

Project #3

2016229082

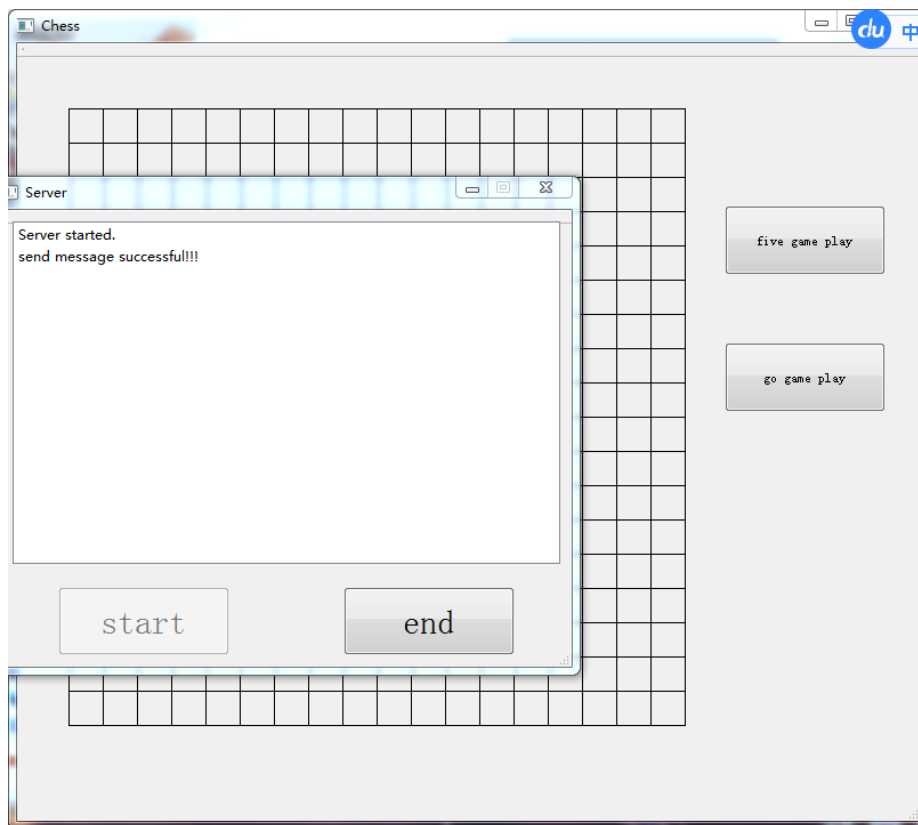
王雨滕

1. 输入和输出

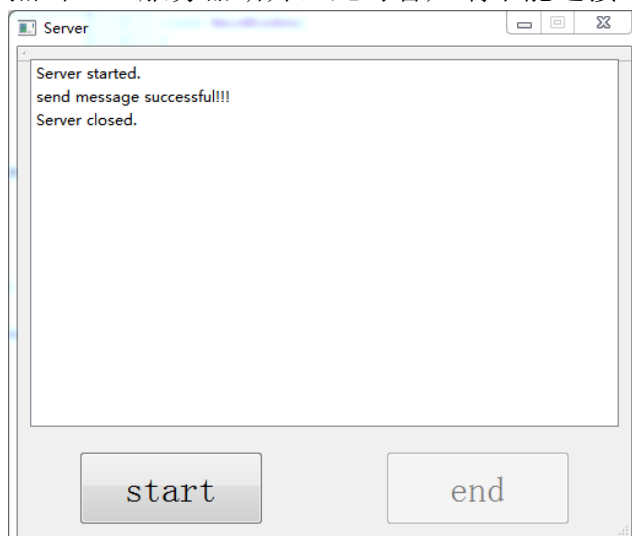
- 服务器端：

运行 myGameServer, 弹出服务器控制框。

在服务框中，点击 **start** 开启服务器，再打开客户端，累计两人并选择相同棋类，就可以开始下棋。



点击 **end** 服务器断开，此时客户端不能连接。

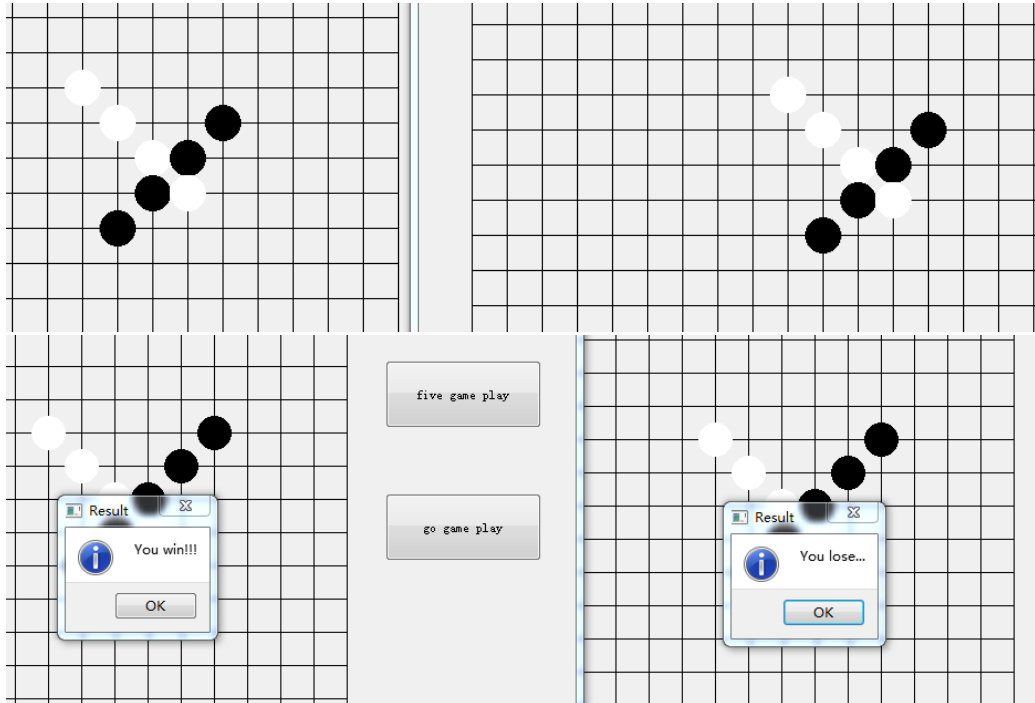


- 客户端

运行 myGame 项目，弹出棋盘游戏窗口，选择棋类型，进行下棋。

当服务器端打开时，两个客户端点击了相同的棋类型，就可以开始下棋。
先选择棋类的用户被分配黑棋，后来的为白棋。黑棋先手。

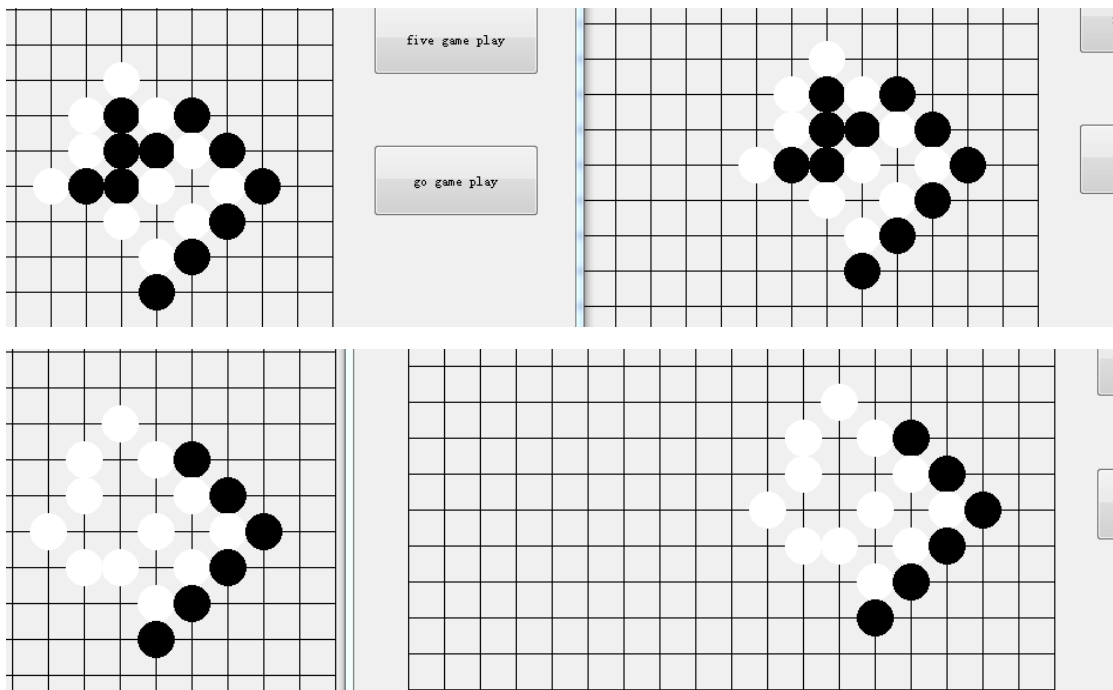
- 五子棋



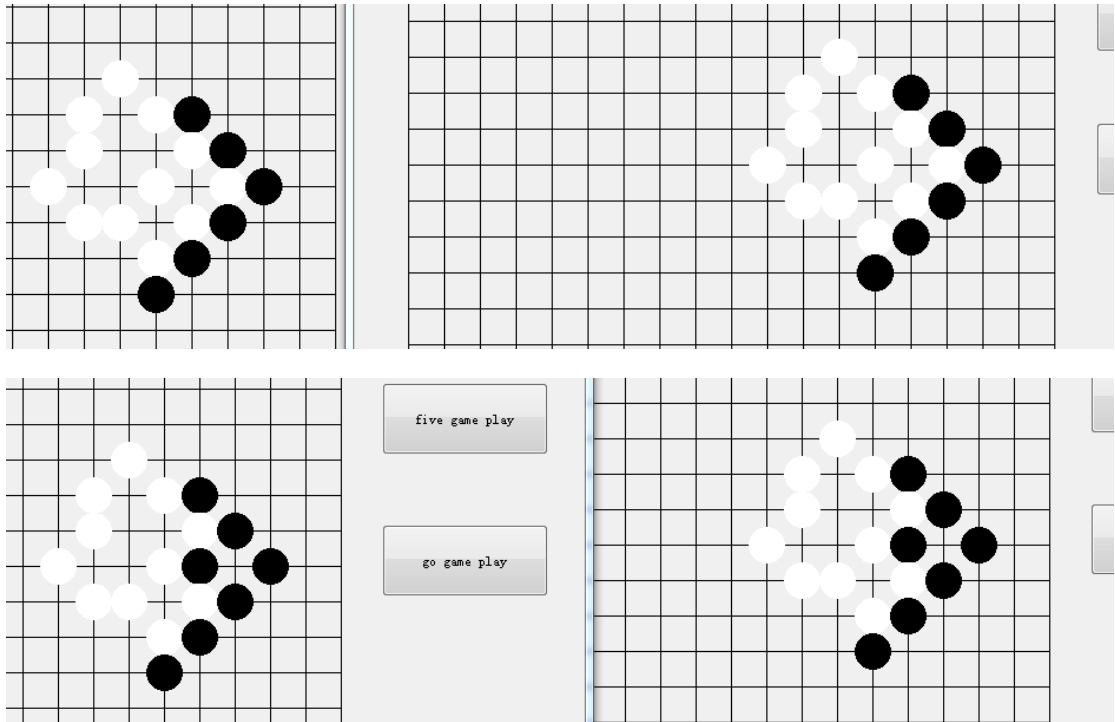
- 围棋

如果周围有气，或者可以反杀的时候，可以落子。

落子后，导致周围反色子没有气的，即吃掉子。



虽然没有气但是可以反杀，就可落子：



2. 相关的数据结构

● 棋子状态

服务器端和客户端棋盘上的都用二维数组 `board[][]` 来保存。值 0 代表该位置没有棋，1 代表黑棋，2 代表白棋。服务器端的二维数组用来判断下棋和吃子等，客户端的二维棋盘数组用来在棋盘上显示棋子。

● 数据通信

服务器端和客户端之间的交互，采用 `Tcp` 建立连接和互相通信。

通信的内容采用 `Json` 类进行传递，把所有想传递的数据写进一个 `Json` 对象中，传给 `tcpSend` 函数，这个函数将 `Json` 对象用 `to_json()` 方法转为字符串，再转为 `QString`，发送即可；接收时，先将 `QString` 的字符串转为 `stdString` 类，再建立带字符串的 `obj` 对象，即可解析此对象的内容。

● 围棋吃子

围棋当 `gameBoard` 文件中的 `checkStone` 函数判断有可移除的棋子后，把被吃掉的棋子保存在 `Array` 类型的 `removeBlock` 中，例如子为 `(x, y)`，则保存为 `100*x+y`。这样便于存储，且传递给 `Json` 对象时更方便。

3. 每部分功能说明

● 服务器端

`Mainwindow` 中进行服务器打开与关闭，保存用户，以及判断是否可以开始游戏。

`Userinfo` 类中建立 `socket`，进行数据的发送与接收，并当接收到对方围棋吃

子信息时，改变自己的 `gameBoard` 中棋子状态。

● 客户端

`Mainwindow` 中响应按钮点击事件，当选择界面中 `Five game play` 按钮，`isFive` 设置为 `true`；反之选择 `Go game play` 按钮，`isGo` 设置为 `true`。`isFive` 和 `isGo` 这两个布尔变量的设置，方便游戏开始和采用不同游戏规则的判断。

`MyWidget` 类为棋盘类，建立和服务器的连接，进行数据的发送和接收，响应鼠标释放事件，绘制和重绘棋盘，