相关技术及开发流程介绍

11计算机1班 王立鹏

201103130138

刚接到题目的时候我就在思考怎么实现整个项目，刚开始几天对整体架构没有什么概念，就先写了一份单机版，当然最后的完成版本是从服务器开始写的，单机版作为纪念我没有移除，在完整版中运行登陆界面后如果用户名用single，密码忽略不填则会进入单机版。

好了，下面开始详细说一下完整版的各种技术：

1. server工作在被动模式中，即server不会主动向任何客户端（登陆界面，注册界面，大厅，房间）主动发送任何信息。所有信息均有客户端首先发起，然后server收到信息后判断是何种信息然后做相应处理返回给客户端。
2. 用户信息管理使用成熟的小型关系型数据库mysql，小巧轻便，同时做到server方便安全管理用户信息。
3. 心跳检测：server维护着一个当前在线用户的列表***private******static*** *Vector<User> userList;* 一个当前房间的列表：***private******static*** *Vector<Room> roomList;* ， server有一个线程定期清楚所有在线用户：

/\*

\* 定时清除登录的用户列表，用于"心跳检测"

\*/

**class** ClearUsers **extends** Thread

{

**public** **void** run()

{

**while**(**true**)

{

***userList*.clear();**

**try** {

*sleep*(30000);

} **catch** (InterruptedException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

}

}

}

然后用户在大厅界面每秒会向服务器申请一次当前房间列表：

**class** refreshTable **extends** Thread

{

**public** **void** run()

{

**while**(**true**)

{

**refreshFunc();**

**try** {

*sleep*(1000);

} **catch** (InterruptedException e) {

// **TODO** Auto-generated catch block

//e.printStackTrace();

}

}

}

}

**refreshFunc() 功能向server发送一个申请包，里面包含此用户信息，服务器接受到后给客户端回复一个包，包含着当前房间列表，同时把此用户重新加到在线用户列表中：**

**case** 5://房间列表

{

RoomListNet roo = (RoomListNet) cmd;

**boolean** inUserList = **false**;

**for**(User u:*userList*)

{

**if**(u.getUsername().equals(roo.getUser().getUsername()))

{

inUserList = **true**;

**break**;

}

}

**if**(!inUserList)

{

*userList*.add(roo.getUser());

}

RoomListNet net = **new** RoomListNet(*roomList*);

out.writeObject(net);

out.flush();

**break**;

}

这样就可以大致维护一个在线用户列表，同时**客户端大厅房间列表采用Jtable，获取到房间列表后会自动刷新显示房间列表。**

1. 此项目实现了维护注册唯一性、登陆唯一性、创建房间唯一性、进入房间唯一性。

**注册唯一性：**一个用户名只允许注册一次。

**登陆唯一性：**一个用户名只允许登录一次。

**创建房间唯一性：**一个用户在当前登录状态下只允许同时创建一个房间。

**进入房间唯一性：**当前用户只允许在一个房间中或在大厅中等待。

**5．**在房间界面，棋手图片由性别决定，不同性别的图片不相同。

**6.**无上限限制当前房间数量，可在服务器内存允许下创建尽量多的房间，允许**百万级**用户同时在线，每个房间不限制观看游戏人数数量，无上限限制注册用户数量，可在服务器数据库承受能力下创建尽量多的用户。

**7.**客户端下棋界面设计成丢弃所有用户无效输入（点击棋盘无效位置等），这样就减少了客户端与server之间的网络通讯，减少网络流量，减少服务器负载。

**8.**server会自动统计当前房间人数，如果发现此房间没有用户则会自动销毁此房间，减少垃圾空间占用，同时释放此房间名，允许在线用户再次创建这个名称的房间。

**9.**用户密码储存在数据库中并用md5($password)方式储存，这样即使发生安全事故，比如数据库泄露也不会直接泄漏用户密码，防止发生csdn泄漏用户资料的问题发生。相关链接（<http://www.admin5.com/article/20111224/399415.shtml>）

10.游戏结束后会更新用户积分信息

/\*

\* 下棋结束后更新用户信息

\*/

**private** **void** upUserInfo(String name,**int** Score)

{

String cm = "UPDATE users SET score = "+Score+" WHERE username = \'"+name+"\'";

**try** {

ExecSql.*state*.execute(cm);

} **catch** (SQLException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

}

11.大厅列表中房间状态自动更新：当创建房间后房间状态为**等待玩家**，当开始游戏后房间状态会自动变为**游戏中。。。**所有用户大厅中房间列表都会得到更新

12.服务器配置与客户端配置文件化：在项目文件夹中的config文件夹下有两个配置文件，database.props里面储存服务器用到的相关数据库信息，方便在不改动代码的情况下切换数据库，也可以实现server与数据库server分开独立运行，增加项目稳定性。serverAddress.props是客户端读取的服务器地址，方便用户修改服务器地址。

13.使用了java虚拟机环境检测，可以在不修改源代码的情况下在windows和linux等其他系统下正常工作：

*ImageIcon img =* ***new*** *ImageIcon("res"+File.separator+"chess\_background.png");*