

# 五子棋文档

## 目录

五子棋文档

概要

通信协议

建立连接

进行游戏

游戏逻辑

用户界面

所有函数和类

1

1

1

2

2

3

3

3

## 概要

运行本五子棋程序时，两玩家分别作为服务端和客户端进行连接以进行游戏。服务端和客户端运行同一份程序，用户可自行选择是作为服务器还是连接到别的服务器。作为服务端的玩家可以收到来自多个客户端的连接请求，该玩家在列表中选择其中一个确认连接。连接建立后，游戏双方点击准备按钮，游戏随机决定黑方白方，并通知玩家。玩家随即开始游戏，某方胜利时，游戏通知双方并结束游戏。玩家可随时点击按钮重新开始游戏。

## 通信协议

本程序使用 TCP 通信，并在其上使用类似 JSON-RPC 的通信协议。每次发送的信息都是一个 JSON 字符串，遵循以下格式：`{ "method": "method-name", "params": { "key": "value" } }`，其中“method”键对应的值是一个字符串，表示命令名称；“params”键对应的值是一个 object，表示参数表。由于 TCP 是基于流的协议，传输过程中可能造成一条信息被截断为两部分，或两条信息被连接在一起，使用上述协议可以清晰地判断出每条信息的位置，同时又可以利用 Qt 对 JSON 的支持方便的解析。为了简洁，下文中将直接以“method(key=value)”的格式，表明传输的信息是“method”命令、带有值为“value”的“key”参数。

所有命令如下简表，下文会详述。

命令	参数	含义
hello	无	确认连接
helloAck	无	回复确认连接

ready	无	表明玩家准备好进行游戏
startGame	bool color	表明游戏开始，并告知对方所执颜色
draw	int row, int column	在某处下一子
restart	无	重新开始游戏

## 建立连接

作为服务端的玩家点击建立服务器按钮后，可输入服务端 IP。玩家输入前，程序会自动检测本机的某个 IP 作为该输入框的缺省值。为了检测本机 IP，程序调用 `QNetworkInterface::allAddress()` 获取本机所有可能 IP，再滤除 IPv6 IP、Loopback (local-host) 和位于 169.254.1.0~169.254.254.255 地址段中的 IP。位于 169.254.1.0~169.254.254.255 地址段中的 IP 可能是 DHCP 获取 IP 失败后分配的不能路由的 Link-local，故须滤除。所剩 IP 中取第一个作为缺省值，用户也可手动输入其他 IP。

用户点击确认后，服务器在上述 IP 端口的 8388 端口监听 TCP 连接，其中 8388 端口号定义于 `const.h`，可以方便修改。程序异步等待有新连接的 signal，每次有新连接时，程序接受该连接，将其作为备选客户端显示在列表中。程序还监听该连接的 `disconnect` 信号，信号发生时将其从列表中移除。用户可以随时点击按钮取消监听，这时所有备选客户端连接将会断开。

用户可双击某备选客户端以接受之，此时服务端向客户端发送“hello()”（含义见上文“通信协议”一节），客户端回复“helloAck()”来让双方均知晓，进入等待游戏开始状态，并将 UI 中显示的状态改为“已连接”。

服务端接受客户端后，对 socket 的处理流程与客户端大致相同。其中，若 socket 发出 `disconnect` 信号，程序从新回到未连接状态。

## 进行游戏

建立连接后，双方玩家点击“准备”按钮准备游戏，点击该按钮时，程序发送“ready()”给对方。当双方准备完毕后，服务端随机决定黑白方，并发送“startGame(color=0)”或“startGame(color=1)”告知客户端游戏开始。此时，服务端与客户端弹出窗口告知玩家所执颜色。此后客户端与服务端的流程完全相同。

在客户端与服务端均会判断游戏逻辑，包括此刻轮到哪位玩家，以及判断输赢。轮到某玩家时，该玩家才能落子，落子后，程序发出“draw(row=x, column=y)”通知对方，其中(x,y)是落子位置。接受本地玩家落子信息的类和接受远程玩家落子信息的类继承自同一基类 `Input`，这样程序可以统一地处理双方玩家的落子。玩家落子后，一方面程序将该落子画在棋盘上，另一方面程序将该位置填入独立的数组中以备逻辑判断，做到了 UI 和数据分离。

某方胜利后，双方的程序均会判断游戏结束，并锁定棋盘，玩家可点击“重新开始”按钮

清空棋盘，再重新点击“准备”按钮分配黑白方并重新开始游戏。“重新开始”按钮点击后，程序发出“restart()”通知对方程序也清空棋盘。

## 游戏逻辑

游戏逻辑由单独的类进行判断，做到数据与 UI 分离。每下一子时，程序扫描该子的八个方向上是否连成大于等于 5 子，若是，则游戏胜利。

使用“危险提示”功能时，为确定某点是否为危险，程序扫描该点的四个方向，判断是否有大于等于 2 个方向上有对方 3 子（将会连成 4 子），或无阻挡对方 2 子（将会连成 3 子），若是，则此点为危险。

## 用户界面

总体上界面分为左侧的棋盘和右侧的按钮组两部分。

棋盘实现自一个 QGridLayout，每个单元重载了 paintEvent 画出十字，并在有棋子时画出棋子，有星位时画出星位。如果格子在边缘，则不画完整十字。所有单元无间隔地拼在一起，形成完整的棋盘。使用 QGridLayout 的好处在于可以方便地判断鼠标点击，而不用手动计算点击位置在哪个交叉点上。

右侧按钮组有“设为服务器”、“连接到服务器”、“取消连接”、“准备游戏”、“重新开始”和“危险提示”四个按钮，按钮在无意义时自动变为不可点击状态。其中，“危险提示”按钮在按下时才提示，松开就关闭提示，避免选手长时间打开提示功能。

## 所有函数和类

查看 [doc/api/html/index.html](http://doc/api/html/index.html) 可查看所有类、函数和变量的文档，该文档由 Doxygen 自动生成。