

项目报告

局域网内双人聊天游戏程序

信息科学技术学院

程靖博

沈金亮

付忠旺

2022.6.30

目录

1	引言	3
2	项目说明	3
2.1	登入界面	3
2.2	聊天界面	4
2.3	中国象棋	4
2.4	五子棋、围棋	5
2.5	贪吃蛇	5
3	核心代码	7
3.1	聊天界面	7
3.2	棋类	7
3.3	贪吃蛇	10
4	分工与思考	12

1 引言

本程序基于 qt5.14.2 开发，支持 window 系统，基于 udp 协议可在局域网内实现通信，双人游戏等功能。该程序为轻量级，打包为可执行程序后，仅通过输入对方 ip 地址即可实现联机游戏功能，适合同学之间的游戏娱乐使用。

2 项目说明

2.1 登入界面

通过输入对方的 ip 地址和端口号来进行连接

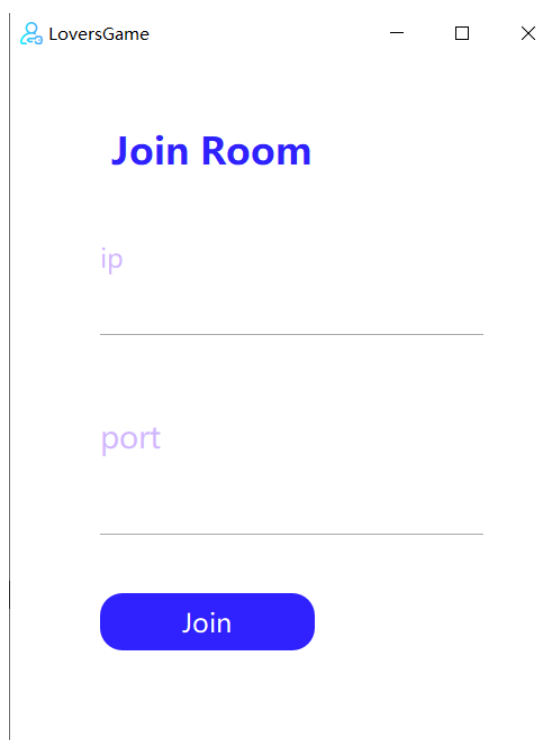


图 1: 登入界面

2.2 聊天界面

输入发送的话后，点击 Send 按钮即可发送到对方客户端上，下方有四个小游戏入口，分别为象棋，五子棋，围棋，和双人贪吃蛇，点击即可与对方进入双人联机游戏。

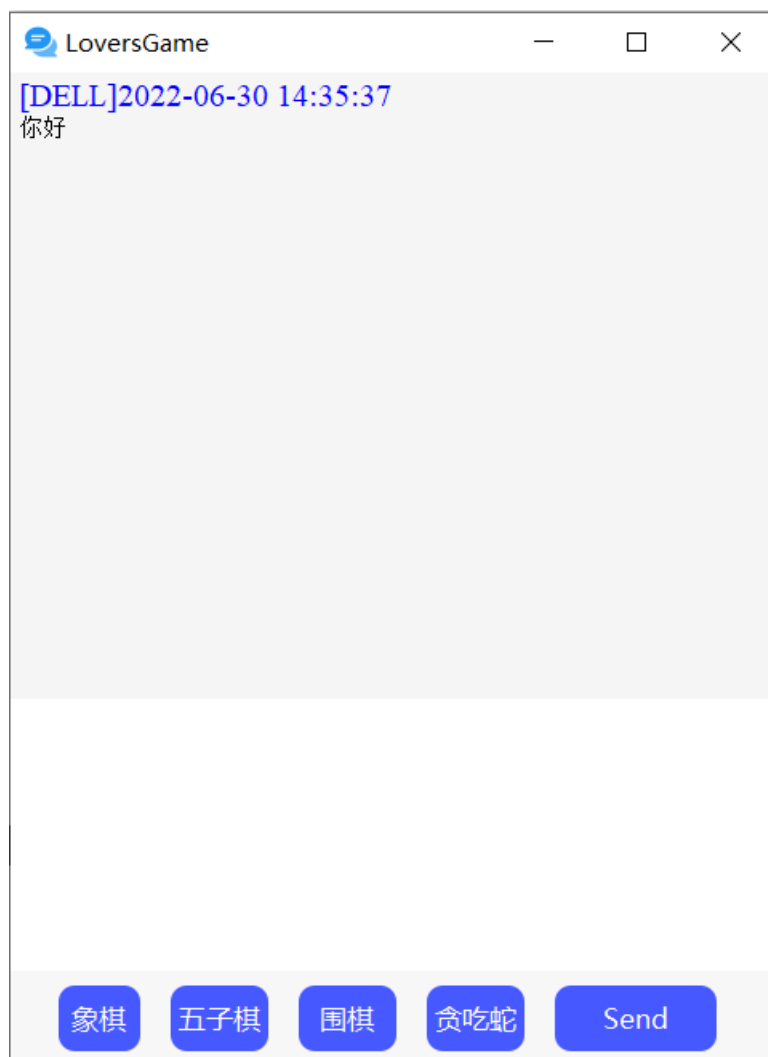


图 2: 聊天界面

2.3 中国象棋

点击下方的中国象棋按钮双方即刻进行游戏，其中先打开按钮的一方为先手

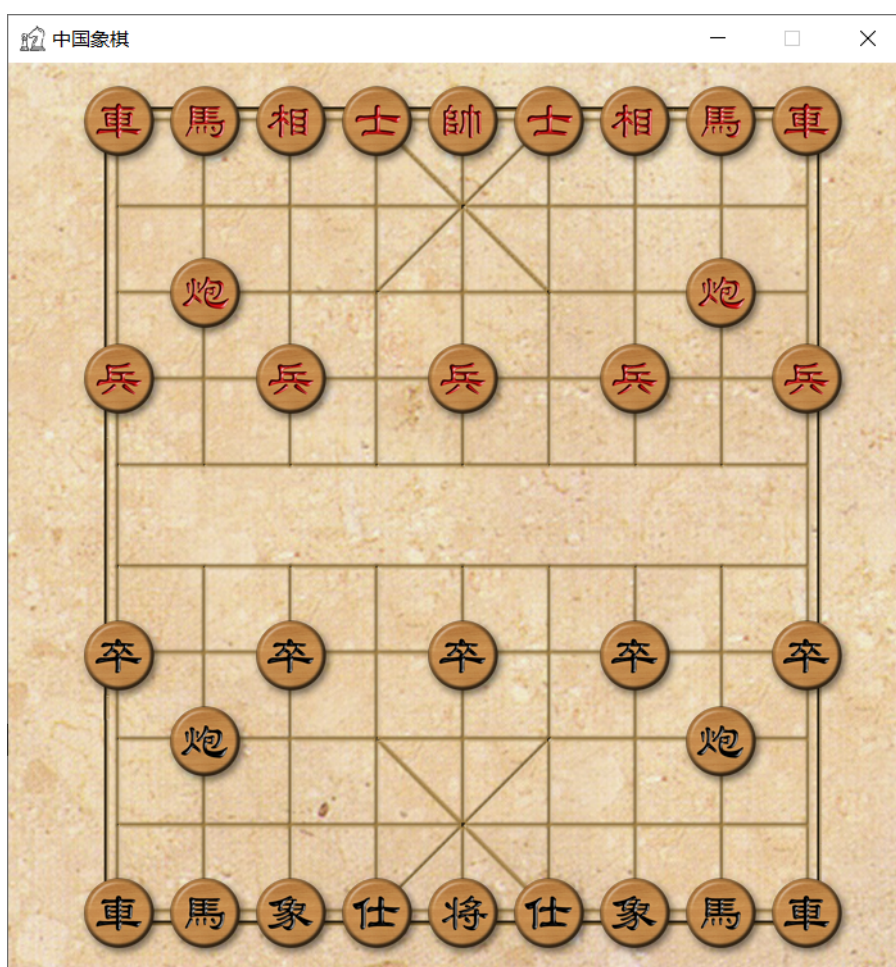


图 3: 中国象棋

2.4 五子棋、围棋

同象棋的打开方式，且五子棋与围棋打开方式类似，先打开的一方为先手

2.5 贪吃蛇

进入后有自定义速度以及大小，确定之后即可进入双人贪吃蛇游戏中

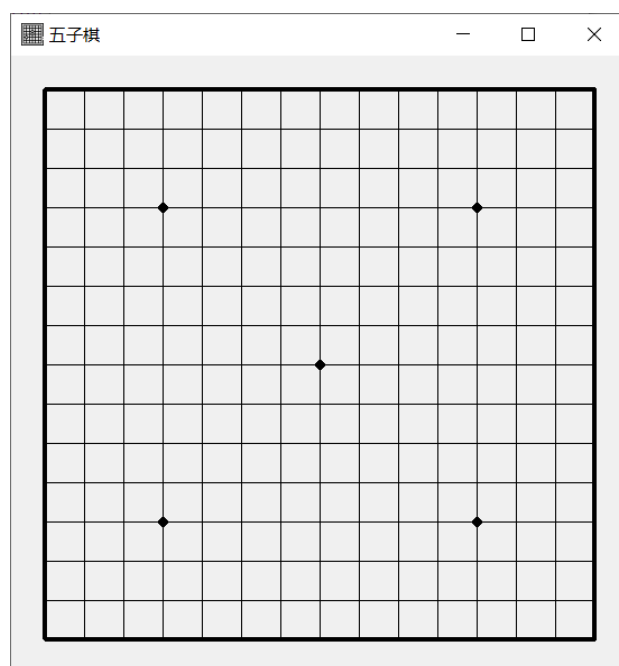


图 4: 五子棋

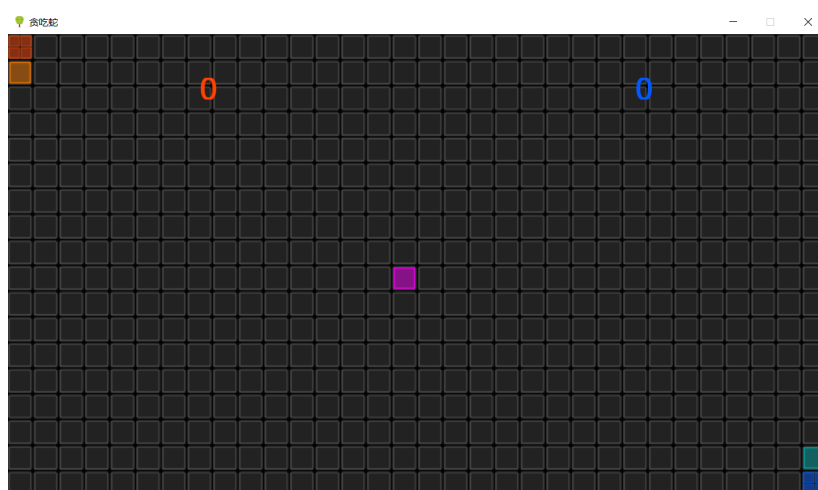


图 5: 贪吃蛇

3 核心代码

3.1 聊天界面

通过槽函数监听信息输入，接受信息后，执行对应的动作

```
1 udpSocket = new QUdpSocket(this);
2 udpSocket->bind(port, QUdpSocket::ShareAddress | QUdpSocket
    ::ReuseAddressHint);
3 connect(udpSocket, SIGNAL(readyRead()), this, SLOT(
    processPendingDatagrams()));
```

3.2 棋类

对方下过棋之后发送当前棋的状态，以此更新对方界面的棋盘信息，以及当前的轮次

```
1 void GoBangGame::recieveGoBang() {
2     while(udpSocket->hasPendingDatagrams()) {
3         qDebug() << "Gorecieve";
4         QByteArray datagram;
5         datagram.resize(udpSocket->pendingDatagramSize()
6             );
7         udpSocket->readDatagram(datagram.data(), datagram
8             .size());
9         QDataStream in(&datagram, QIODevice::ReadOnly);
10        int type;
11        in >> type;
12        switch (type) {
13            case GoGameState:{
```

```
12         for (int i = 0; i < 15; i++) {
13             for (int j = 0; j < 15; j++) {
14                 in >> Five->board[i][j];
15             }
16         }
17         break;
18     }
19     case GoGameClose: {
20         udpSocket->close();
21         close();
22         break;
23     }
24     case GoGameWin: {
25         QMessageBox::information(this, "游戏结束",
26                                 "失败");
27         udpSocket->close();
28         close();
29         break;
30     }
31     update();
32     if (currentStone==WHITE)
33         currentStone=BLACK;
34     else if (currentStone==BLACK)
35         currentStone=WHITE;
36 }
37 }
```

其中围棋中的逻辑用到了广度优先搜索，搜索棋是否有气

```
1 bool GoGame::go_bfs(int x,int y,StoneType current,std:::
```



```
queue<point> &save){  
2    //广度优先搜索，如果当前一片区域的棋子没有气，则  
        remove  
3    int dir[4][2]= {{-1,0},{1,0},{0,-1},{0,1}}; //记录四  
        个方向  
4    point temp;  
5  
6    //把当前坐标加入队列  
7    temp.x=x,temp.y=y;  
8    q.push(temp);  
9    save.push(temp);  
10   while(!q.empty())  
11   {  
12       //获取队头坐标  
13       temp=q.front();  
14       q.pop();  
15       x=temp.x;  
16       y=temp.y;  
17       vis[x][y]=true;  
18       //对当前坐标的周围进行查看  
19       for(int i=0; i<4; i++)  
20       {  
21           int u,v;  
22           u=x+dir[i][0];  
23           v=y+dir[i][1];  
24           //坐标 u,v 没有棋子 代表有气存在 返回true  
25           if(this->Go->checkStone(u,v)!=BLACK&&this->  
                Go->checkStone(u,v)!=WHITE)  
26               return true;  
27           //否则 将坐标u, v加入队列
```

```
28         else if (this->Go->checkStone(u,v)==current
29                 &&!vis[u][v])
30         {
31             temp.x=u;
32             temp.y=v;
33             q.push(temp);
34             save.push(temp);
35         }
36     } //end for
37     return false;
38 }
```

3.3 贪吃蛇

贪吃蛇的界面利用了 OpenGL，实现了阴影效果，其中一个函数举例如下

```
1 void SnakeGame::drawSnake() {
2     bool head;
3     int length, idx;
4
5     head = true;
6     length = red_player->snake.size();
7     idx = 0;
8     for (auto coord : red_player->snake) {
9         float percent = static_cast<float>(idx) /
10                        static_cast<float>(length);
11         QVector2D scale(tile_width, tile_height);
12         QVector3D color = Mix(red_player->front_color,
13                               red_player->back_color, percent);
```

```
12         drawTile(QVector2D(coord.x() * scale.x() - scale
        .x() * 0.5f, coord.y() * scale.y() - scale.y
        () * 0.5f), QVector4D(color, 1.0f), head);
13
14         head = false;
15         ++idx;
16     }
17
18     head = true;
19     length = blue_player->snake.size();
20     idx = 0;
21     for (auto coord : blue_player->snake) {
22         float percent = static_cast<float>(idx) /
        static_cast<float>(length);
23         QVector2D scale(tile_width, tile_height);
24         QVector3D color = Mix(blue_player->front_color,
        blue_player->back_color, percent);
25         drawTile(QVector2D(coord.x() * scale.x() - scale
        .x() * 0.5f, coord.y() * scale.y() - scale.y
        () * 0.5f), QVector4D(color, 1.0f), head);
26
27         head = false;
28         ++idx;
29     }
30 }
```

4 分工与思考

程靖博：负责代码编写以及整体思路

沈金亮：负责写总结报告

付忠旺：负责界面设计

项目的思考：通过这次项目，学习到了有关互联网的知识，对网络编程有了一个基本的认识，也对 qt 的编程更加熟悉，也通过一个一个游戏的搭建，对自己的代码能力有了一个提升。不足的是没能把功能做的更加齐全。