

# DMFB 模拟系统

2018011365 计 84 张鹤潇

## 一、 概述

本程序用于模拟 DMFB 控制界面，通过 Graphics View 架构进行图形绘制；在完成了七项基本功能的基础上，利用 QGraphicsview 类原有的接口增加了一些新的功能，提高了程序的实用性。

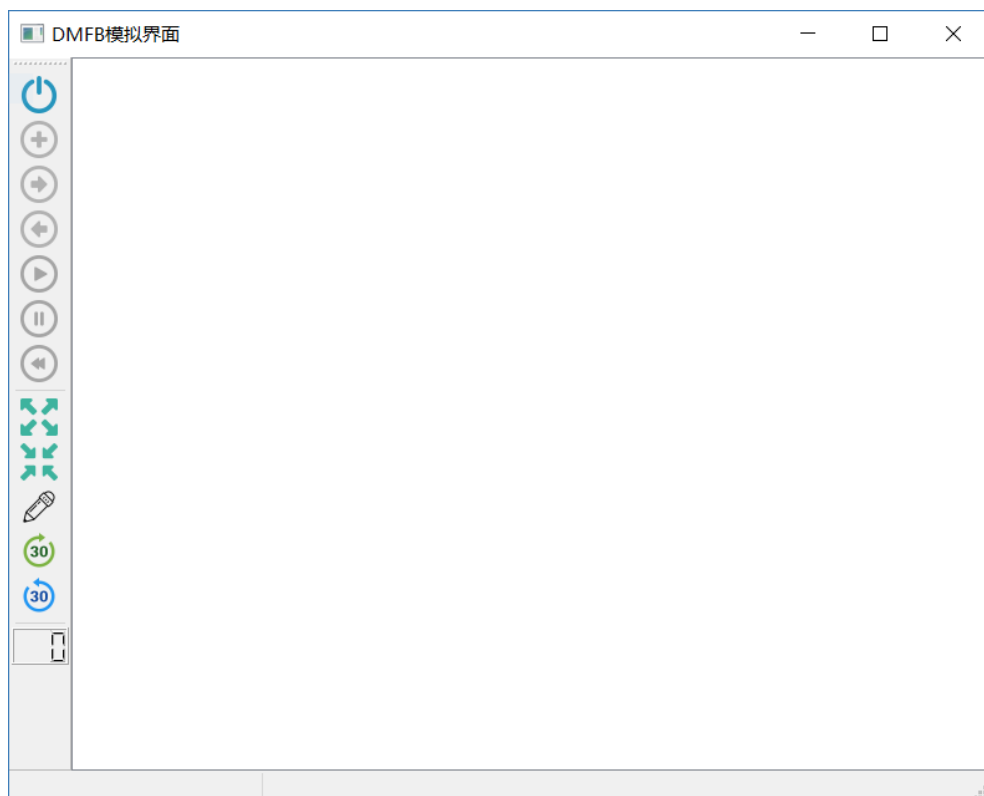
## 二、 开发环境

Windows 10 家庭中文版 1803

Qt 5.9.8 mingw 32-bit/Qt Creator 4.8.2

## 三、 功能展示

### 1. 界面概览

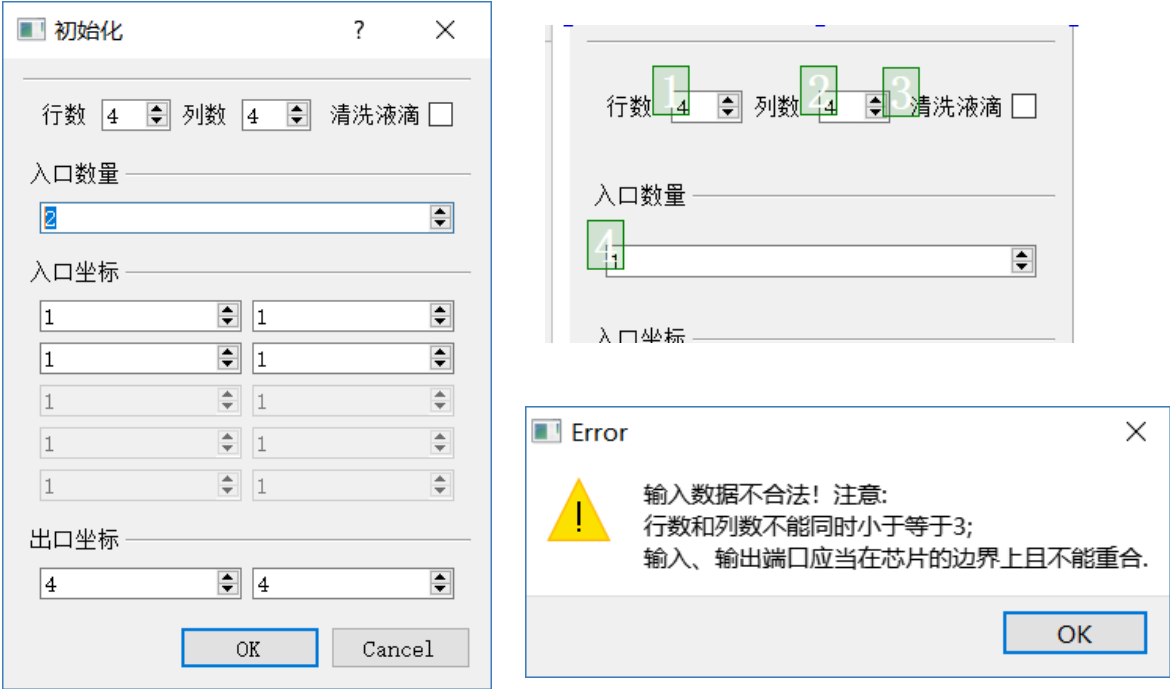


本程序的初始界面如图所示。左侧为工具栏(可拖动)，下方为状态栏，界面主体为一个 QGraphicsview 控件。

工具栏从上到下对应的功能分别为：初始化(Key: Ctrl+N)，打开文件(Key: Ctrl+O)，下一步(Key: Right)，上一步(Key: Left)，播放(Key: 空格)，暂停(Key: 空格)，复位，放

大，缩小，添加文字，顺时针旋转，逆时针旋转。根据当前状态的不同，程序将自动更新各个 Action 按钮的可操作性。工具栏最下方的 LCD 板显示了当前状态对应的时间。

2. 初始化



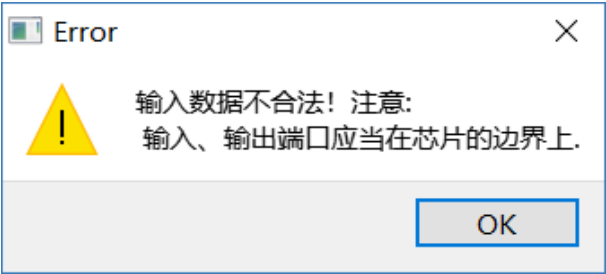
初始化界面和按 Tab 键的访问顺序如图。

按下“初始化”按钮，打开初始化对话框。随着入口数量的变化，入口坐标输入端口的数量也会发生变化；改变行数和列数将改变可输入坐标值的范围。默认的输入输出端口分布于 DMFB 的左上角和右下角。

如果用户的输入不满足条件，将弹窗报错。

3. 打开文件

按下“打开”按钮，打开一个 txt 文件。如果该文件与当前 DMFB 不匹配，将弹窗报错。

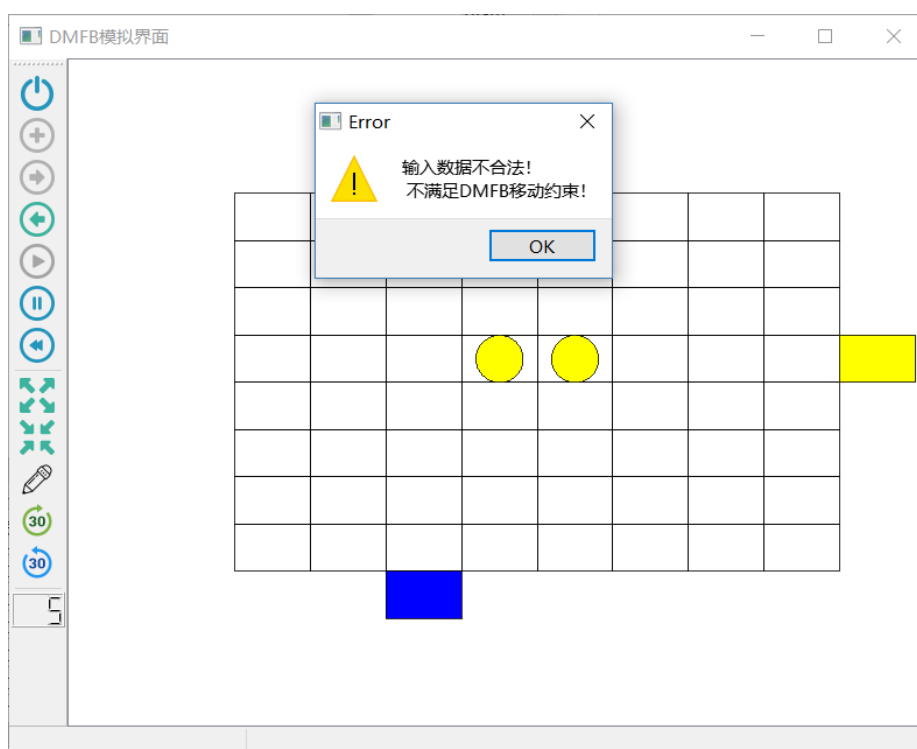


4. 播放与约束检查

通过“下一步”，“上一步”，“播放”，“暂停”，“复位”按钮观察每一个时刻对应的状态。

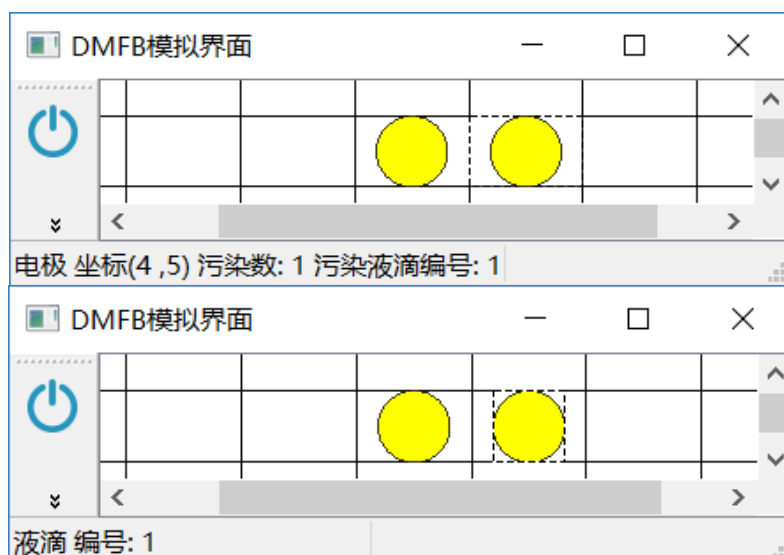
使用 QSound 类播放当前状态对应的音效。

如果液滴的移动规则不满足约束条件，则停止执行并弹窗报错。



## 5. 污染情况绘制

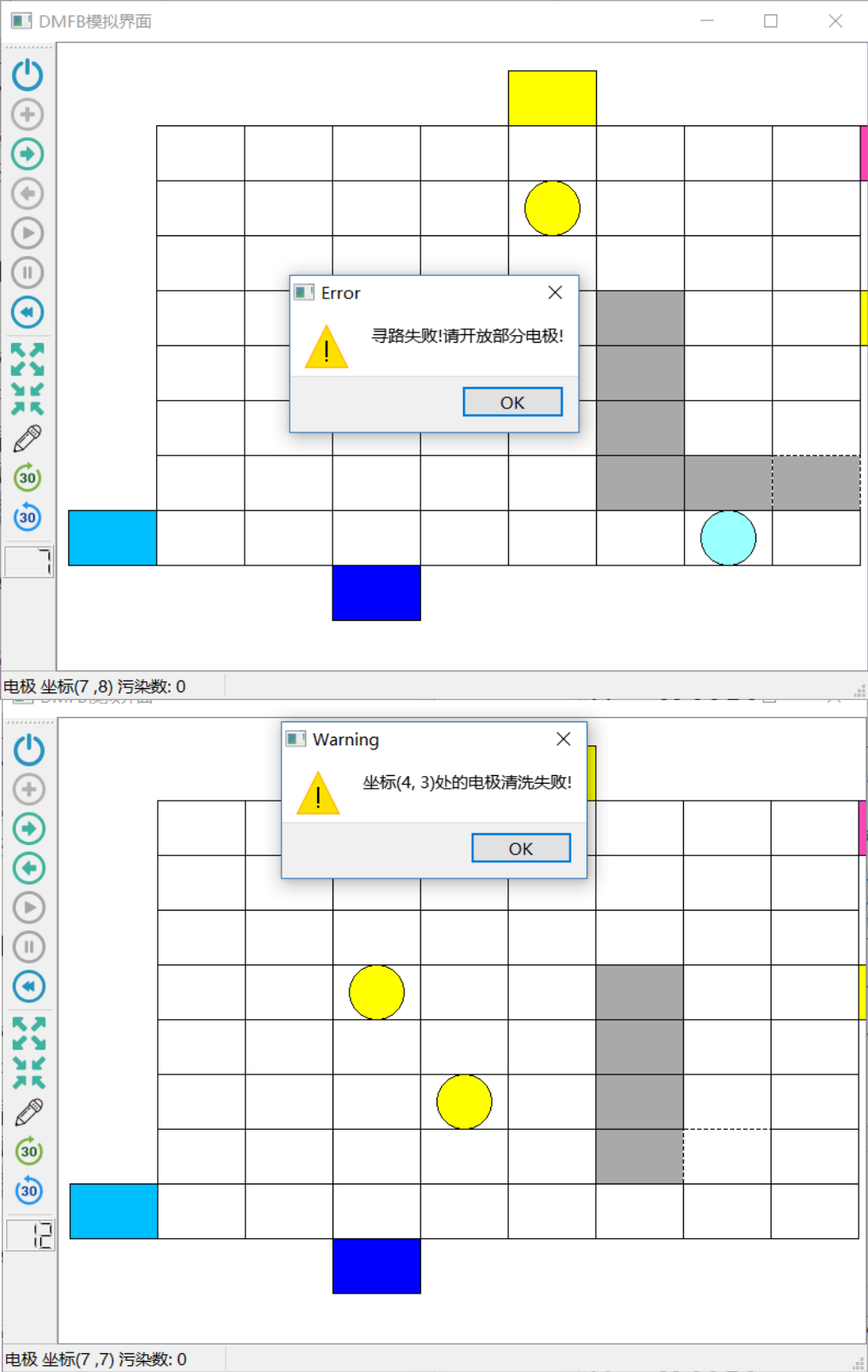
左键双击电极方格，可以在状态栏获取该电极在该时刻的污染状况；左键双击液滴，可以在状态栏获取液滴编号。既方便了用户获取信息，又不致使图形过于混乱。



## 6. 清洗模式

在初始化窗口中，勾选“清洗液滴”将开启清洗模式。清洗入口和出口固定于 DMFB 的左下角和右上角。在清洗模式下，“播放”，“暂停”按钮将不能使用；双击电极将导致电极变为灰色，表示清洗液滴无法经过该电极。再次右键点击，则取消该电极的障碍，恢复原来的颜色。

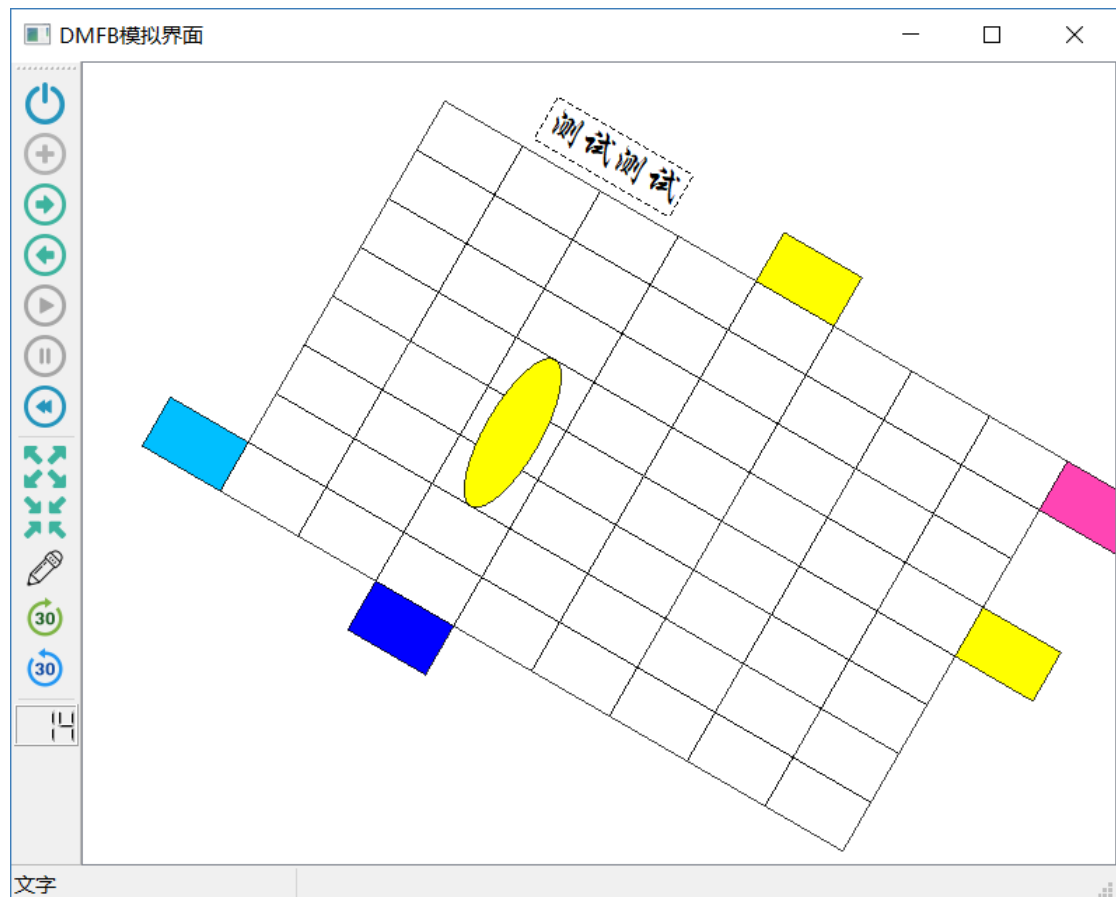
如果程序发现某一位置将发生污染，且清洗液滴可以清除污染，将进入清洗状态。此时LCD板停止增加，其它液滴不动；不断点击“下一步”，清洗液滴会向目标移动。**在清洗过程中增减不可经过的电极时，清洗液滴的路径将自动改变。**如果无法清洗，将弹窗报错。  
被清洗的电极污染数会归零。



## 7. 附加功能

按下“放大”，“缩小”键，整个图像将放大或缩小；按下“顺时针旋转”，“逆时针旋转”键，图像将顺时针或逆时针旋转  $30^\circ$ 。

按下“文字”键，可添加文字。双击文字，可对文字的字体，颜色等属性进行编辑。文字可拖动。在选中文字时按下键盘上的 Delete 或 Backspace 键，可将文字删除。



## 四、 架构与算法简述

InputDialog 类继承 QDialog，用于在初始化时弹出窗体。

QWGraphicsView 类继承 QGraphicsView，用于在画图时发射键盘和鼠标事件信号。

Solver 类用于清洗液滴的寻路。

MainWindow 类继承 QMainWindow，用于控制主窗体和协调调用其他模块。

### 1. 绘图

绘图模块使用了 Qt 的 **Graphics View 架构**。由于 QGraphicsView 类没有预设的键盘和鼠标事件信号，所以继承了一个新的 QWGraphicsView 类，补充了鼠标单击，双击和键盘的信号。

利用了 Graphics Item 的 data 属性保存了每个电极的位置和污染情况，以及每个液滴

的编号。当鼠标选中某一个 item 时，程序会将对应的信息显示在状态栏中。

利用 Graphics Scene 的自有接口，实现了添加文字，放缩，旋转等附加功能。

## 2. 预处理

读入文件时，在 PreProcess 函数中进行预处理，将每一秒的存在于界面上的液滴预先存放在 QMap liquid 中。播放时，移除上一状态的液滴，渲染当前状态的液滴。

在预处理中，还进行了**约束检测**；如果开启了清洗模式，还将**检测并记录在哪些时刻哪些电极可能会发生污染**，以便在对应的时间段发出清洗液滴。

## 3. 清洗寻路

寻路算法为 BFS。QMultiMap<left(int), QMap<right(int), position(QPair)>> toWash 记录了在 [left, right] 时间段内需要清洗的电极坐标。当进入 [left, right] 时间段后，程序将检测清洗液滴能否成功清洗目标电极；如果能，就发出清洗液滴，进入清洗状态。

在清洗状态中，每次点击“下一步”按钮时，程序将检测**不可访问的电极是否发生变化**，**如果变化，就重新寻路**。