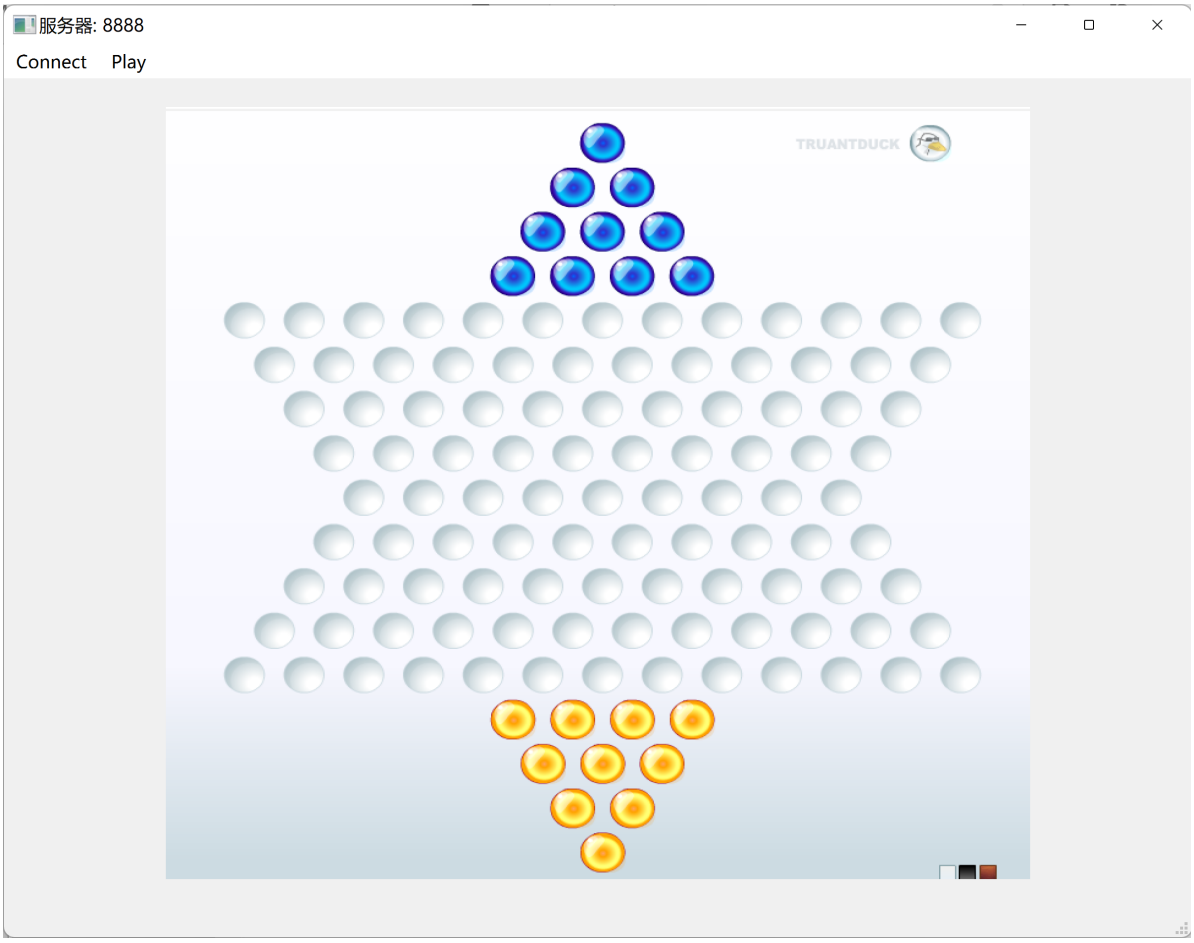
中国跳棋设计文档

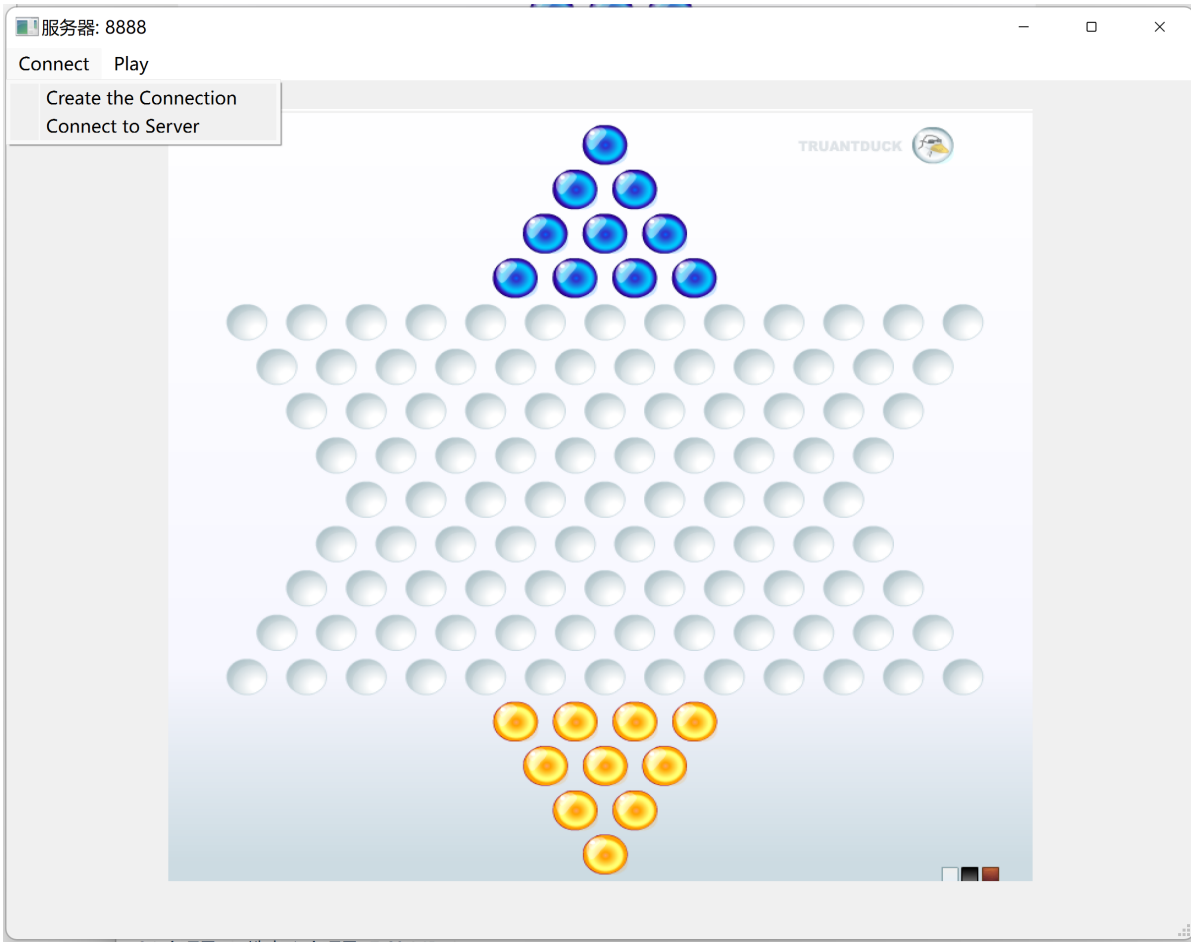
软件功能

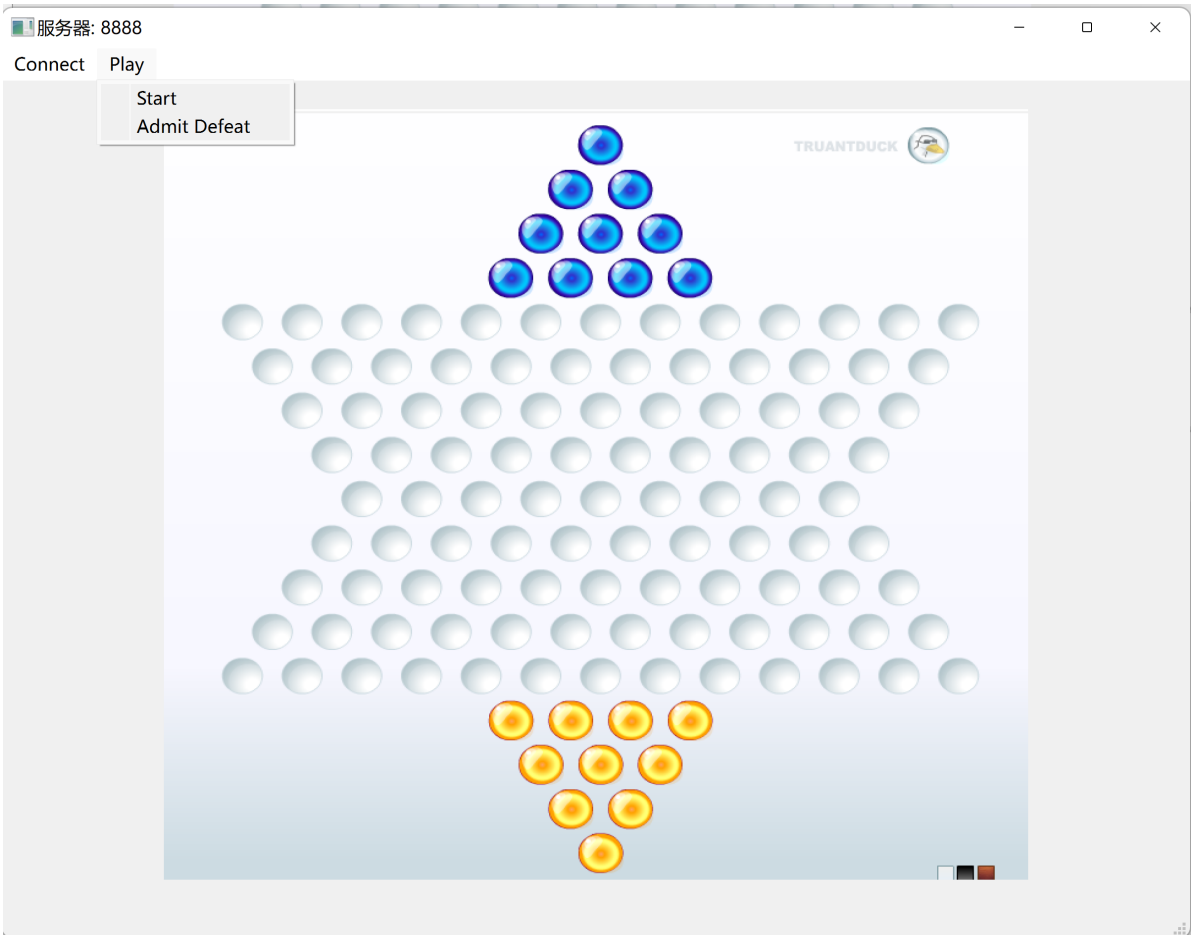
利用 Qt 实现的一款简单的网络对战跳棋小游戏。实现中国跳棋六芒星的布局。实现两个游戏端之间的通信，可以同步棋盘情况。能够自动判断输赢，若是违反规则而输棋，会在弹窗上显示输棋原因，具体提示见下文。具有开始、认输功能。可以打开两个游戏端进行双人对战，也可只打开一个游戏端自己和自己下，但此时没有认输功能，需要点两次开始（start）。

图形界面

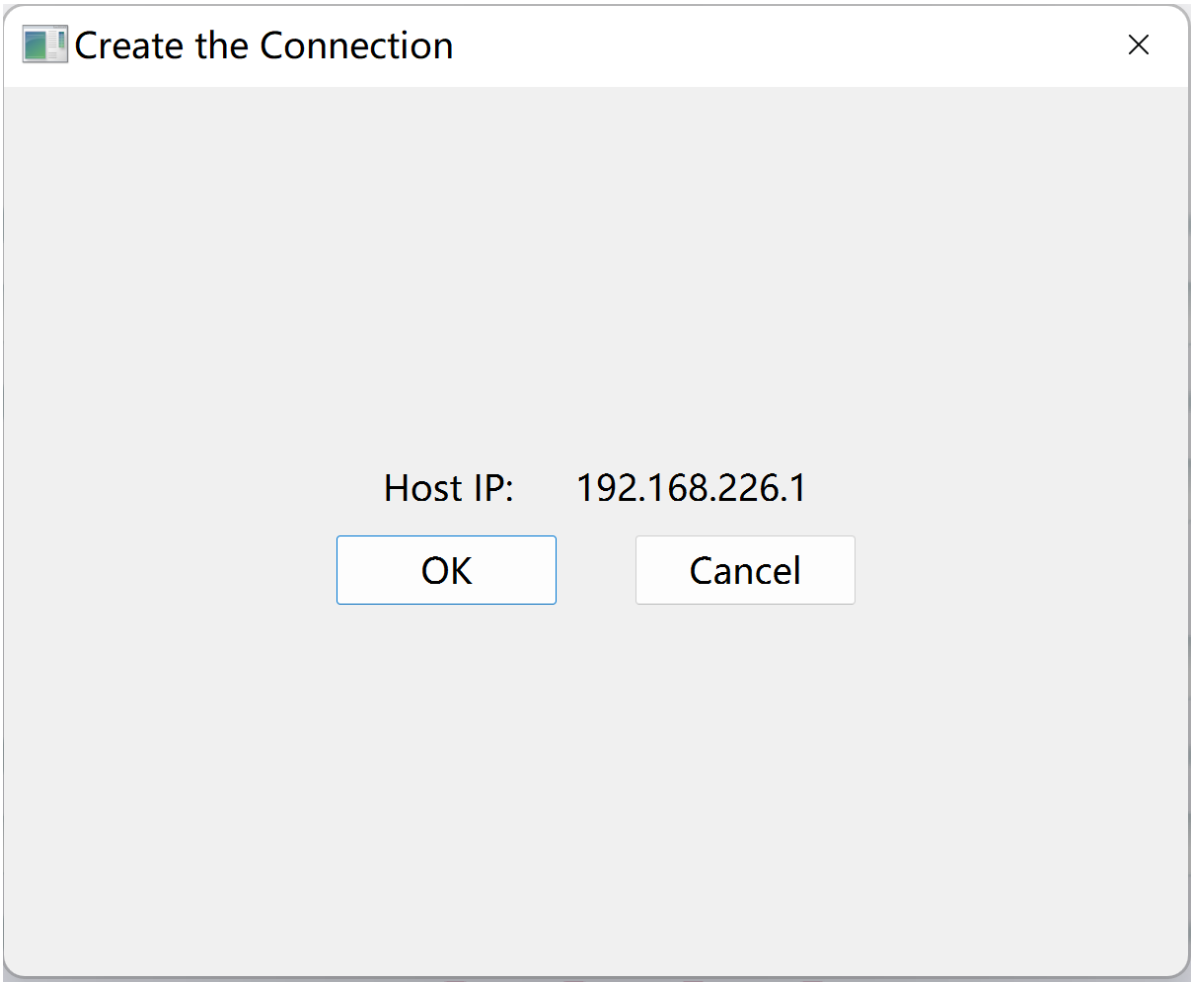


如图为初始界面，可以点击菜单栏的 Connect 和 Play，分别弹出 Create the Connection、Connect to Server选项，以及 Start、Admit Defeat 选项，如下图所示

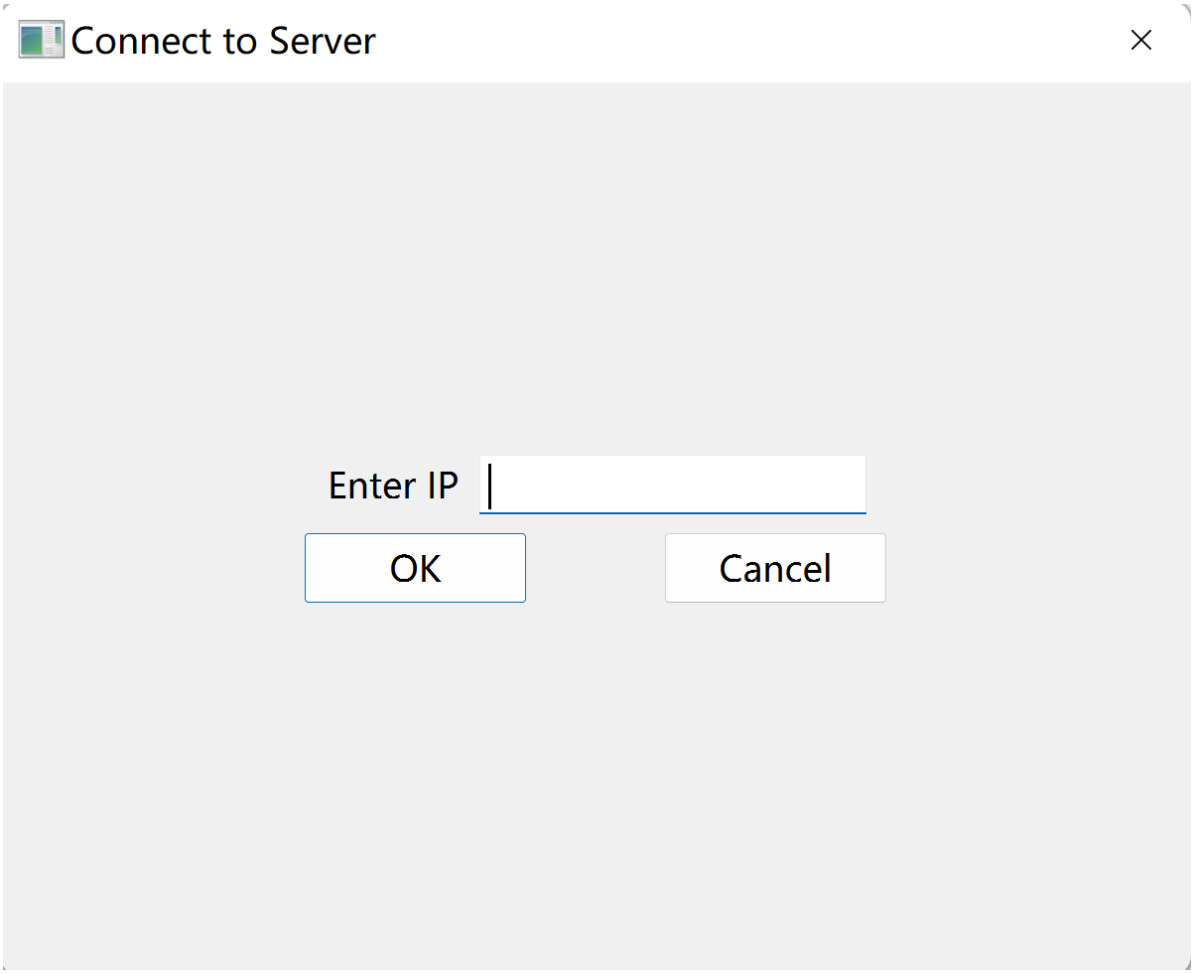


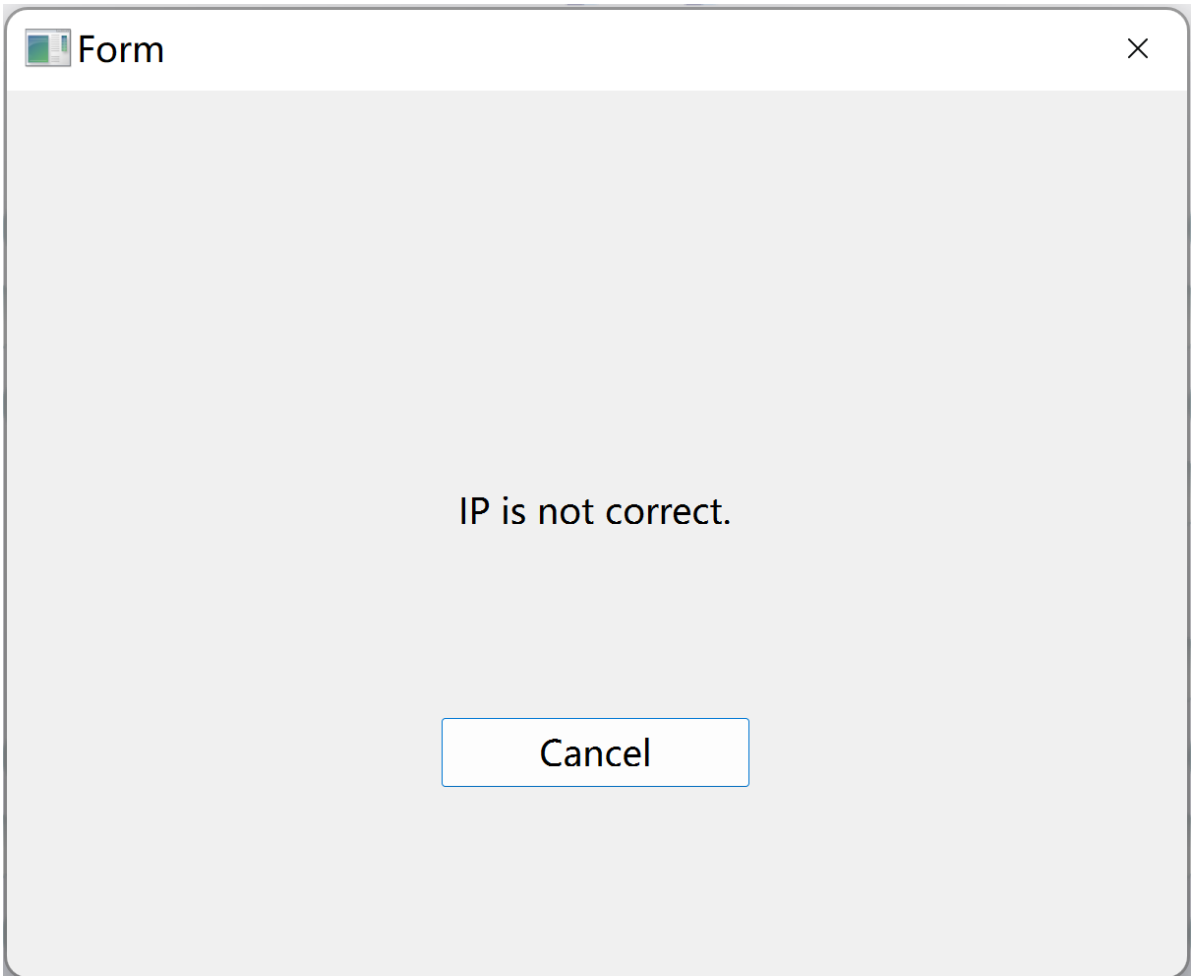


点击 Create the Connection，会弹出服务器（Server）端创建服务器界面，显示 IP 地址。点击 OK，对话框消失，绘制棋盘，等待连接。若点击 Cancel，则不会绘制棋盘，也无法连接。

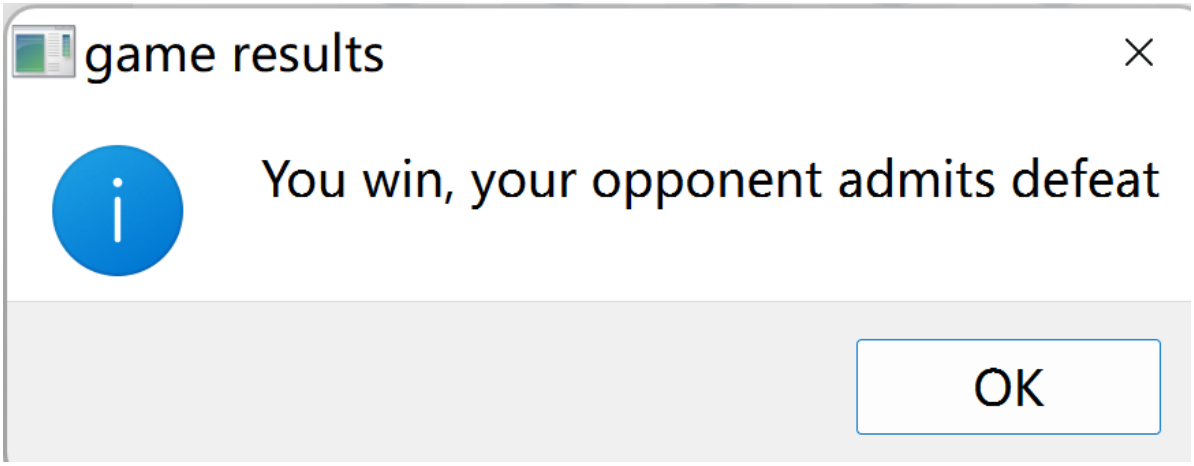


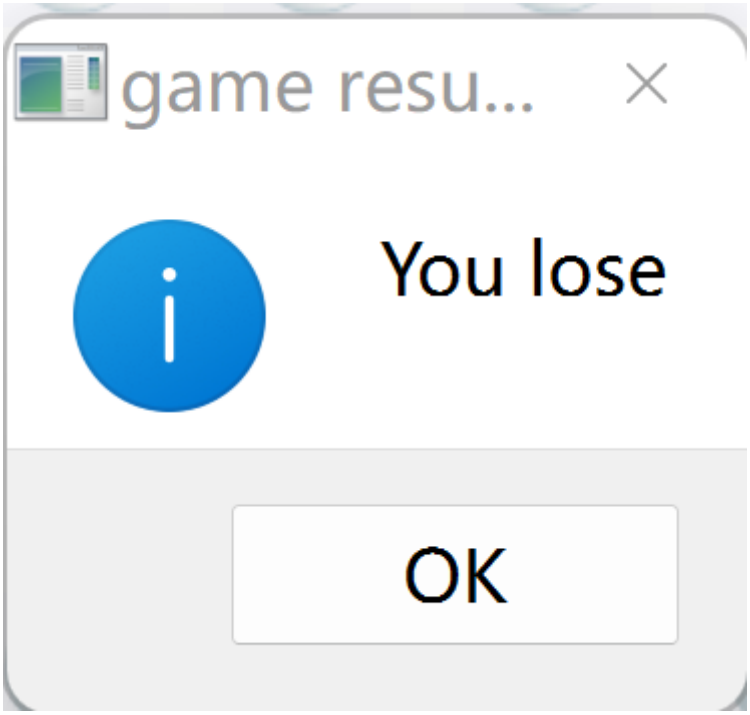
点击 Connect to Server，弹出客户（Client）端连接界面，需要输入服务器端显示的 IP 地址，然后点 OK，若输入的地址格式正确，则会尝试与服务器连接，若成功连接，则会显示棋盘，同时服务器端产生的棋盘会弹出 connected 信息。此后可以在服务器端和客户端的主菜单点击 start，出现游戏开始的弹窗，点击 OK 后，游戏开始，若输入的 IP 格式错误，则会出现错误提示弹窗，需要重新输入 IP。



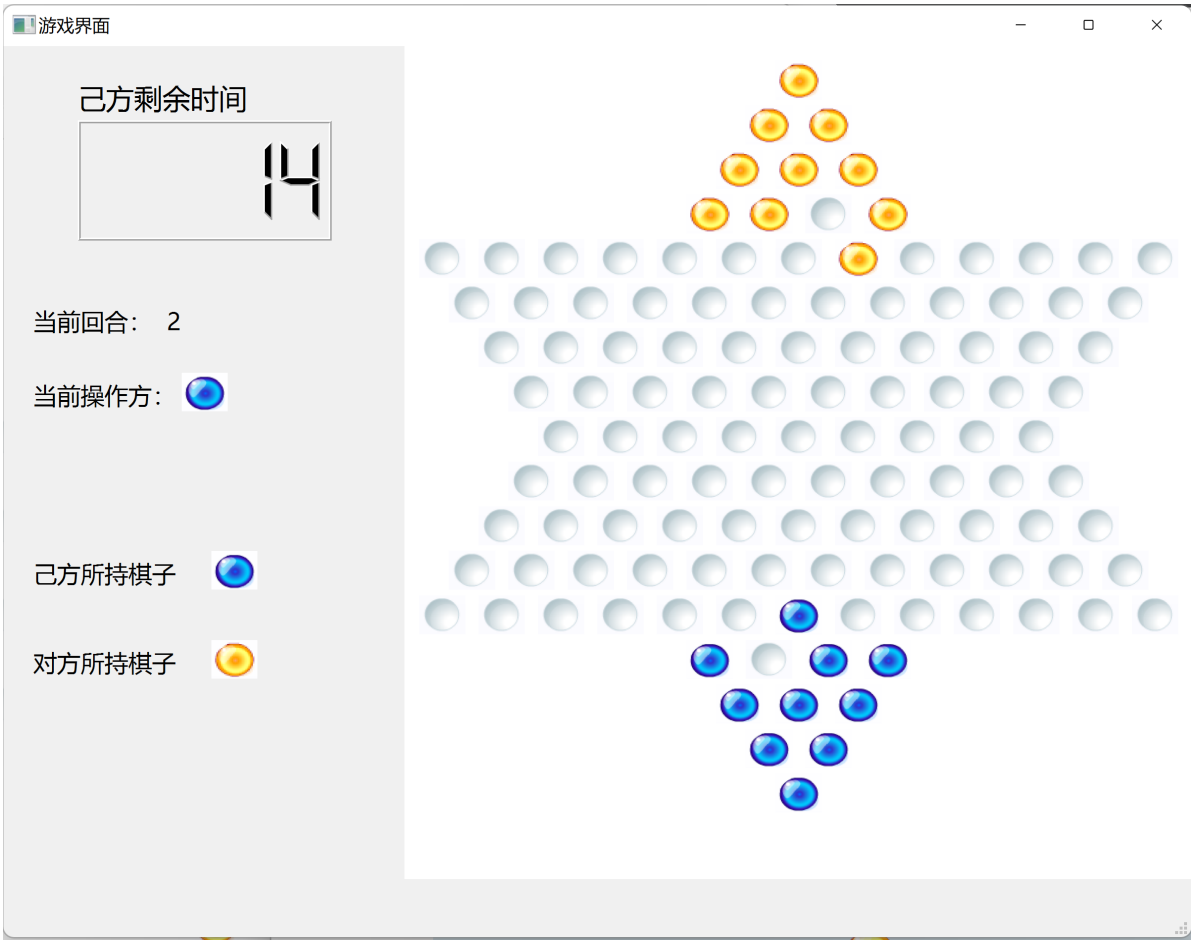


若一方点击认输按钮，则会询问是否真的想要认输，若按下 Yes，则会出现以下胜负弹窗。

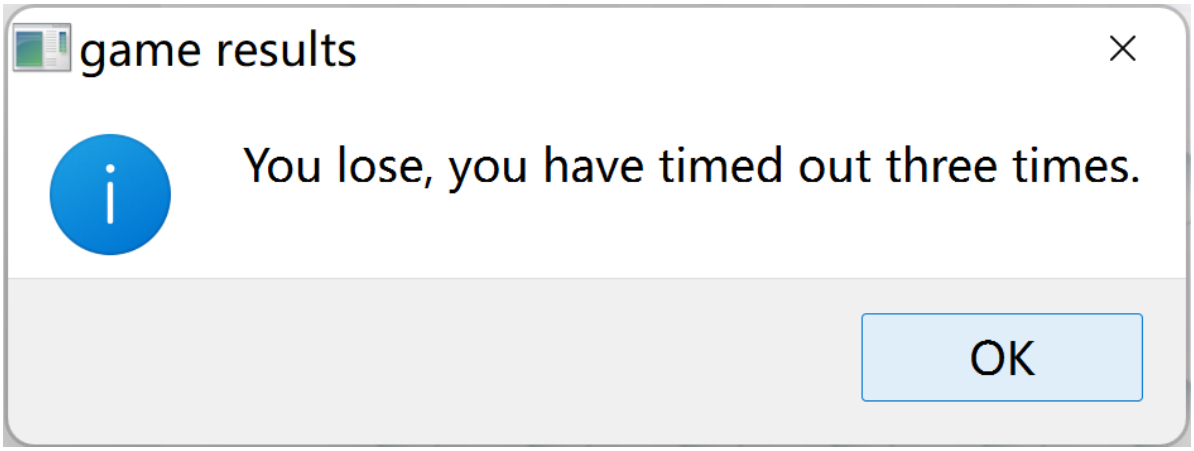


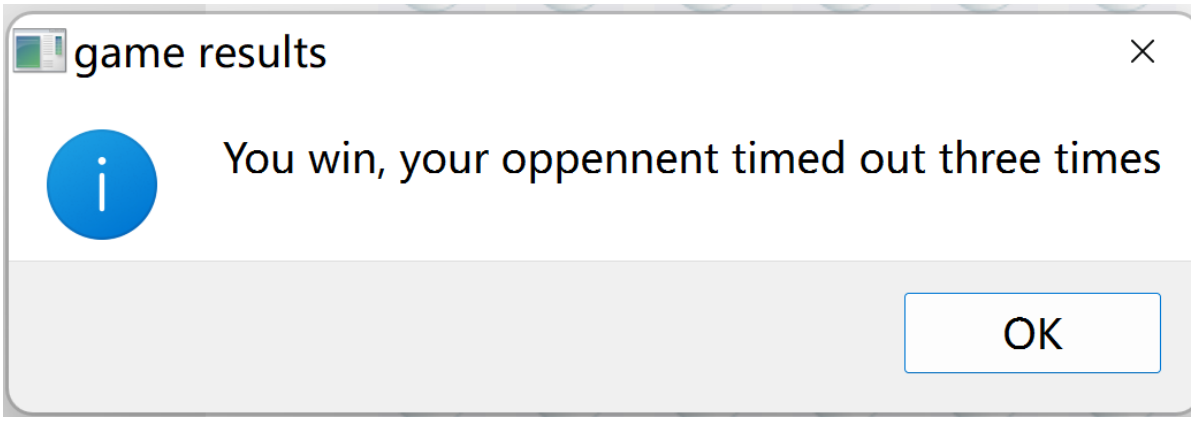


下棋界面会显示己方剩余时间，当前回合，当前操作方，己方所持棋子，对方所持棋子，己方剩余时间采用倒计时，后三者均用棋子颜色表示。

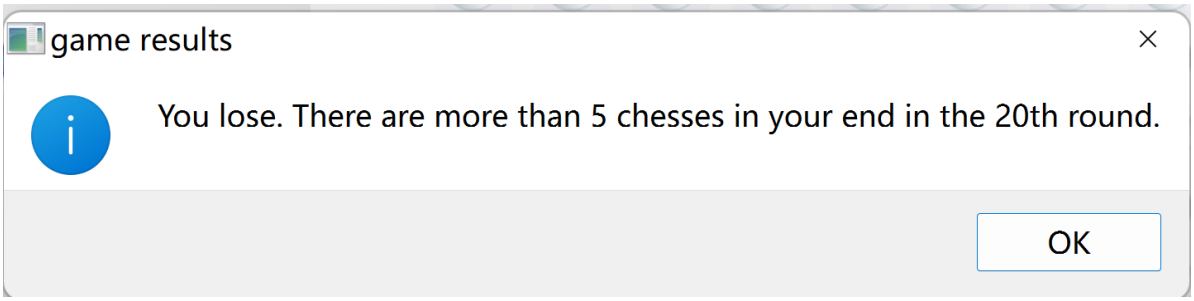


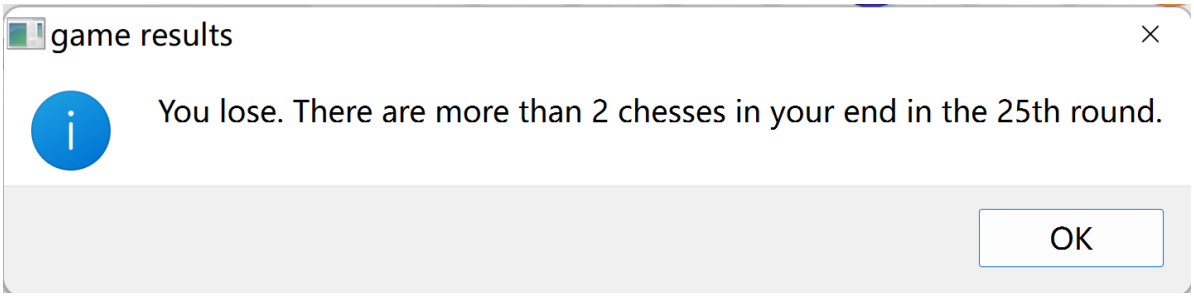
在某一方超时三次后，自动判罚，出现胜负弹窗。

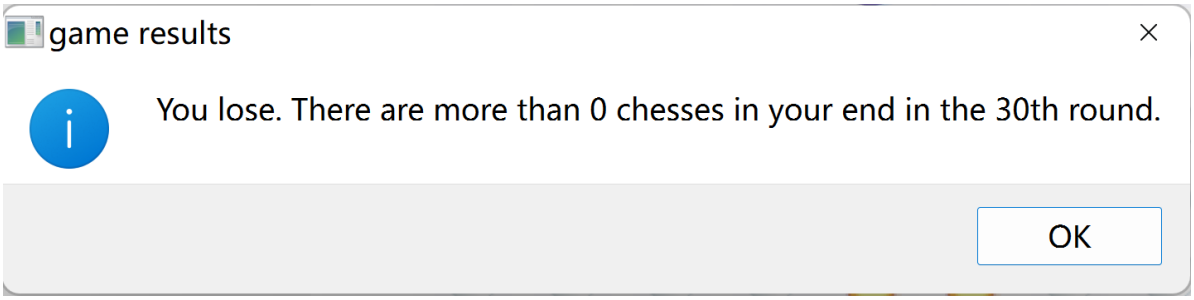


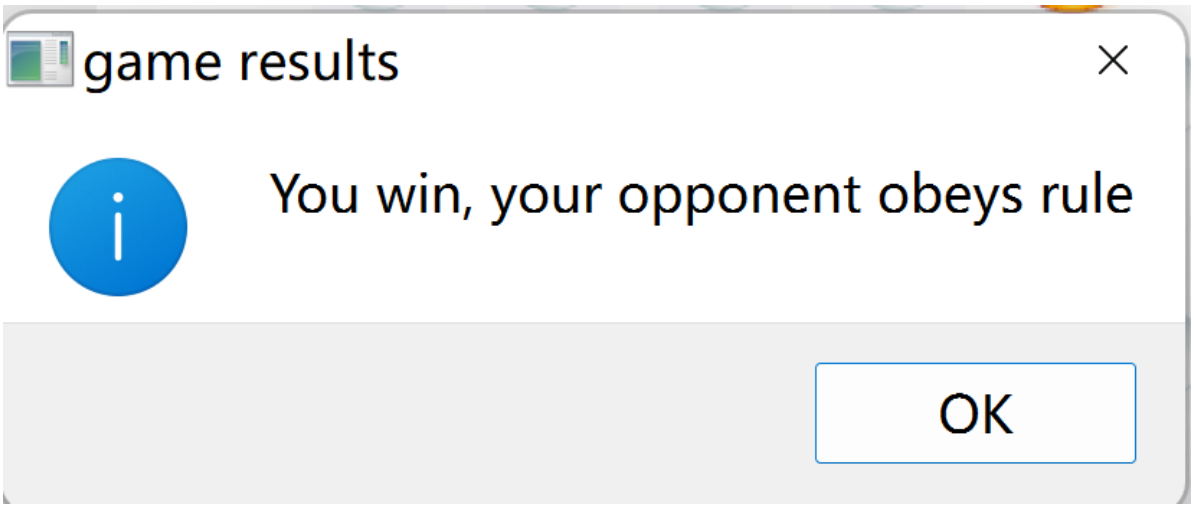


在由于规定步数之内仍有规定数量的棋子没有出大本营，则会出现以下胜负弹窗。

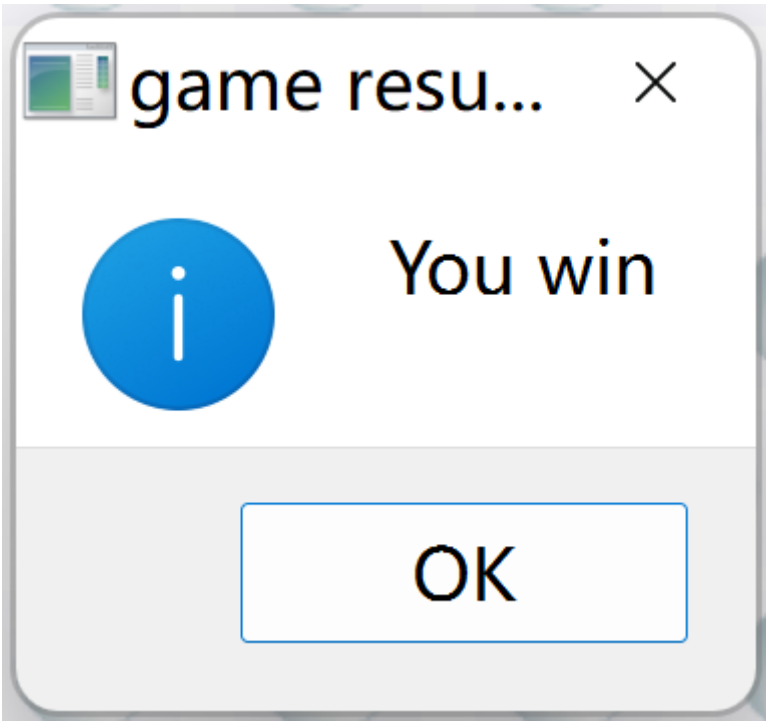


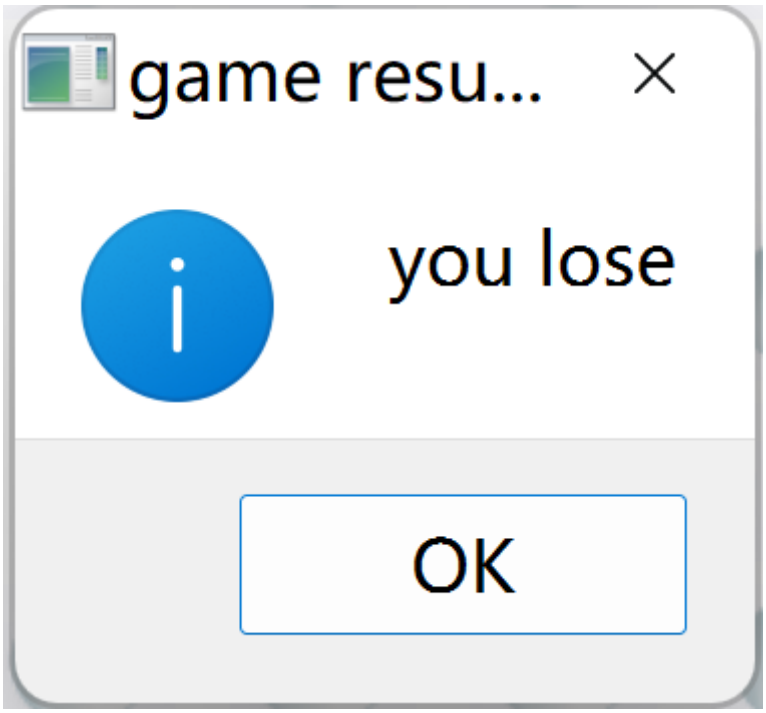






若一方已将全部棋子移动到对方的大本营，则此方获胜，出现胜负弹窗

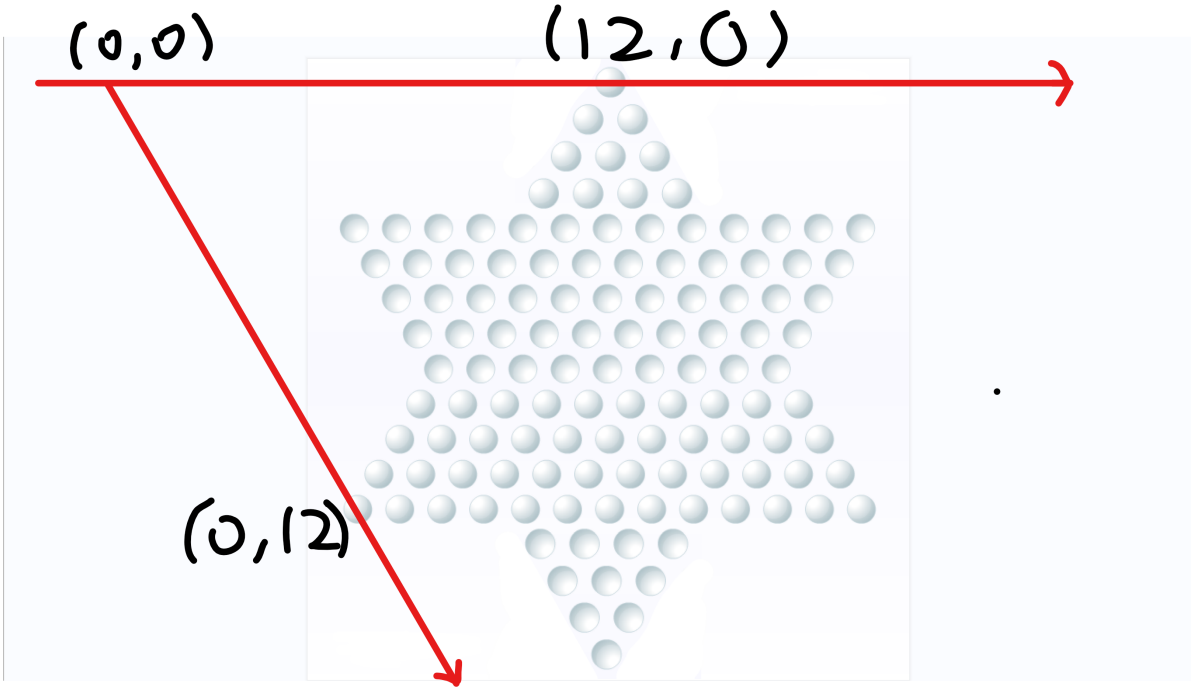




工作流程

棋盘简介：

在设计过程中，写了一个新类 Cell 来表示棋子，Cell 继承自 QPushButton。Cell 中包含棋子的坐标信息以及类型（空格位置为 0，蓝色棋子为1，黄色棋子为2）。坐标为斜着设置的，每一个格子视作一个点，如图所示



棋盘为 121 个棋子组成的，重设了棋子的大小，以及棋子的位置。

socket 通信：

客户端和服务器端使用 TCP协议。在服务器（Server）端创建服务器界面点击 OK 后，服务器端开始监听，端口规定为 8888。在客户（Client）端连接界面输入正确的 IP 后，尝试连接主机 IP 的 8888 端口。

当连接成功后，connected 信号会激发 lambda 函数，依据是否有两个游戏端发送不同的信号给服务器端，服务器端接收到信号后弹出弹窗。若有两个游戏端，则信息形式为`QString str = r`，在此时给客户端设置先手玩家，由服务器端向客户端传输信息形式为"r.先手棋子类型"的信息，否则信息形式为`QString str = c`。

若有两个游戏端，则在服务器端以及客户端的主界面点击开始后，分别向对方传输开始的信息，信息形式为`QString str = "1"`。

在棋盘同步上，会在每一次下棋之后，传输改变棋子的信息，信息形式为`QString`，格式为"空格横坐标.空格纵坐标.棋子横坐标.棋子纵坐标.棋子类型"

当一方超时并且超时次数不超过三次时，向对方传输超时的信息，信息形式为`QString str = "to"`，在接收到信息后，会开始本方的计时，同时设置到了本方轮次。

当一方已经超时三次后，向对方传输超时输棋的信息，信息形式为`QString str = "tw"`，在接收到信息后，出现本方的胜负弹窗。

当一方出现规定步数之内仍有规定数量的棋子没有出大本营时，会向对方传输胜利信息，信息形式为`QString str = "o"`，在接收到信息后，出现本方的胜负弹窗。

当一方认输时，会向另一方发送胜利信息，信息形式为`QString str = "w"`，在接收到信息后，出现本方的胜负弹窗。

当一方胜利时，会向另一方发送失败信息，信息形式为`QString str = "win"`，在接收到信息后，出现本方的胜负弹窗。

信号与槽：

服务器端界面：点击 OK 后发出信号 paint_board，激活主菜单界面的开始监听的函数 begin_server 和绘制棋盘的函数 show_chessboard。

客户端界面：在 IP 格式不对时发出信号 not_correct，激活主菜单界面的展示错误信息的函数 show_incorrect_info；在点击 OK 后发出信号 show_board，激活主菜单界面的 begin_client函数。

服务端的下棋界面：在超时三次时发出信号 time_out_three，激活主菜单界面的展示胜负弹窗函数 server_time_out_lose；在超时未达三次时发出信号 Time_out_signal，激活主菜单界面的 server_time_out函数。

客户端下棋界面：在超时三次时发出信号 time_out_three，激活主菜单界面的展示胜负弹窗函数 client_time_out_lose；在超时未达三次时发出信号 time_out_signal，激活主菜单界面的 client_time_out函数。

主菜单界面：在想要启动服务器端的下棋界面的 QTimer 时发出信号 server_time_play，激活服务器端的下棋界面的 start_timer；在想要停止服务器端的下棋界面的 QTimer 时发出信号 server_time_stop，激活服务器端的下棋界面的 stop_timer；在想要启动服务器端的下棋界面的 QTimer 时发出信号 client_time_play，激活服务器端的下棋界面的 start_timer；在想要停止服务器端的下棋界面的 QTimer 时发出信号 client_time_stop，激活服务器端的下棋界面的 stop_timer

走棋算法设计：

首先在点击本方棋子时计算能走到那些空位，将坐标（a, b）以数字形式（a * 20 + b 由于一共数组维数设置为20*20，所以不会出现重复）存储到 vector 中，再在点击空位时寻找是否可以到达此坐标。若能到达，则移动；若不能到达，则不能移动。

在计算点击的棋子能走到哪些空位时，先判断能走一步到周围六个方向的哪些位置，再判断能跳到哪些位置。在判断能跳到哪些位置时，使用 dfs 算法。