1. 概述

此项目通信方面采用epoll+线程池方式完成，服务器启动后，读取玩家信息（账号，密码，积分），等待客户端连接，当客户端连接时，服务器将此客户端加入epoll监听。等待客户端进行注册或登录，若为登录将客户端输入的信息发送至服务器，存在且密码正确则回弹登陆成功信息，若为注册则将信息写入玩家信息文档。随后判断是否有空桌，若无则向线程池中加入任务，创建对局；若有则直接加入对局。

1. 对局信息传输方式

在服务器端，创建vector table保存桌子信息，每个成员中保存两个玩家信息结构体；

Struct player{

String account；//玩家账号

String password;//密码

Int socekt;//套接字

String info;//传输的信息

}

保存格式大致为table[[player1,player2],[player3,player4]]。

当玩家落子、悔棋等等操作时将信息以特定格式（协议见3）发送到服务器，服务器解析后保存在玩家结构体中，若需转发则转发给同桌的玩家即可（指定套接字）

1. 协议
   1. 登录/注册协议：{way:login/register,account:xxxx,password:xxxx}
   2. 加入房间:{way:join,object:xxx}
   3. 发送棋盘信息：{way:Checkerboard\_Info,arry:a[13][13]}
   4. 下棋：{way:down,local:(x,y),color:black/white}
   5. 悔棋：{way:back,local:(x,y),color:black/white}
   6. 输或赢：{object:xxx,result:win/lose}
   7. 下一把或者退出：
2. 补充：