|  |
| --- |
| **浙江大学城市学院** |
| **概要设计说明书** |
| **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 |



版 本 历 史

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 版本/状态 | 作者 | 参与者 | 起止日期 | 备注 |
| 0.1.0 | 赵伟宏 | 赵伟宏 | 2018.05.20 | 起草 |
| 1.0.0 | 赵伟宏 | 赵伟宏 | 2018.06.27 | 第一次修订 |

# 1引言

## 1.1编写目的

为明确总体设计阶段的计划，同时整理已经做完的工作。

读者对象：开发人员

## 1.2背景

软件工程专业课题：h5网页小游戏——套一套

## 1.3定义

列出本文件中用到的专门术语的定义和外文首字母组词的原词组。

暂无

## 1.4参考资料

张海潘，牟永敏。软件工程导论（第6版） 北京：清华大学出版社，2013

郗晓勇，概要设计说明书， https://blog.csdn.net/beijiguangyong/article/details/5906367 #\_Toc521464967， 2018.5.18

滴滴答答要努力，程序、算法和数据结构的关系，https://blog.csdn.net/qq\_22499377/art icle/details/78326758，2018.5.18

# 2总体设计

## 2.1需求规定

说明对本系统的主要的输入输出项目、处理的功能性能要求，详细的说明可参见附录C。

主要的输入输出项目：玩家的最高分，玩家id

处理的功能性能需求：

最长响应时间：3s。（首次进入可能会花费较长时间）

最大同时在线人数：10人

## 2.2运行环境

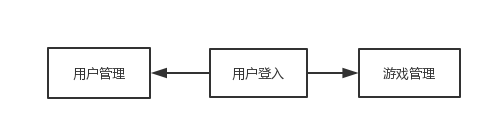
网页

## 2.3基本设计概念和处理流程

说明本系统的基本设计概念和处理流程，尽量使用图表的形式。

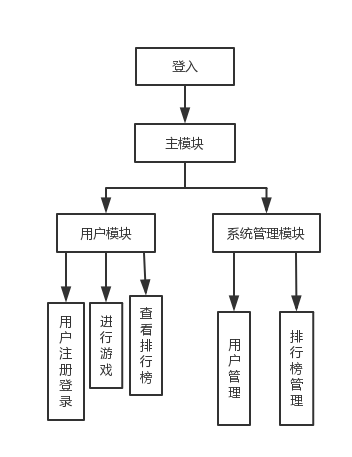
基本设计概念：本系统作为一个网页小游戏，主要为游戏中使用套环套中物品，最终得分将进行排名。

处理流程：



## 2.4结构

用一览表及框图（流程图）的形式说明本系统的系统元素（各层模块、子程序、公用程序等）的划分，扼要说明每个系统元素的标识符和功能，分层次地给出各元素之间的控制与被控制关系.



## 2.5功能需求与程序的关系

本条用一张如下的矩阵图说明各项功能需求的实现同各块程序的分配关系：

|  |  |  |
| --- | --- | --- |
|  | 用户模块 | 系统管理模块 |
| 用户信息管理 | √ | √ |
| 游戏运行管理 |  | √ |
| 排行榜信息管理 |  | √ |

## 2.6人工处理过程

1.系统维护

…

## 2.7尚未问决的问题

黑白盒测试，性能测试，集成测试。

# 3接口设计

## 3.1用户接口

1.用户注册、登录

2.得分上传

3.排行榜查看

## 3.2外部接口

系统基于MySQL的导入和导出功能，方便用户与管理员的数据交换。

## 3.3内部接口

游戏分数的确认上传，排行处理。

用户注册信息上传。

用户登录信息核对。

# 4运行设计

## 4.1运行模块组合

说明对系统施加不同的外界运行控制时所引起的各种不同的运行模块组合，说明每种运行所历经的内部模块和支持软件。

用户注册，登入。运行用户模块

用户进行游戏，上传分数，查看排行榜。运行系统管理模块

## 4.2运行控制

说明每一种外界的运行控制的方式方法和操作步骤。

用户登入需要由系统判断是否为新用户，新用户则注册账号密码，并将最高得分设为0；老用户则只要从数据库确认并提取相关信息。

游戏运行得分需要视情况存入数据库，以便更新排行榜。

查看排行榜是访问数据现存数据，可由系统直接给出。

## 4.3运行时间

说明每种运行模块组合将占用各种资源的时间。

由于是游戏，因此游戏管理模块会占用大部分服务器资源。

用户信息管理只需在登入和游戏得分更新时会占用少量服务器资源。

排行榜信息管理只需在用户查看或有高分产生需要更新时占用资源

# 5系统数据结构设计

## 5.1逻辑结构设计要点

给出本系统内所使用的每个数据结构的名称、标识符以及它们之中每个数据项、记录、文卷和系的标识、定义、长度及它们之间的层次的或表格的相互关系。

### 表汇总

|  |  |
| --- | --- |
| 表名 | 功能说明 |
| login | 用户id和密码 |
| user | 用户id和最高分 |

### 表login

|  |  |  |  |
| --- | --- | --- | --- |
| 表名 | login | | |
| 列名 | 数据类型（精度范围） | 空/非空 | 约束条件 |
| userid | Char[5] | 非空 | （主键）用户id |
| password | Char[5] | 非空 | 用户密码 |
| 补充说明 | 用户登录信息 | | |

### 表user

|  |  |  |  |
| --- | --- | --- | --- |
| 表名 | user | | |
| 列名 | 数据类型（精度范围） | 空/非空 | 约束条件 |
| userid | Char[5] | 非空 | （主键）用户ID |
| score | int | 非空（初始可为0） | 此用户游戏最高分 |
| 补充说明 | 用户游戏得分信息 | | |

## 5.2物理结构设计要点

1.数据存储：用户从客户端输入数据至服务器，再由服务器处理后存入数据库。

2.数据读取：由客户端发送请求至服务器，服务器再从数据库中取出数据传回客户端。

## 5.3数据结构与程序的关系

说明各个数据结构与访问这些数据结构的形式:

login表只在用户注册或登录的时候访问，前一种是为了建立新用户存入用户名和密码，后一种是为了确认用户信息是否正确。

user表只在注册或游戏结束上传成绩时访问，前一种是为了在用户得分初填入初始值0以防止某些不必要的问题，后一种是为了判断是否需要更新数据库，并确定是否进行后续操作。

# 6系统出错处理设计

## 6.1出错信息

用一览表的方式说朗每种可能的出错或故障情况出现时，系统输出信息的形式、含意及处理方法。

若是程序错误，服务器端会直接输出错误相关信息， 程序员可根据此信息进行修改。

若是数据库查询问题，则每种返回结果都有相应不同的返回值，前端可根据不同的返回值进行提醒。

## 6.2补救措施

说明故障出现后可能采取的变通措施，包括：

1. 后备技术说明准备采用的后备技术，当原始系统数据万一丢失时启用的副本的建立和启动的技术，例如周期性地把磁盘信息记录到磁带上去就是对于磁盘媒体的一种后备技术；
2. 降效技术说明准备采用的后备技术，使用另一个效率稍低的系统或方法来求得所需结果的某些部分，例如一个自动系统的降效技术可以是手工操作和数据的人工记录；
3. 恢复及再启动技术说明将使用的恢复再启动技术，使软件从故障点恢复执行或使软件从头开始重新运行的方法。

暂无

## 6.3系统维护设计

说明为了系统维护的方便而在程序内部设计中作出的安排，包括在程序中专门安排用于系统的检查与维护的检测点和专用模块。 各个程序之间的对应关系，可采用如下的矩阵图的形式；

如果是系统超负荷引起的崩溃，则需要更换性能更优秀的服务器。

如果是程序问题，则可以根据错误信息和代码注释进行快速修复。