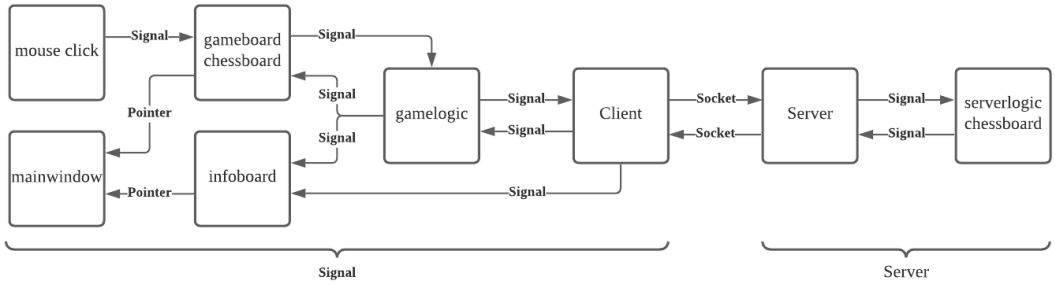
网络对战军棋小游戏设计文档

容逸朗　2020010869　rongyl20@mails.tsinghua.edu.cn

1. 客户端、服务器端的工作流程

工作流程如下图所示。



1.1　开始游戏

当一方开启服务器（客戶端接口自动连接服务器），另一方连接服务器后，服务器会向客户端发去先手信息和棋盘信息，客户端收到信息后会更新棋盘和先手方标记。

玩家可以点击菜单栏“Play – Start”，此时客户端会向服务器发射信号，当服务器接收到双方的信号时，会向两个客户端发出游戏开始的信号，同时在 ServerLogic 类开始计时（新的回合）。

1.2　正常游戏过程

玩家点击游戏面板上的棋子后，棋子对应的 ClickableLabel 会向 Gameboard 发出信号，此时 Gameboard 会判断游戏是否已经开始 / 结束，若游戏仍在进行，则向 GameLogic 发出信号，而 GameLogic::moveChess 会对下棋方进行判断，若是其回合，则会判断此次下棋是本回合内第几只棋，并分别调用 GameLogic::moveChess1 和 GameLogic::moveChess2 。

GameLogic::moveChess1 会判断棋的状态，若为未翻开（chess.status = 0）则进行翻开操作。否则会判断颜色是否和当前执棋方相同，若相同则会进行 GameLogic::dfs 和 GameLogic::checkGameboard ，找出这只棋子的合法移动方案，并更改棋子的状态。（操作面板中，选中的棋子会出现黄色外框，可合法移动的格子则会标记为黄色虚线外框）

GameLogic::moveChess2 会判断此次移动是否合法，若不合法则向执棋方作警告。若合法，则通过 Client::sendDataSlot 向 Server 发信号（具体编码如附录所示）

Server 收到信号后，会对信号进行分类，若为下棋动作，则透过 ServerLogic 更改伺服器端棋盘及地雷数，并开始下一回合，数据回传至 Client 端进行更新。

Client 收到信号后，会对信号进行分类，并更新对应的信息（如：Chessboard,

GameLogic, Infoboard 等），且删去旧有的格子标记，一回合结束。

1.3 非正常游戏过程

1.3.1 投降

玩家点击主页面菜单栏“ Play – Surrender ”后，会通过 Client 向 ServerLogic 传出投降信息，若回合数大于 20，则投降成功，否则返回警告信息。

1.3.2 超时

ServerLogic 检测到倒计时归零后，会自动开启新的回合，并通过 Server 向超时方的 Client 发出警告。若超时次数到达 3 次，则游戏自动结束。

1.4 游戏结束

每回合结束后，ServerLogic:: checkMovable 会对新回合下棋方作检查，若无棋可走，则自动判负。除此之外，ServerLogic 也会对超时次数作记录，超时次数大于等于 3，自动判负。取得军旗获胜由 GameLogic::moveChess2 函数判断，若玩家排完所有雷并成功取得对方军旗，则为胜利。

服务器接收到一方胜利的报文后，会向双方发出游戏完成及胜负信号，游戏结束。

2 服务器、客户端通信

2.1 网络通信编程框架

利用了 QTcpServer 和 QTcpSocket 进行传输。

2.2 通信协议

2.2.1 传输层协议

使用 TCP / IP 协议。其中 IP 端口设置为 8888。

2.2.2 应用层协议

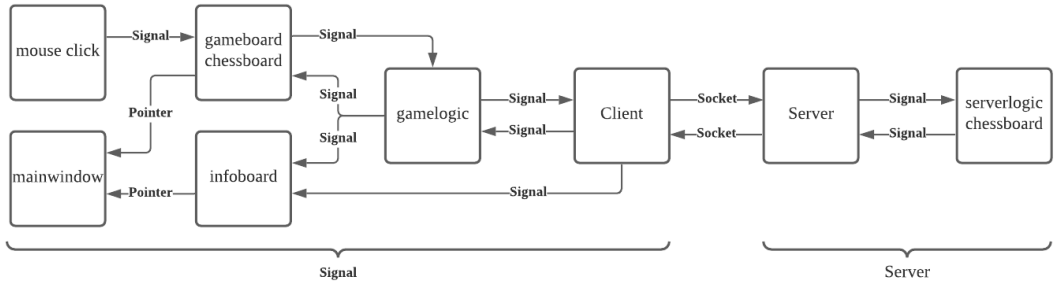
数据以行为单位传输，每行行首为编码，参数之间以空格划分。（如：“c 3 2 0 1 5”等）

編码	格式	作用
b	int, int	更改队伍颜色
c	int, int, int (int, int)	更改格子上的棋子
d	int, int	翻棋
e	/	结束回合
f	int	投降
g	int	游戏结束
i	int	更改玩家序号
l	int	地雷爆炸
m	int	一般信息
o	/	更改玩家
p	int	当前玩家
q	int	先手玩家
s	/	游戏开始
t	int	倒计时剩余时间
w	int	警告信息
z	/	游戏结束标记
0	QString	对战记录
1-6	int, int (int, int, int, int, int)	对战记录

3 信号与槽机制设计

为了在客户端及服务器端内的文件之间传送数据，本游戏采用了信号与槽机制。

结合 2.2.2 中表格，当客户端 / 服务器接收到信息后，会将其解码并传到相应的文件内。（如下图所示）




```
connect(gamelogic, &GameLogic::sendData, client, &Client::sendDataSlot);
connect(serverlogic, &ServerLogic::sendData, server, &Server::sendDataSlot);

connect(client, &Client::changeID, this, &MainWindow::newID);
connect(client, &Client::gameEnded, this, &MainWindow::onGameEnded);
connect(client, &Client::colorDecided, infoboard, &Infoboard::getColorDecided);
connect(client, &Client::playerDecided, infoboard, &Infoboard::getPlayerDecided);
connect(client, &Client::timeRemainDecided, infoboard, &Infoboard::getTimeRemainDecided);
connect(client, &Client::sendMessage, infoboard, &Infoboard::addGameLog);
connect(client, &Client::sendMessage1, infoboard, &Infoboard::addGameLog1);
connect(client, &Client::sendMessage2, infoboard, &Infoboard::addGameLog2);
connect(client, &Client::sendMessage3, infoboard, &Infoboard::addGameLog3);
connect(client, &Client::sendMessage4, infoboard, &Infoboard::addGameLog4);
connect(client, &Client::sendMessage5, infoboard, &Infoboard::addGameLog5);
connect(client, &Client::sendMessage6, infoboard, &Infoboard::addGameLog6);
connect(client, &Client::firstPlayer, infoboard, &Infoboard::setFirstPlayer);
connect(client, &Client::onChangePlayer, gamelogic, &GameLogic::changePlayer);
connect(client, &Client::changeChessNULL, gamelogic, &GameLogic::onChangeChessNULL);
connect(client, &Client::changeChess, gamelogic, &GameLogic::onChangeChess);
connect(client, &Client::start, gameboard, &Gameboard::show);
connect(client, &Client::gameEnded, gameboard, &Gameboard::gameEnded);
connect(client, &Client::start, gamelogic, &GameLogic::checkIsFirst);
connect(client, &Client::firstPlayer, gamelogic, &GameLogic::setPlayer);
connect(client, &Client::changeColor, gamelogic, &GameLogic::setNewColor);
connect(client, &Client::mineBoomed, gamelogic, &GameLogic::onMineBoomed);

connect(server, &Server::selfConnect, client, &Client::setIP);
connect(server, &Server::changeChessNULL, serverlogic, &ServerLogic::onChangeChessNULL);
connect(server, &Server::changeChess, serverlogic, &ServerLogic::onChangeChess);
connect(server, &Server::surrender, serverlogic, &ServerLogic::getSurrender);
connect(server, &Server::endGame, serverlogic, &ServerLogic::endGame);
connect(server, &Server::flopChess, serverlogic, &ServerLogic::flopChess);
connect(server, &Server::initialize, serverlogic, &ServerLogic::initialize);
connect(server, &Server::timerStart, serverlogic, &ServerLogic::timerStart);
connect(server, &Server::endTurn, serverlogic, &ServerLogic::changePlayer);
connect(server, &Server::mineBoomed, serverlogic, &ServerLogic::onMineBoomed);
```

4 GUI 界面设计


主视窗设计
菜单
chess Connect Play
Chess
双方颜色、剩余地雷数
倒计时
下棋方标记
游戏记录
棋盘
Player Stats
You (0)
Enemy (1)
Time left
19
[Turn118] The red Captain at C2 wiped out the blue Lieutenant at D1.
[Turn119] The blue Bomb at K1 and the red Landmine at K4 perish together!
[Turn120] A red Sapper moves from E2 to F1.
[Turn121] The blue Major General at E1 wiped out the red Sapper at F1.
[Turn122] A red Bomb moves from I3 to H2.
[Turn123] The blue Major General got the flag!
[Server] Game over! You Lose!
[Server] Client #1 disconnected!


服务器介面
连接服务器介面
Host IP: 192.168.0.140
Cancel
OK
关闭伺服器
开启伺服器
取消、断开连接
Enter IP:
Cancel
OK
服务器IP输入框
连接伺服器