

大作业设计文档

小学期第二周：网络五子棋软件

计 15 2011011300 周若凡

2012/9/2

[WIN sock 和 WIN API 的多线程完成服务器和客户端的后台,利用 Qt 将界面和后台连接,完成一个网络五子棋对战平台。]

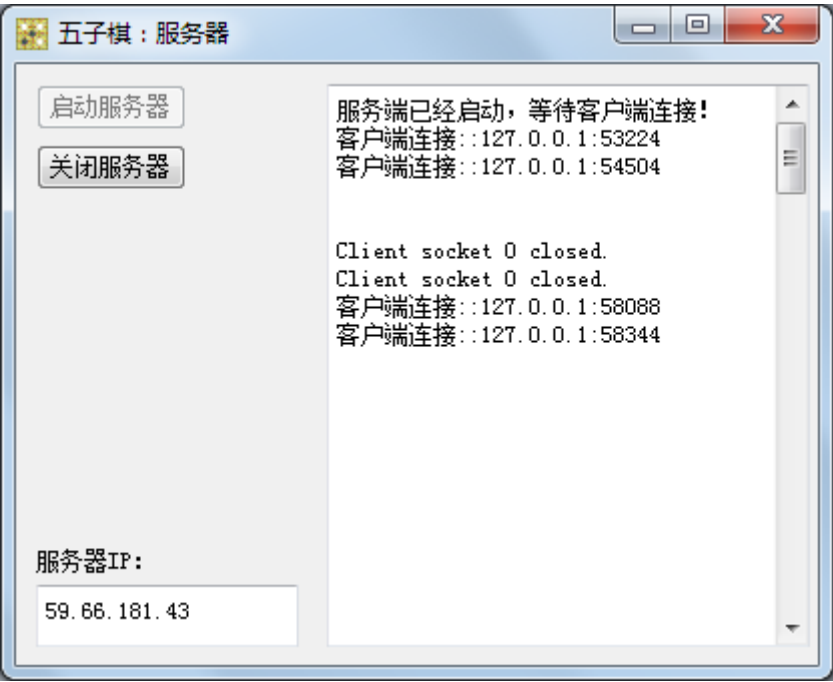
1. 界面及功能介绍

界面介绍



1. 服务器界面

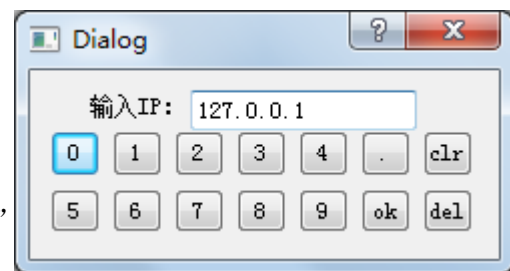
点击“启动服务器”后会打开服务器，并且在左下显示当前主机的 IP 地址；启动过程中会监听连入的客户端并显示客户端的连入（最多两个）和断开。点击“关闭服务器”后可以随时关闭服务器。



2. 客户端_连接服务器和准备开始

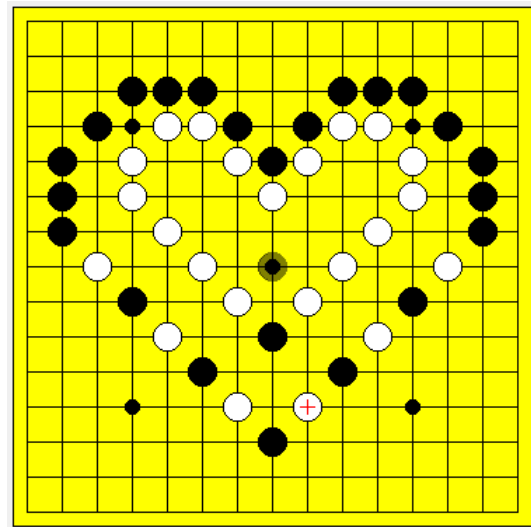
点击“连接服务器”后会出现如右图软键盘，输入 IP 之后客户端会连接到服务器（右边的显示框会提示连接成功与否）。服务器会根据已经连入的客户端的棋子颜色赋予该用户的棋子颜色（如果之前没有连入的客户端则为黑子）。

用户需等待对手出现才可以点击“准备/开始”按钮，只有当两个客户端都点击了该按钮，游戏才会开始。



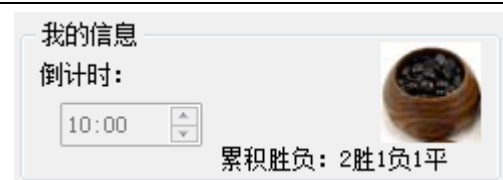
3. 客户端_棋盘信息

左侧棋盘会随时更新目前战况。其中对方最后一个下的子会用红色十字标出（如右图右下的一枚白色棋子上。但如果上一步执行了悔棋则不会出现红色十字）。另外棋盘满足鼠标跟踪，即鼠标移动到相应位置会出现一个透明度较低的棋子提示用户点击鼠标后会下子的位置（右图中间的灰色棋子，鼠标由于系统截屏原因无法显示）。



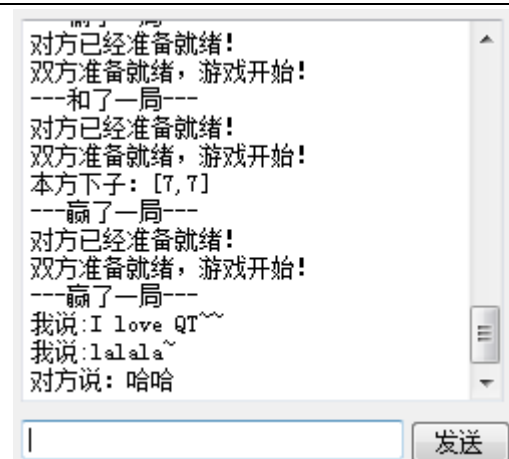
4. 客户端_我与对手的信息

我的信息中会显示倒计时（每局给每个人的时间为 10 分钟，即本方执棋思考时间最多为 10 分钟，到时则自动判定对方胜利。）、我的累积战况以及我所执棋的颜色（如果没有自行交换，则每次对战结束后会交换颜色）。对手的信息不包括战况，其他都会包括。



5. 客户端_消息框

右边的消息框会不停提示双方的战况（更新每人每次下棋的位置，以及交换棋子、悔棋、认输或者胜负消息），也可以通过它即时聊天。



6. 客户端_交换、悔棋、认输、和棋

点击下方这些按钮后服务器会自动询问另一个客户端并通过双方的交涉结果完成相应命令。

7. 客户端_退出

若游戏中客户端与服务器断开连接且若对方仍与服务器有连接，则自动判定对方胜利。

功能介绍：（下划线为附加项）

1. 服务器可以随时创建连接和断开。客户端可以随时连接入服务器。
2. 网络数据传输更新棋盘。
3. 五子棋胜负判断。
4. 服务器能显示自己的 IP 地址。
5. 客户端连入 IP 地址时有软键盘可操作。
6. 客户端的图形界面可以正确、即时更新棋盘状况。可以提示鼠标所在位置的棋子。可以提示上一次对手下的棋子位置。
7. 可以显示当前落子信息。可以即时与对方聊天。
8. 一轮游戏结束客户端会弹出对话框。
9. 可以交换棋子、悔棋、认输和和棋。
10. 有倒计时、可以显示我方所执棋子颜色、我方累积战况。

2. 服务器工作流程

界面主线程：

初始化客户端数据。

点击“启动服务器”后以本机 IP（端口 6000）初始化 Socket，绑定端口。开启子线程 Accept 进行对客户端的监听，同时开启子线程 Worker 接受客户端发来的消息。

子线程 Accept：

进入建立连接循环，将 accept 返回的 Socket 存入相关客户端数据，并通过相关客户端数据更新新加入客户端数据（如棋子颜色等）并将更新消息传给该客户端。

子线程 Worker：

无限循环接受客户端发来的消息。若连接数为 0 则等待 100ms 后进入下一次循环。

通过 select 语句将可读的 Socket 筛选出。通过自己制定的网络通讯协议解读客户端发来的消息并进行处理以及适当的转发。

3. 客户端工作流程

界面主线程：

初始化数据。通过 paintEvent() 事件更新相关按钮是否可点，同时更新棋盘。

mouseMoveEvent() 完成鼠标跟踪棋子位置。

mouseReleaseEvent() 完成点行行为。

相应按钮的 on_xxxButton_clicked() 槽完成相应数据传输给服务器的任务。

点击“连接服务器”通过一个软键盘对话框输入服务器 IP（默认为 127.0.0.1，端口 6000）。初始化 Socket 并与服务器连接。连接成功后开启子线程 Worker 接收客户端发来的消息。

子线程 Worker：

无限循环接收服务器传来的消息，并对消息进行分类处理，更新和修改主线程 MainWindow 中的数据。

4. 网络通信编程框架

服务器端采用多路复用 I/O 模型，通过 select 函数在 Worker 线程中监听两个连接的 socket。

5. 通信协议

客户端与服务器之间传输的信息均为一个名为 MsgInfo 的结构体，它的定义如下：

```
typedef struct{
    char point[15][15];
    int x,y;
    char Msg;
    char message[100];
}MsgInfo;
```

其中 point 为棋盘信息，(x, y) 对应最新下子位置，Msg 为消息类型标记，message 为备注。相应信息解析如下（point 始终存当前棋盘信息则不在下表显示）：

客户端接受信息解析表：

Msg	(x, y)	message	消息解析
A/B	N/A	N/A	表示当前无配对对手且系统分配棋子颜色为黑色（Msg 为 A）或白色（Msg 为 B）
a/b	N/A	N/A	表示当前已有配对对手且系统分配棋子颜色为黑色(Msg 为 a)或白色(Msg 为 b)
O	N/A	N/A	有对手加入
X	对方所下 棋子位置	N/A	对方最新下棋位置更新
W	N/A	N/A	取得胜利
L	N/A	N/A	该局失败
r	N/A	N/A	对方已经准备就绪
R	N/A	N/A	游戏开始
C	N/A	N/A	对方要求交换棋子
c	N/A	N/A	双方交换棋子成功
d	N/A	N/A	双方交换棋子失败
l	N/A	N/A	对方认输
H	N/A	N/A	对方要求和局
h	N/A	N/A	和局
i	N/A	N/A	请求和局失败
T	N/A	N/A	对方超时
U	N/A	N/A	对方要求悔棋
u	N/A	N/A	悔棋成功
q	N/A	N/A	悔棋失败
o	N/A	N/A	对方断线
S	N/A	对方所发信息	对方发送聊天消息

服务端接受信息解析表：

Msg	(x, y)	message	消息解析
X	下棋位置	N/A	该客户端最新下棋位置
R	N/A	N/A	该客户端准备就绪
C	N/A	N/A	该客户端要求交换棋子
c	N/A	N/A	该客户端同意交换棋子
d	N/A	N/A	该客户端不同意交换棋子
L	N/A	N/A	该客户端认输
H	N/A	N/A	该客户端要求和局
h	N/A	N/A	该客户端同意和局
i	N/A	N/A	该客户端不同意和局
T	N/A	N/A	该客户端超时
U	N/A	N/A	该客户端要求悔棋
u	N/A	N/A	该客户端同意对方悔棋
q	N/A	N/A	该客户端不同意对方悔棋
S	N/A	所发信息	该客户端发送聊天消息

6. 总结

本次大作业主要是用 winsock 完成一个对战平台之间信息的传递，并用之前学过的 Qt 使之可视化。

由于扩展的功能较多，制定连接协议对相互传发的信息解析带来了一定的困难，但也保证了信息的正确性，也拓展了游戏的可玩性。所以在服务器和客户端之间制定一个合理的协议是相当重要的。

Socket 代码虽然长，但一般的构架只需要简单的拷贝代码就可以了。

在写程序，无论是小作业还是大工程，都必须先考虑清楚程序各个部分的功能以及需要创建哪些重要变量，以及使用什么架构来完成整个程序。

实践出真知。

附图，简单的游戏流程图：

