Connect5 - An Ancient Chinese Board Game with network battle implementation, based on Qt

苏克
2014011402, 计45, nagizero@foxmail.com

2015 年 9 月 5 日

目录

I Introduction

本次的大作业要求实现网络五子棋程序，作为网络和多线程编程的练习作业。笔者本想学习QML从而实现更美观的界面，却因事务繁忙失败。之后一定会有相应的更新。

下面具体地说明程序的编写思路和模块化构造。

II System Overview

简单地说，游戏除了游戏主窗口类Game和用以打开不同模式的游戏窗口的类Launcher，大多数都是游戏逻辑和交互的模块化。下面按照具体的情况进行说明：

III Basic Data Structure

3.0.1 Map.h

枚举类Cell给出一个格子上可能的所有状态：白子、黑子、空。

Pos类存储一个二维向量（相当于一个坐标），给出x()和y()两个接口。

CellMatrix类作为“棋盘”的数据结构存在。重要的外部接口函数如下：

- bool move(const Player& p, const Pos& p)

  用于一次下子。返回下子成功或失败的结果。

  实际程序中，Game类的move接口通过调用这个函数来进行双方落子的操作。

- bool checkWin()

  当前棋局是否已经分出胜负？如果有人胜利，返回true，否则返回false。

五子棋的规则决定了胜利方胜利时，胜利方一定唯一，而且最后的落子一定直接导致胜利——亦即最后的落子一定属于某个“五长连块”。因此只要记录并更新棋盘上最后一处落子点的位置，在它的一周八个方向进行探索计数，得到四个方向（水平、竖直、主副对角线方向）上同色联通长列的长度，判断其是否为5以上即可。

当然，如果想要节省空间，也可以做这样的设计：每次move()内成功落子之后，判断是否胜利。这样便不需要记录最后落子点。

但是本程序中因为有“悔棋”的需求，记录落子顺序（或最后落子点压栈）成为了必须。便没有进行上述设计。

重要的外部槽如下：

-

IV System Architecture

V Human Interface Design