

李墨珩 2017050025

程序设计第一周大作业说明文档

September 2, 2018

程序设计第一周大作业

微流控生物芯片流体模拟计算界面

一：主界面 MainWindow

采用三段式设计：控制区、芯片模拟区和诊断区。

控制区内所有按钮均通过信号槽与 MainWindow 中槽函数相连，用来唤出对话框、调用计算函数、控制视图缩放。另外，控制视图缩放的按钮采用 SignalMapper 以优化代码结构。

芯片模拟区添加了一个自定义的 PaintWidget，继承自 QWidget。并在其中添加了 GraphicsView 部件。采用 GraphicsView 架构来绘制芯片。选择 GraphicsView 架构的原因和操作 GraphicsView 架构的方式会在文档的第三部分详细说明。PaintWidget 是 MainWindow 与 GraphicsView 中间的桥梁，以减少 MainWindow 代码的冗余，MainWindow 收到芯片参数的设定，告知 PaintWidget，PaintWidget 在进行 GraphicsView 视图的绘制与调整。

诊断区主要负责显示当前的芯片参数、计算结果，如果程序出错，还可以显示帮助信息。

二：对话框

程序中涉及到4个较为重要的对话框（输入所有参数、输入行列数、输入出入口位置、输入管道宽度），其重要作用均为获取用户的输入，所以4个对话框的主要实现流程大致相同。父窗体某个部件被触发发出信号 - 父窗体槽函数被调用 - 创建对话框 - 准备接收对话框发出的带参信号 - 对话框弹出 - 用户输入 - 对话框某个部件被触发 - 对话框槽函数被调用 - 对话框槽函数获取用户的输入并检查用户输入合法性（若不合法

则弹出警告不做任何动作) - 若合法则发出对话框带参信号 - 父窗体收到带参信号 - 父窗体带参槽函数被调用

为进一步规范用户输入，需要输入数字的对话框的输入部件均采用了 spinBox 并限制了最大值、最小值等参数。

三：GraphicsView 架构

选择 GraphicsView 架构的原因：程序运行时，需要根据用户的动作对芯片的某一根管道进行增添、删减和自定义宽度等操作。GraphicsView 架构可以把绘制每一根管道看成单独的对象，并赋予可以更改的属性，可以响应鼠标事件，进行碰撞检测，进行动态的管理。

具体操作：由于管道的高度定制化，我新建了一个继承自 QObject 和 QGraphicsItem 的子类 TubeItem，继承自 QObject 是为了使用信号槽机制，继承自 QGraphicsItem 是为了使用 GraphicsView 架构。

每一个 TubeItem 对象有 x y w h i j id State c 九个属性，分别代表 x坐标 y坐标 宽度 高度 处于第i行 第j列 管道编号 是否已启用 管道宽度变化带来的阻值。在 TubeItem 里，重新定义了管道边界，绘制方式，如何响应鼠标事件。

关于响应鼠标事件：当鼠标单击时，更改管道启用属性 State，根据属性决定管道颜色，重新绘制。当鼠标右击时，若管道为启用状态，唤出设置管道宽度对话框，收到信号后，修改位置长宽等属性，重新绘制。每次鼠标点击，都向 MainWindow 发送包含管道id和阻值的信号，MainWindow 收到信号后及时更新管道状态数组，以便后续进行计算。

四、整合内核算法

将内核算法除 main函数 外全部放入 cal.h 头文件中，main函数 中的操作放入 MainWindow 中。

五、具体代码实现

已在代码中作出很详细的注释，请移步代码。