因为我作了两个版本的客户端,所以你有两种语言可以选择用于编写客户端的代码

分别是**c# 和java**,这里只介绍java版的接口使用方式.

首先,你必须知道**客户端**就是指这次比赛里选手所使用的程序,而**服务端**指的是裁判程序,下面为了方便,我就用客户端和服务端来表示.

系统是通过[socket长连接]实现**客户端和服务端**的数据通信.对每个客户端来说,只需持续监听来自服务端的指令,并作出回复即可.

客户端的控制逻辑是下面这样的:(源码ChessClient\_java\src\SocketClient)

客户端连接服务器(按照课上演示的棋子颜色,先连接的选手被指定为**黑棋**,后连接的被指定为**红棋**)

等所有选手连接上服务器后,开始下棋,服务器会按照规则决定由谁先走,我设定为黑棋先走.这时服务器会发送do\_algorithm类型指令给黑棋的客户端.

客户端持续接收服务端的发来的消息,一旦接收到会先分类,按照指令类型不同作处理

类型分为[timeout,Illegal,do\_algorithm,result]四类,他们分别代表:

timeout:服务器通知你下棋后,没有在限定时间内回复作出决策.

Illegal:你发送的行动序列违法.

do\_algorithm:轮到你走棋了.

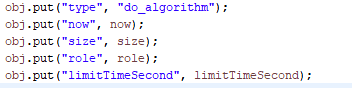
Result:游戏结束,服务端会发送一个结果类型的指令.

你们需要设计算法,根据当前的棋局,返回行动序列

服务器发来的指令是由几个**键值对**表示的,在客户端这边会被转换为HashMap:



如果是轮到你下棋的指令,那么RequestMap里有此次下棋需要的元素,通过get方法获取需要的信息例如: requestMap.get("now") 获取棋局数组.(获得的只是object对象,需要自己转换为相应类型).可通过publicClass\API类下的**getNextEpisodeAPI**方法查看包含的信息.



如上其中棋局是二维的char型数组表示,’a’或者’A’代表”黑棋”,’b’或者’B’代表红棋,’0’代表空.

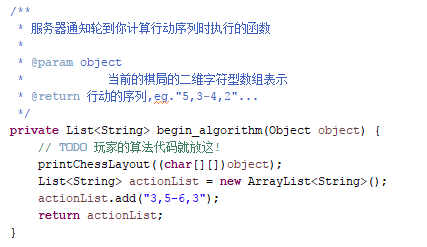
各键的值如下:

now: 当前棋局(char[][] 类型)

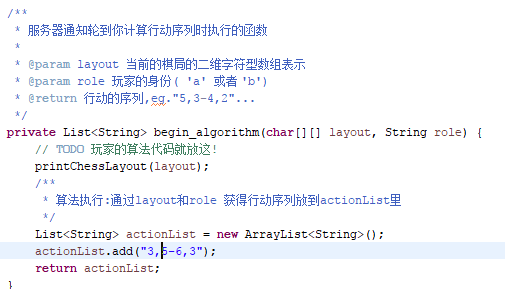
size: 棋局的长宽(常量“8-8”)

role: 你的身份(‘a’ 或者 ‘b’)

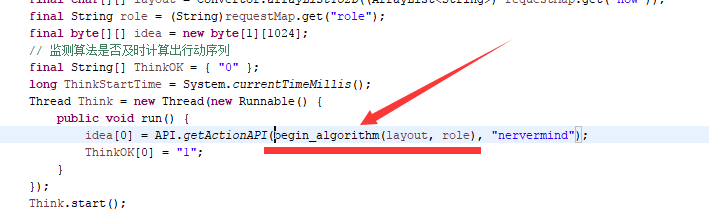
你们需自定以下函数的参数和方法体,把需要的参数传进来,调用算法计算之后返回一个包含行动序列的List<string> 即可.



可以修改为---->



然后在这个位置调用函数.



上面还通过**final** String role = (String)requestMap.get("role")获取了身份.

#代码确实写的非常混乱,水平有限,希望各位理解,有什么问题在群里留言,我看到了就回答.#