概要设计说明书

<牵手跑胡子>

文 档 版 本 历 史

|  |  |  |  |
| --- | --- | --- | --- |
| **日期** | **版本** | **说明** | **作者** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**目 录**

[1. 引言 3](#_Toc490232408)

[1.1 目的 3](#_Toc490232409)

[1.2 术语及缩略语 3](#_Toc490232410)

[2. 系统概述 3](#_Toc490232411)

[3. 设计约束 3](#_Toc490232412)

[4. 系统总体结构 3](#_Toc490232413)

[4.1 技术架构 3](#_Toc490232414)

[4.2 系统总体结构逻辑图 3](#_Toc490232415)

[4.3 模块功能汇总表 3](#_Toc490232416)

[5. 系统接口设计 4](#_Toc490232417)

[5.1 系统外部接口设计 4](#_Toc490232418)

[5.1.1 俱乐部 4](#_Toc490232419)

[5.1.2 创建私人房 4](#_Toc490232420)

[5.1.3 进入房间 5](#_Toc490232421)

[5.2 系统内部接口设计 6](#_Toc490232422)

[5.2.1 房间快照 6](#_Toc490232423)

[5.2.2 桌面快照 7](#_Toc490232424)

[5.2.3 断线重连 8](#_Toc490232425)

[6. 模块的结构与功能 8](#_Toc490232426)

[7. 数据字典 8](#_Toc490232427)

[8. 系统出错信息设计 8](#_Toc490232428)

[9. 环境配置 9](#_Toc490232429)

[9.1 开发环境 9](#_Toc490232430)

[9.2 测试环境 9](#_Toc490232431)

[9.3 运行环境 9](#_Toc490232432)

# 引言

## 目的

实现棋牌游戏跑胡子的后端设计

## 术语及缩略语

|  |  |
| --- | --- |
| **术语、缩略语** | **解 释** |
| 跑胡子 | 线下的一种纸牌游戏，玩法类似于麻将 |

# 系统概述

模拟线下的玩法

# 设计约束

系统运行在linux系统下，在window10下使用visual studio 进行开发

# 系统总体结构

## 技术架构

无

## 系统总体结构逻辑图

无

## 模块功能汇总表

|  |  |  |
| --- | --- | --- |
| **模块名称** | **功能说明** | **实现策略** |
| 房间管理 | 游戏中所有房间的管理，包括分配，回收… |  |
| 消息管理 | 对客户端发过来的消息进行处理 |  |
| 房间内部操作 | 同一房间中，玩家操作，广播，结算等 |  |

说明：实现策略是根据分析设计过程中的实现准则进行制定，一般分为：新开发、复用、购买三种方式。如果某个组件实现策略为复用，需注明复用源。

# 系统接口设计

## 系统外部接口设计

### 俱乐部

协议号：1041 检查是否在俱乐部里面

int32 默认0为普通创建房间 俱乐部创建房间的俱乐部id

int32 台费

返回

int32 err 非0的话就没在这个俱乐部里面

int32 台费

int32 0：不能代开 1：可以代开

### 创建私人房

协议号：1010 请求创建私人房

请求：

string type 创建房间类型

int32 局数

int32 番数 0表示不限番

int32 玩法 1是全名堂 2是红黑点 0位默认无

int32 是否参与牌局 0是参与牌局 1不参与牌局 2 扣代理商房费

int32 人数

int32 翻醒0 跟醒1 不带醒2

int32 连庄0 中庄1

int32 默认0 底分2分 1

int32 默认0为200息封顶 1为400息封顶 永州0默认是没有封顶 1是300封顶

int32 默认0为自摸翻倍 1为无

int32 默认0为点炮必胡 1为有胡必胡 2为无

int32 默认0为不能毛胡 1为可以毛胡

int32 默认0为（3人是9胡起胡 4人为3胡起胡） 1为6胡起胡 默认0为十五胡自摸不能胡 1为十五胡自摸能胡

int32 默认0为3胡一等 1为1胡一等

int32 默认0为连庄 1为不连庄

int32 默认0为不加拓 1为加拓

int32 默认0为不能一五十 1为一五十

int32 默认0为不打鸟 1为打鸟 娄底

int32 默认0为普通创建房间 俱乐部创建房间发俱乐部的id

int32 台费 默认0

string 俱乐部名字

int32 可胡示众牌 默认0

返回

int32 err 错误码 -5：服务器即将关闭 -3：在黑名单中 -4：金币不足 -9:已经在房间中了 -10：无可创建的房间 -7:无效操作 -22：不可免费创建 -23 代理商钱不够

int32 是否参与牌局 0是参与 1不参与

int32 roomid

string type 创建房间类型

int32 局数

int32 番数 0表示不限番

int32 玩法 1是全名堂 2是红黑点 0位默认无

int32 人数

int32 翻醒0 跟醒1 不带醒2

int32 连庄0 中庄1

int32 默认0 底分2分 1

int32 默认0为200息封顶 1为400息封顶

int32 默认0为自摸翻倍 1为无

int32 默认0为点炮必胡 1为有胡必胡 2为无

int32 默认0为不能毛胡 1为可以毛胡

int32 默认0为（3人是9胡起胡 4人为3胡起胡） 1为6胡起胡

int32 默认0为3胡一等 1为1胡一等

int32 默认0为连庄 1为不连庄

int32 默认0为不加拓 1为加拓

int32 默认0为不能一五十 1为一五十

int32 默认0为不打鸟 1为打鸟 娄底

int32 默认0为普通创建房间 俱乐部创建房间发俱乐部的id

int32 台费 默认0

string 俱乐部名字

int32 可胡示众牌 默认0

### 进入房间

协议号：1001 请求进入房间

int32 roomid 房间id

int32 version

int32 0 默认 1：ip写成4G

int32 默认0为进入普通房间 俱乐部进入房间发俱乐部的id

int32 0 默认拉入 1：手动输入进入房间

返回：

int32 err 错误码 0：成功 -5：服务器即将关闭 -3：在黑名单中 -2：房间不存在 -6：房间已满人 -19：版本不一致 err不为0时 后面会再发一个俱乐部的会费

int32 seatid1

string roomtype

int32 座位数

int32 番数 0表示不限番

int32 玩法 1是全名堂 2是红黑点 0位默认无

int32 翻醒0 跟醒1 不带醒2

int32 连庄0 中庄1

int32 默认0 底分2分 1

int32 默认0为200息封顶 1为400息封顶

int32 默认0为自摸翻倍 1为无

int32 默认0为点炮必胡 1为有胡必胡 2为无

int32 默认0为不能毛胡 1为可以毛胡

int32 默认0为（3人是9胡起胡 4人为3胡起胡） 1为6胡起胡

int32 默认0为3胡一等 1为1胡一等

int32 默认0为连庄 1为不连庄

int32 默认0为不加拓 1为加拓

int32 默认0为不能一五十 1为一五十

int32 默认0为不打鸟 1为打鸟 娄底

int32 默认0为普通创建房间 俱乐部创建房间发俱乐部的id

int32 会费

string 俱乐部名字

int32 可胡示众牌 默认0

## 系统内部接口设计

### 房间快照

协议号：1002 房间快照 进入房间成功后服务器推的

int32 playercount 玩家个数

循环 玩家个数

{

int32 seatid 座位号

int32 mid

int32 sex

string name

string icon 头像地址

int32 vip

int64 gold

int64 jifen

string ip

}

循环 玩家个数

{

int32 seatid 座位号

string 经纬度(是个json)

}

### 桌面快照

协议号：1003 桌面快照 进入房间成功后服务器推的

int32 房间号

string type

int32 总局数

int32 庄家座位号

int32 当前玩家的座位号

string 当前操作的牌

int32 当前操作的牌的出牌座位号~

int32 类型 1为自摸 0为别人出的牌

int32 桌上的玩家数

循环 玩家数10

{

int32 座位号

int32 mid

int32 是否暂离 1为是 0为否

int32 是否准备

int32 胡息

int32 玩家已出的牌数

循环 已出的牌

{

string 牌

}

int32 玩家已成手牌型数（碰，偎等）

{

int32 类型（0：碰牌，1：一句话 2：特殊的一句话，3：比牌，4：特殊的比牌，5：绞，6：偎， 7：跑， 8：提， 16：臭偎）

int32 牌的数量

循环 牌的数量

{

string 牌

}

}

}

int32 房主座位号~

int32 游戏的状态 1游戏开始，0没开始

int32 桌上的玩家数

循环 玩家的数量

{

int32 座位号

string json数组， 第一个是是否加拓0：否 1：是 第二个是打鸟0：无鸟 1:1鸟 2:2鸟 3:3鸟

}

### 断线重连

协议号：1035 断线重连后的服务器推自己手上的牌

int32 底牌的数量

循环 牌的数量

{

string 牌

}

# 模块的结构与功能

略

# 数据字典

表membercommongame0

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **字段名称** | **字段类型** | **数据域** | **长度** | **备注** |
| mid | int | 无 | 10 | 用户id |
| gold | bigint | 无 | 20 | 金币 |
| vip | tinyint | 无 | 2 | Vip等级 |
| bankpasswd | varchar | 无 | 32 |  |
| bankAssets | bigint | 无 | 10 |  |

表membergame0

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **字段名称** | **字段类型** | **数据域** | **长度** | **备注** |
| mid | int | 无 | 10 | 用户id |
| jifen | bigint | 无 | 20 | 积分 |
| type | int | 无 | 11 | 游戏类型 |

# 系统出错信息设计

本地测试的时候重大bug会导致系统直接荡机，容易发现问题；线上是有日志，出错时可以进行日志查询从而定位问题

# 环境配置

## 开发环境

win10下用Microsoft Visual Studio进行开发，linux系统下编译运行

## 测试环境

linux系统（内核版本：Linux version 2.6.32-220.el6.x86\_64 (mockbuild@c6b18n3.bsys.dev.centos.org) (gcc version 4.4.6 20110731 (Red Hat 4.4.6-3) (GCC) )）

## 运行环境

linux系统（内核版本：Linux version 2.6.32-220.el6.x86\_64 (mockbuild@c6b18n3.bsys.dev.centos.org) (gcc version 4.4.6 20110731 (Red Hat 4.4.6-3) (GCC) )）