军棋对战设计文档

1. 网络通信编程框架

服务器端为处理器，客户端为显示器。

所有游戏数据，包括开始游戏、结束游戏判定、行棋逻辑判定均在服务器端处理，并将结果以状态码+游戏数据形式发送给客户端，客户端将这些信息翻译并显示。

客户端向服务器端发送数据仅发送客户端鼠标动作，由服务器对鼠标动作的有效性及对应游戏动作进行处理。

1. 客户端与服务器端通信协议

基于TCP/IP协议

为避免粘包现象，客户端与服务器端单向通信时保证每次发送数据相同格式相同大小。

客户端向服务器端发送数据格式：状态码（int）+鼠标动作（struct clickdata）

服务器端向客户端发送数据格式：状态码（int）+游戏信息（struct GAME）

若状态码对应状态不含对应鼠标动作或游戏信息（如发送“请求开始游戏”状态），则发送状态码+无效鼠标动作或游戏信息。

接收时对应接收。

1. 服务器端工作流程

创建服务器：创建服务器并弹窗输出监听的服务器IP地址及端口信息（触发方式：菜单栏-创建服务器选项）

连接客户端：生成与客户端连接对应的QTcpSocket，内部标记成为服务器端（触发方式：服务器成功连接客户端）

响应客户端动作：根据客户端数据（格式：状态码+客户端鼠标动作）获知“开始游戏”、“客户端鼠标点击位置”、“客户端认输”等游戏信息，根据这些信息开始、更新或结束游戏对局，并发送更新后的对局数据（格式：状态码+游戏数据）（触发方式：服务器端收到客户端发送的数据）

响应本地动作：根据服务器端游戏动作等导致的游戏数据变化随时向客户端发送对局状态和对局数据（触发方式：计时器变化、超时、服务器端游戏动作、服务器端认输、胜负判定等）

断开连接：向客户端发送断开连接状态码（触发方式：菜单栏-断开连接选项）

1. 客户端工作流程

连接服务器：弹出输入IP地址窗口，检查IP地址有效性，连接服务器（触发方式：菜单栏-连接服务器选项）

响应服务器端信息：根据服务器端数据获知“开始游戏”、“客户端认输”、“超时”、“胜利”、“确定阵营”等信息及游戏数据，根据状态码做出响应，用新游戏数据覆盖旧游戏数据

（触发方式：客户端收到服务器端发送的数据）

发送本地鼠标动作：向服务器端发送本地鼠标动作（触发方式：本地鼠标单击事件）

断开连接：与服务器断开连接（触发方式：菜单栏-断开连接选项）

1. 信号与槽机制设计

功能键信号与槽：

菜单栏-“创建服务器”“连接服务器”“断开连接””开始游戏”“认输”皆使用信号槽连接对应函数

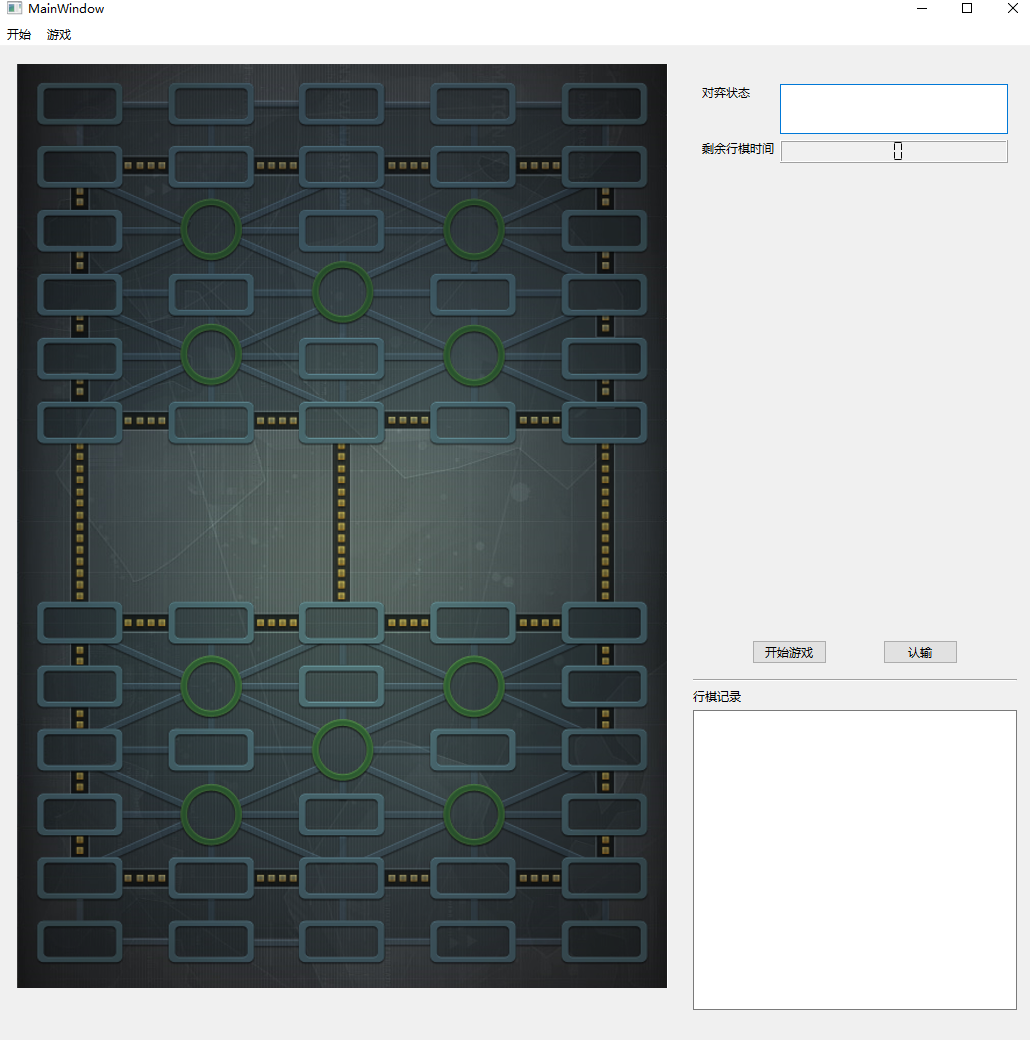
输出信号与槽：

由于游戏核心为MainWindow.ui下的子对象，因此输出时无法直接调用MainWindow的函数，故使用信号与槽机制将游戏状态信息、游戏日志、剩余时间利用MainWindow组件输出；各项提示信息利用MainWindow的弹窗输出。

网络连接信号与槽：

服务器端的监听、服务器端与客户端数据接收时皆使用信号与槽。

1. Gui界面设计



上图为主界面。

菜单栏：

开始：创建服务器、连接服务器、断开连接

游戏：开始游戏、认输

左侧为游戏界面。

右侧上方为对弈状态与剩余时间提示栏，分别标识当前行棋回合数、行棋方及对应阵营和当前回合剩余时间。

右侧下方为开始游戏按钮、认输按钮与行棋记录栏。

行棋记录栏于服务器端生效，记录双方所有行棋历史。

