

第 2 周大作业说明文档

肖易佳 2018011347 计 83

实验目的

通过实现网络对战国际象棋软件，进一步巩固 qt，同时，学会使用 socket 实现网络传输数据。

工作流程

连接：

基本变量初始化后，选择程序扮演的角色，不妨设为服务器，则点击服务器后，程序初始化 TCP socket 和 TCP server，并且进入监听状态，监听任何地址的连接请求，如果接收到连接请求，则 newConnection 的这一信号会发出，用 connect 函数连接信号函数与建立连接的槽函数；进一步 acceptConnection 即可建立连接。关于在连接成功前，可取消连接的功能，主要操作将监听器 (TcpServer) 设为关闭即可。从客户端的角度来看，客户端相对简单一些，调用 connectToHost 函数，向对应的主机地址发送请求，接受请求、成功建立连接后，即可实现数据交互。

数据互通：

由于棋盘格子的数目是确定的，每一个格子上，是否有存在棋子，（如果存在棋子）棋子的状态，以及棋子的颜色，这几部分的数据个数、长度等是固定的，所以可以统一地将当前棋局转变成一个字符串，之后用 QByteArray 将其编码，最后调用 socket.write() 将数据发送给对方，对方在接受到数据包之后，内部信号函数会 emit 信号，（负责读取信号的）槽函数接收到数据，并且解析收到的数据，而后更新本地的棋盘，实现双方的实时对战。

判负：

关于超时判负和认输，超时判负部分使用 QTimer 类，可以实现计时，并且在时间结束后发出信号，而认输的点击“认输”按钮的效果，等同于 timeout 这一信号，这两个信号都会通过 QObject::connect()，调用 sendData() 这一函数，并且发送的数据的头部会插入一个状态参数（一个字符），"F"代表本方已经认输。由此，可以实现双方的状态以及棋盘布局的同步。

合法性：

主要内容的最后一部分，关于落子合法性的问题，在写工程的过程中，我认为最难的是 Pawn 的部分，因为 Pawn 不仅有升变，还有直走斜吃等特性，对于这些特殊的情况，

耐心分类讨论，不重不漏（按照 Pawn 所在的棋盘行数来讨论），即可解决

关键函数：

```
void MainWindow::mousePressEvent(QMouseEvent *mevent)
```

//检测鼠标点击的位置，并且通过点击的起始和终止，来判断走子情况，以及阻止不合法的走子

```
void MainWindow::init()
```

//初始化棋盘界面，根据国际象棋的规则，棋盘布局将黑白两方的棋子摆好

```
void MainWindow::load()
```

//读取 log 文件，并且根据规定的格式将数据载入并显示和同步

```
void MainWindow::clear()
```

//将棋盘完全清空，在读取数据前要进行一次此操作

```
void MainWindow::initServer()
```

//初始化服务器

```
void MainWindow::connectHost()
```

//连接到主机（客户端调用）

```
void MainWindow::acceptConnection()
```

//接受连接（服务器端调用）

```
void MainWindow::sendData(QString ins)
```

//发送数据

```
void MainWindow::revData()
```

//接受数据