

# Land Battle Chess 设计文档

徐磊\* 2013011344

2014 年 9 月

## 1 功能介绍

Land Battle Chess 是一个支持网络对战的军棋游戏。

## 2 设计分析

### 2.1 网络通信

网络通信部分使用 socket API，在程序中使用 EzServer 与 EzClient 两个类进行了封装。这两个类都支持阻塞的发送消息与多线程的接受消息，利用 QT 的信号槽机制来实现线程之间的通信。EzServer 除了支持收发消息外，还提供了创建监听的函数。EzClient 则包括了连接服务器的函数。

### 2.2 棋盘模型

棋盘模型是单独的 ChessModel 类，这个类包括了双方玩家共 50 个棋子的坐标，以及双方玩家对对方棋子的猜测信息。通过在网络上直接传递 ChessModel 来实现信息的交互。ChessModel 可以根据玩家的信息生成用于绘制的棋盘状态（包括自动旋转棋盘，使得用户位于屏幕的下方）。支持判定胜负，设置初始局面，判断是否可以移动，处理旗子移动等。

---

\*leopard.lie@gmail.com

## 2.3 棋盘绘制

棋盘绘制类 ChessBoard 继承自 QLabel，可以根据棋盘模型生成的绘制数据进行绘制。另外，ChessBoard 还监控鼠标点击事件，将对应的旗子坐标以消息的形式发送给控制器。

## 2.4 控制器

MainWindow 除了包含一个界面用来放置棋盘和其它必要的按钮外，主要的功能是控制器。在 Mainwindow 中，需要将创建服务器、连接服务器的消息关联到 EzServer、EzClient 恰当的函数上，根据 EzServer、EzClient 接收的数据来更新本地的 ChessModel，并将 ChessModel 生成的绘制信息发送给 ChessBoard。MainWindow 需要处理 ChessBoard 发送的按键信号，以及自己保存的历史按键信息，来做出正确的操作。另外，MainWindow 中还包括一套复杂的计时机制。

## 2.5 其它

根据要求，需要实现一个奇怪的输入 IP 的 Dialog，写作 MyInputDialog。需要实现用户信息的传递，包装了 User 类包括 ID 和图片，UserWidget 用于显示 User 信息。

# 3 通信协议

系统自动将服务器端设为玩家 A，客户端设为玩家 B。一条有效地通信长度永远是  $\text{sizeof}(\text{ChessModel})+1$ ，第一个字符若为 A 或 B，表示 A 玩家或 B 玩家走了一步，之后的  $\text{sizeof}(\text{ChessModel})$  的长度为 ChessModel。若第一个字符为 a 或 b，表示 A 玩家或 B 玩家完成了棋子的布局，等待开始游戏，之后有一个 ChessModel 来表明棋盘的状态。若第一个字符为 T，表示求和，而第一个字符为 t 表示同意求和。第一个字符为 -1 表示认输。

发送用户名和头像的协议，首字母为 U 表示发送用户名，之后  $\text{sizeof}(\text{ChessModel})$  的长度为用户名（默认用户名少于该大小，用 0 补足长度）。首字母为 I 表示与头像相关的信息，首先发送只包含 5 个字节有效信息的数据，用 0 补足长度，该信息的后四位是一个整数表示图片的大小。随后以 I 开头的若干条数据来发送该图片。

# 4 总结

Land Battle Chess 基本将数据、图形和控制分开，是程序的耦合性较小。