详细设计-附表

HIPO

文档编号：详设-2-x

定稿日期：2018/5/16

子系统：游戏子系统（服务器端）

子系统概述：服务器端处理游戏结束传入的数据

负责人：张荣阳

实现本子系统需要调动的资源：前端游戏结束的输出数据

子系统设计原则：（应参照系统设计原则稍作变动）

|  |  |  |  |
| --- | --- | --- | --- |
| **日期** | **版本** | **说明** | **作者** |
| **2018.05.16** | 0.1 | 初稿 | 张荣阳 |
|  |  |  |  |

**注：H+IPO图 ——**

***开始工作前请先阅读概要设计文档-附录I-1 体系结构设计报告，尤其是系统总体架构。***

*H代表Hierarchy 层次*

*IPO代表 Input Process Output 输入输出处理*

*所以首先要有这个模块的层次图，再对这个层次图中的各个最小单元进行IPO分析，IPO表中的各项内容解释参考放于同文件夹下的详细设计文稿。*

*简而言之，就是一张结构层次图加上若干张IPO表。*

*下面将附上一份根据这份模板改造的一张IPO图，若不会使用Visio（当然最好是学会，还是挺简单的），可以画一张手绘的层次图然后发给我，由我来代画。*

*HIPO根据我的理解主要还是面向过程的一种设计方法，如果不适应可以自行研究面向对象的设计方法提交相应的文档即可。*

*详细设计的方法，详见书本。*

*六个子系统：*

*1、资源子系统 -YaoT*

*2、账户子系统 -Czf*

*3、评论子系统 -Wsq*

*4、~~前端整合子系统~~*

*5、~~框架自带全权后台管理子系统~~*

*6、~~日常维护子系统~~*

# H图

游戏处理模块

用户历史最高积分

用户最新最高积分

用户套环皮肤最新信息

数据更新（导入数据库）

相关信息提取（从数据库导出）

# IPO表-1-游戏处理模块积分提取

（可做缩进以适应层次图结构）

|  |  |
| --- | --- |
| **游戏处理模块积分提取** | |
| 1. 程序描述：   位于服务器端，关于游戏结束后从数据库将用户上次游戏的积分调出 | |
| 1. 功能：   为当前用户显示曾经游戏的积分记录（仅仅是最高分记录） | 1. 性能（可选） |
| 1. 输入：   用户相关信息认证（有游戏记录） | 1. 输出：   用户曾经游戏的最高记录（仅仅是针对积分） |
| 1. 流程逻辑 | |
| 1. 注释设计 | 1. 限制条件：用户有过游戏记录 |
| 1. 接口：   *用图的形式说明本程序所隶属的上一层模块及隶属于本程序的下一层模块、子程序，说明参数赋值和调用方式，说明与本程序相直接关联的数据结构（数据库、数据文卷）。*  *包括IN和OUT。*  对外接口：  对内接口： | |
| 1. 测试计划 | |
| 1. 尚未解决的问题 | |
| 1. 设计说明 | |
| 编号：01  设计者：张荣阳 | |

# IPO表-2-游戏处理模块信息更新

（可做缩进以适应层次图结构）

|  |  |
| --- | --- |
| **游戏处理模块信息更新** | |
| 1. 程序描述：   位于服务器端，关于游戏结束后从数据库将用户最新游戏的积分和皮肤信息更新 | |
| 1. 功能：   每次用户游戏结束后自动将相应信息更新（积分，皮肤） | 1. 性能（可选） |
| 1. 输入：游戏端传入相应数据 | 1. 输出：将需要更新的信息导入 数据库 |
| 1. 流程逻辑： | |
| 1. 储存分配（可选） | |
| 1. 注释设计 | 1. 限制条件 |
| 1. 接口：   *用图的形式说明本程序所隶属的上一层模块及隶属于本程序的下一层模块、子程序，说明参数赋值和调用方式，说明与本程序相直接关联的数据结构（数据库、数据文卷）。*  *包括IN和OUT。*  对外接口：  对内接口： | |
| 1. 测试计划 | |
| 1. 尚未解决的问题 | |
| 1. 设计说明 | |
| 编号：02  设计者：张荣阳 | |

## PDL

#积分提取

procedure <处理来自/remark的请求> interface<http请求对象> is

begin

if <从request对象的session中得出没有登录> then

重定向到‘用户未登录’页面

if <请求是POST请求> then

form = 调用RemarkForm生成的表单

if <表单验证成功> then

myremark = 调用Remark生成的Remark模型

myremark.id = 表单中'id'字段的数据 //资源id

myremark.subject = 表单中'subject'字段的数据 //用户subject

myremark.score= 表单中'score'字段的数据 //积分score

保存myremark并上传至游戏前端

出现“相应积分：”提示信息

else

form = 空的RemarkForm

返回‘主界面’并显示该用户当前积分

end

#信息更新

procedure <处理来自/remark/delete?id=的请求> interface<http请求对象> is

begin

if <从request对象的session中得出没有登录> then

重定向到‘非法操作’页面

else if <请求不是get请求> then

raise 404异常

else #请求是get请求

if <请求信息不存在> then

返回用户信息未找到出错页面

begin <更新信息>

更新相应信息至数据库

end

出现新的积分排行榜

返回‘主界面’并显示该用户当前积分

end