骑士周游问题

# 1 算法设计说明

骑士周游问题采用递归加回溯的方法来遍历整个棋盘，若每次只判断当前位置的下一个可行点，则寻找较慢，需要引入剪枝，即求当前位置的下一个可行点的可行方向数，选择最小的可行方向数的可行点作为下一个位置，再进行递归与回溯。其算法流程图如下：



# 2 程序设计说明

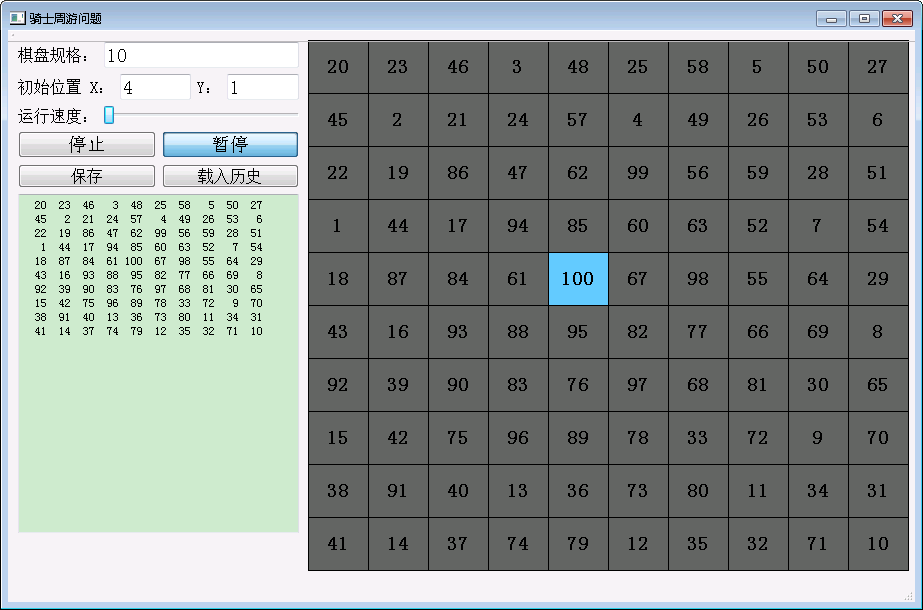
本程序开发环境为Win7 64bits电脑的Qt5.9软件下实现的，工程需要放置于无中文字符的路径下才可正确编译、运行。

工程中窗口界面程序为mainwindow.cpp/h，负责显示及调用信号槽函数完成参数设置如棋盘规格大小及初始点位置，并可以设置开始/结束/暂停/继续等功能。

工程中主要算法程序为qknighttour.cpp/h，继承了QThread以继承多线程功能，并实现骑士周游的寻路算法，如第1章所述。

具体代码细节可参照代码，已给出详细备注。

运行结果：



针对具体要求：

1. 完整的图形用户界面，可设置棋盘初始位置、动画速度等参数；

在主窗口里获取Line Edit的值以设置初始位置，获取Horizontal Slider的值以设置线程睡眠时间以改变动画速度。

1. 算法支持多线程；

QKnightTour类继承QThread后重新run函数以支持多线程。

1. 执行过程可以暂停/继续；

通过设置QMutex互斥量可在主线程里控制子线程运行暂停/继续功能。

1. 算法执行的结果（包括中间结果）可以持久化（以自定义数据格式保存到文件中）；

通过设置写文件函数QFile、QTextStream可将结果保存到文件里。

1. 可从文件中读取数据，并在此基础上继续执行；

通过设置读文件函数QFile、QTextStream可按照存储顺序将文件数据提取到工程变量里。

1. 实时图形化显示执行结果；

在递归函数里发射信号以触发窗口函数刷新界面，可实现动画效果。

1. 可在执行完成后，或暂停时，打印执行的当前结果；

通过设置写文件函数QFile、QTextStream可将结果保存到文件里。