网页游戏用户更好的了解我们的游戏操作方式，同时了解我们游戏的初衷

## 1.2项目背景

【应包括项目的来源、委托单位、开发单位和主管部门。】

来源：该游戏项目由陈帆，张荣阳，赵伟宏三人设计开发实现；

委托单位:杨枨老师

开发单位：G07

主管部门：软件工程课程小组

## 1.3定义

[1]h5:

H5是一系列制作网页互动效果的技术集合，即H5就是移动端的web页面。而H5游戏，你可以看作是移动端的web游戏，无需下载软件即可体验，这就是H5在传播上的优势。对于许多手游玩家来说，H5可能是个十分陌生的名词，其实H5就是HTML的高级版本，目前H5技术正在不断完善。

[2]Ubuntu:

Ubuntu（友帮拓、优般图、乌班图）是一个以[桌面](https://baike.baidu.com/item/%E6%A1%8C%E9%9D%A2)应用为主的开源GNU/Linux操作系统，Ubuntu 是基于[Debian](https://baike.baidu.com/item/Debian) GNU/[Linux](https://baike.baidu.com/item/Linux)，支持x86、amd64（即x64）和[ppc](https://baike.baidu.com/item/ppc/150" \t "_blank)架构，由全球化的专业开发团队（Canonical Ltd）打造的。

[3]MySQL:

MySQL是一个[**关系型数据库管理系统**](https://baike.baidu.com/item/%E5%85%B3%E7%B3%BB%E5%9E%8B%E6%95%B0%E6%8D%AE%E5%BA%93%E7%AE%A1%E7%90%86%E7%B3%BB%E7%BB%9F/696511)**，**由瑞典MySQL AB 公司开发，目前属于 [Oracle](https://baike.baidu.com/item/Oracle) 旗下产品。MySQL 是最流行的[关系型数据库管理系统](https://baike.baidu.com/item/%E5%85%B3%E7%B3%BB%E5%9E%8B%E6%95%B0%E6%8D%AE%E5%BA%93%E7%AE%A1%E7%90%86%E7%B3%BB%E7%BB%9F/696511)之一，在 WEB 应用方面，MySQL是最好的 RDBMS (Relational Database Management System，关系数据库管理系统) 应用软件。

[4]tomcat:

Tomcat是Apache 软件基金会（Apache Software Foundation）的Jakarta 项目中的一个核心项目，由[Apache](https://baike.baidu.com/item/Apache/6265)、Sun 和其他一些公司及个人共同开发而成。由于有了Sun 的参与和支持，最新的Servlet 和JSP 规范总是能在Tomcat 中得到体现，Tomcat 5支持最新的Servlet 2.4 和JSP 2.0 规范。因为Tomcat 技术先进、性能稳定，而且免费，因而深受Java 爱好者的喜爱并得到了部分软件开发商的认可，成为目前比较流行的Web 应用服务器。

[5]shell:

在计算机科学中，Shell俗称壳（用来区别于核），是指“提供使用者使用界面”的软件（命令解析器）。它类似于[DOS](https://baike.baidu.com/item/DOS)下的command.com和后来的cmd.exe。它接收用户命令，然后调用相应的应用程序。

[6]云服务器

云服务器是一种简单高效、安全可靠、处理能力可弹性伸缩的计算服务。其管理方式比物理[服务器](https://baike.baidu.com/item/%E6%9C%8D%E5%8A%A1%E5%99%A8)更简单高效。用户无需提前购买硬件，即可迅速创建或释放任意多台云服务器。

## 1.4参考资料

【列出有关资料的作者、标题、编号、发表日期、出版单位或资料来源，可包括：

1. 项目的计划任务书、合同或批文；G07
2. 项目开发计划；G07
3. 需求规格说明书；G07
4. 概要设计说明书；G07
5. 详细设计说明书；G07
6. 测试计划；G07
7. 手册中引用的其他资料、采用的软件工程标准或软件工程规范。

# 2．软件概述

## 2.1目标

实现h5小游戏的所有预定功能。

成功发布h5小游戏。

## 2.2功能

## 我们设计这款游戏可以使用户在碎片化的时间里进行放松和娱乐。同时兼具社交层面的互动，帮助人们加固人际关系。

## 2.3性能

1. 数据精确度【包括输入、输出及处理数据的精度。】

输入:char或者int

输出:同上

处理数据:服务器

1. 时间特性【如响应时间、处理时间、数据传输时间等。】

响应时间：0.5s

处理时间: 0.5s以内

数据传输时间：0.5s

1. 灵活性【在操作方式、运行环境需做某些变更时软件的适应能力。】

操作方式:(暂定模仿跳一跳)

运行环境：网页

# 3．运行环境

## 3.1硬件

1. 管理端计算机型号:服务器（ubuntu）
2. 用户端设备型号：任意带有浏览器的个人设备
3. 输入设备：触控屏或鼠标
4. 数据传输设备：能运行网页的手机
5. 数据转换设备：服务器
6. 数据存储设备：服务器

## 3.2支持软件

1. 操作系统名称：windows、安卓、苹果ios
2. 数据库管理系统:MySQL
3. 其他必要的支持软件：tomcat

# 4.使用说明

## 4.1安装和初始化

程序的存储形式：html5网页（前端）、服务器通讯响应程序（服务器）、数据库操作程序（服 务器）

操作命令：在服务器的shell中运行“echo\_server.js”文件，具体命令为“node echo\_server”。

反馈信息：当服务器的shell中成功运行“echo\_server.js”时，shell中会打印“########”

前端安装：需要在服务器中挂在域名，在域名指向的文件夹中放置html文件。玩家使用浏 览器访问该域名后，应能正常载入游戏界面。

## 4.2输入

### 4.2.1数据背景

1. 用户注册信息：id+key（char）
2. 数据来源：玩家在前端“用户注册”界面输入的用户名和用户密码。
3. 存储媒体：数据库。
4. 出现频度：一次/玩家。
5. 限制：一个玩家仅有一次注册机会。
6. 质量管理：不允许出现错误。
7. 游戏信息：分数+用户id（int+char）
8. 数据来源：玩家在前端登陆后进行游戏，当取得的分数高过其最高成绩时，覆盖该数据。
9. 存储媒体：数据库
10. 出现频度：一次/游戏一次
11. 限制：一个登陆玩家每游戏一次可以覆盖该数据。
12. 质量管理：允许出现小规模错误。

### 4.2.2数据格式

一、用户名

1. 长度：2—36字节
2. 格式基准：字符串
3. 标号：userid
4. 密码
5. 长度：0—18字节
6. 格式基准：字符串
7. 标号：password
8. 最高成绩
9. 长度：4字节
10. 格式基准：非负整数
11. 标号：score

### 4.2.3输入举例

登录注册

001

123456

分数：

01

56

## 4.3输出

### 4.3.1数据背景

1. 用户注册信息：id+key（char）
2. 数据来源：玩家在前端“用户注册”界面输入的用户名和用户密码。
3. 存储媒体：数据库。
4. 出现频度：一次/玩家。
5. 限制：一个玩家仅有一次注册机会。
6. 质量管理：不允许出现错误。
7. 游戏信息：分数+用户id（int+char）
8. 数据来源：玩家在前端登陆后进行游戏，当取得的分数高过其最高成绩时，覆盖该数据。
9. 存储媒体：数据库
10. 出现频度：一次/游戏一次
11. 限制：一个登陆玩家每游戏一次可以访问该数据。
12. 质量管理：允许出现小规模错误。
13. 游戏排行榜信息
14. 数据来源：玩家在前端登陆后进行游戏，当游戏结束后，可访问该数据。
15. 存储媒体：数据库
16. 出现频度：一次/游戏一次
17. 限制：一个登陆玩家每游戏一次可以访问该数据。
18. 质量管理：允许出现小规模错误。
19. 验证登陆
20. 数据来源：玩家登陆成功或失败后，服务器向前端发送数据。
21. 存储媒体：无。
22. 出现频度：一次/一次登陆尝试。
23. 限制：当玩家尝试登陆后会获得该信息。
24. 质量管理：不允许出现错误。
25. 验证注册
26. 数据来源：玩家注册成功或失败后，服务器向前端发送数据。
27. 存储媒体：无。
28. 出现频度：一次/一次注册尝试。
29. 限制：当玩家尝试注册后会获得该信息。
30. 质量管理：不允许出现错误。
31. 验证成绩
32. 数据来源：玩家游戏一次后，服务器向前端发送数据确认是否为新成绩。
33. 存储媒体：无。
34. 出现频度：一次/一次游戏。
35. 限制：当玩家登陆后每游戏一次会获得该信息。
36. 质量管理：允许出现小范围错误。

### 4.3.2数据格式

1. 尝试注册后服务器返回数据
2. 长度：4字节
3. 格式基准：整数
4. 标号：your\_enroll.data
5. 尝试登陆后服务器返回数据
6. 长度：4字节
7. 格式基准：整数
8. 标号：your\_log
9. 分数检测后服务器返回数据
10. 长度：4字节
11. 格式基准：整数
12. 标号：your\_renew
13. 游戏结束后服务器返回排行榜的数据
14. 长度：1000—4400字节
15. 格式基准：数组
16. 标号：your\_renew

### 4.3.3举例

1. 尝试注册后服务器返回数据
2. 表示成功：1
3. 表示用户已存在，注册失败：0
4. 尝试登陆后服务器返回数据
5. 表示成功：3
6. 表示用户不存在：1
7. 表示用户密码错误：2
8. 分数检测后服务器返回数据
9. 表示更新成功：1
10. 表示分数没有突破，不需更新：2
11. 游戏结束后服务器返回排行榜的数据
12. { {1,user1,100},{2,user2,99},{3,user3,66} }
13. { {1,test1,63},{2,test3,56},{3,test56,55},{4,test100,20} }

## 4.4出错和恢复

1. 出错信息：数据库连接失败
2. 出现环境：在服务器运行“echo\_server.js”时出现
3. 含义：数据库没有打开
4. 措施：在服务器中打开数据库
5. 出错信息：游戏后不显示排行榜
6. 出现环境：在前端游戏结束后
7. 含义：前端未联网或服务器未部署正确
8. 措施：检测前端联网状况或在服务器上运行“echo\_server.js”文件

## 4.5求助查询

请发求助信息到QQ：1670867189

# 5.运行说明

## 5.1运行表

正常运行：在服务器上部署好数据库、在服务器shell下运行“echo\_server.js”脚本、在服务器挂载域名的文件夹中放入html前端文件。在浏览器中输入域名，可以正常注册、登陆、游戏、更新成绩、查看排行榜。

## 5.2运行步骤

### 5.2.1运行控制

管理员：服务器和数据库

玩家：游戏前端

### 5.2.2操作信息

1. 操作：在服务器中shell下运行“echo\_server.js”脚本
2. 运行目的：使游戏前端可以和服务器正常通信。
3. 操作要求：要求数据库打开且服务器运行shell。
4. 启动方法：在shell下输入node echo\_server
5. 预计运行时间：启动约1s，后续使其维持运行状态。

### 5.2.3输入/输出文件

1. 用户基本信息表
2. 文件的名称及编号：login
3. 记录媒体：数据库
4. 存留的目录:
5. 用户详细信息表
6. 文件的名称及编号：user
7. 记录媒体：数据库
8. 存留的目录:

### 5.2.4启动或恢复过程

* 1. 登陆云服务器
  2. 打开数据库
  3. 建立login表和user表
  4. 在服务器的域名文件夹中放入html5游戏前端文件
  5. 在服务器shell命令下运行“echo\_server.js”脚本

# 6.非常规过程

* 应急操作：当前端反应无法和服务器建立连接，无法注册、登陆、查询成绩，且服务器shell下仅显示前端连接请求而不打印发给前端的数据时，请退出shell面板，重启“echo\_server.js”脚本。
* 维护人员须知：请不要删除数据库中的“login”表和“user”表。

# 7.操作命令一览表

* node echo\_server ：
* 打开服务器监听，实现游戏前端和服务器通信。echo\_server指的是“echo\_server.js”脚本文件。它调用了socket.io框架，调用了处理数据库的文件“Demo.js”、“loginFcn.js”、“LoginManage.js”、“SkinManage.js”、“UserMannage.js”自定义脚本文件。

# 8.程序文件（或命令文件）和数据文件一览表

【按文件名字母顺序或按功能与模块分类顺序逐个列出文件名称、标识符及说明。】

1. 处理数据库：
2. Demo.js：

建立数据库池，供其他函数调用

1. loginFcn.js：
2. signup:用户注册函数
3. ligon:用户登录函数
4. LoginManage.js：
5. addUser:用户添加
6. selectUser:用户查找
7. UserManage.js：
8. selectUser1:通过id查询user信息
9. insertUser:添加user信息
10. updateScore:更新用户分数
11. selectScore:查询用户分数
12. scoreRank:查询用户排名
13. 处理服务器通信响应：
14. echo\_server.js：
15. our\_enroll注册事件
16. your\_log登录事件
17. your\_renew分数更新事件
18. your\_see
19. 处理前端逻辑：
20. main\_scene.js：
21. game\_scene.js：
22. player.js：
23. scrollview\_load.js：
24. block.js：
25. socket.io.js：

# 9.用户操作举例

详见游戏实现