fakeCOMSOL v1.0 - Microfluidic Chip Simulation UI

Introduction

- A MCS(Microfluidic Chip Simulation) User Interface, core by TA.
- Developed by LyricZ.
- A summer programming training project, 2018 Summer.

Features

- ✅ OOP Design Style
- ✅ Efficient
- ✅ Cross-Platform
- ✅ Concentration Computing
- ✅ Colorful Display
- ✅ Simulated Annealing Auto Design
- ✅ Pipe Width Changing
- ✅ Multithread Timer

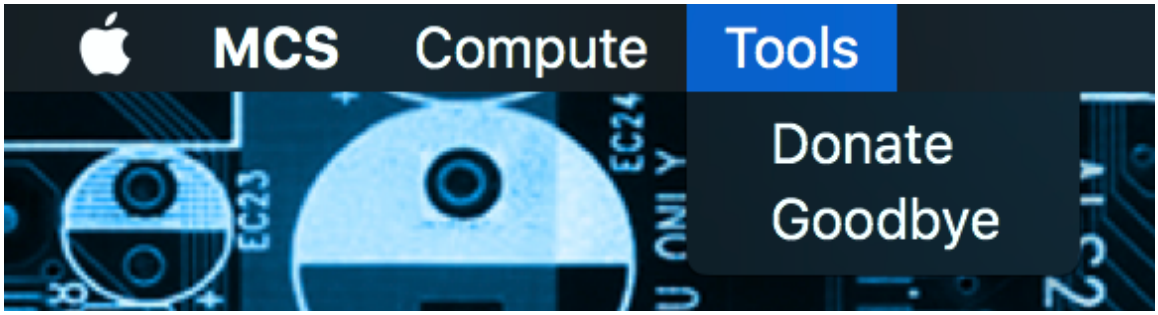
Environments

- Local Compiling Environement
  - macOS 10.13.4
  - QT 5.10.1
  - Clang 902.0.39.1
- When compiling in Windows, there maybe be some errors about isnan (different compiler version causes?).

MainWindow

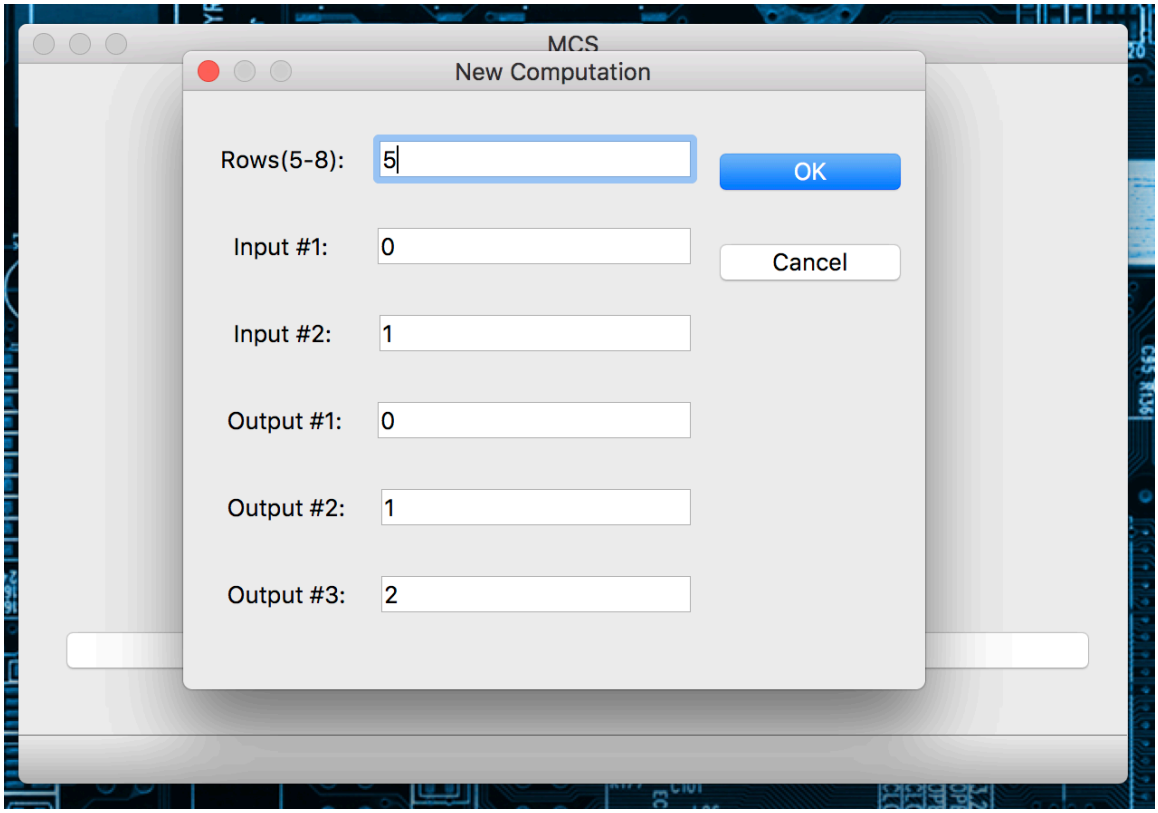
- A simple welcome interface with QPixmap (the fonts are created by PhotoShop and inserted in).

- Using simple QMenuBar.



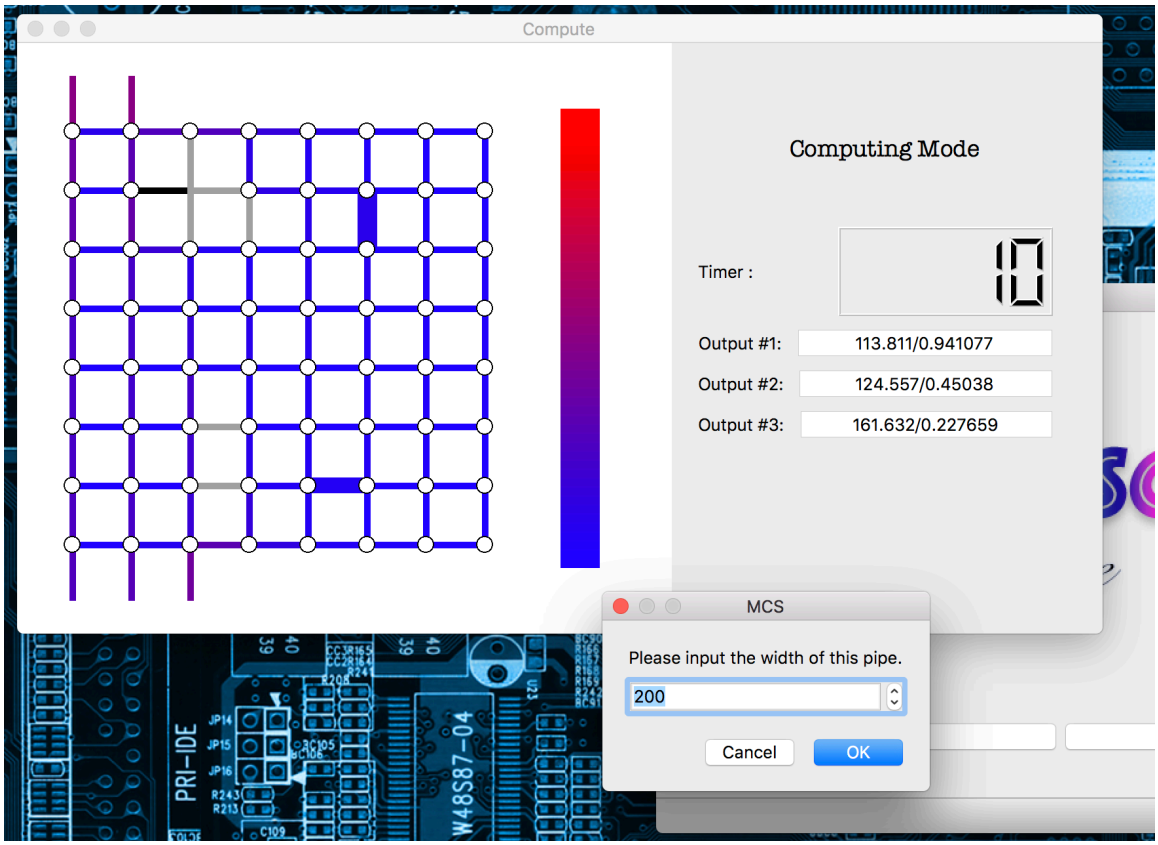
Computing Mode

- Click "Compute - New Computation" to create a new computation.



- If the input has something wrong, a warning dialog will be showed.

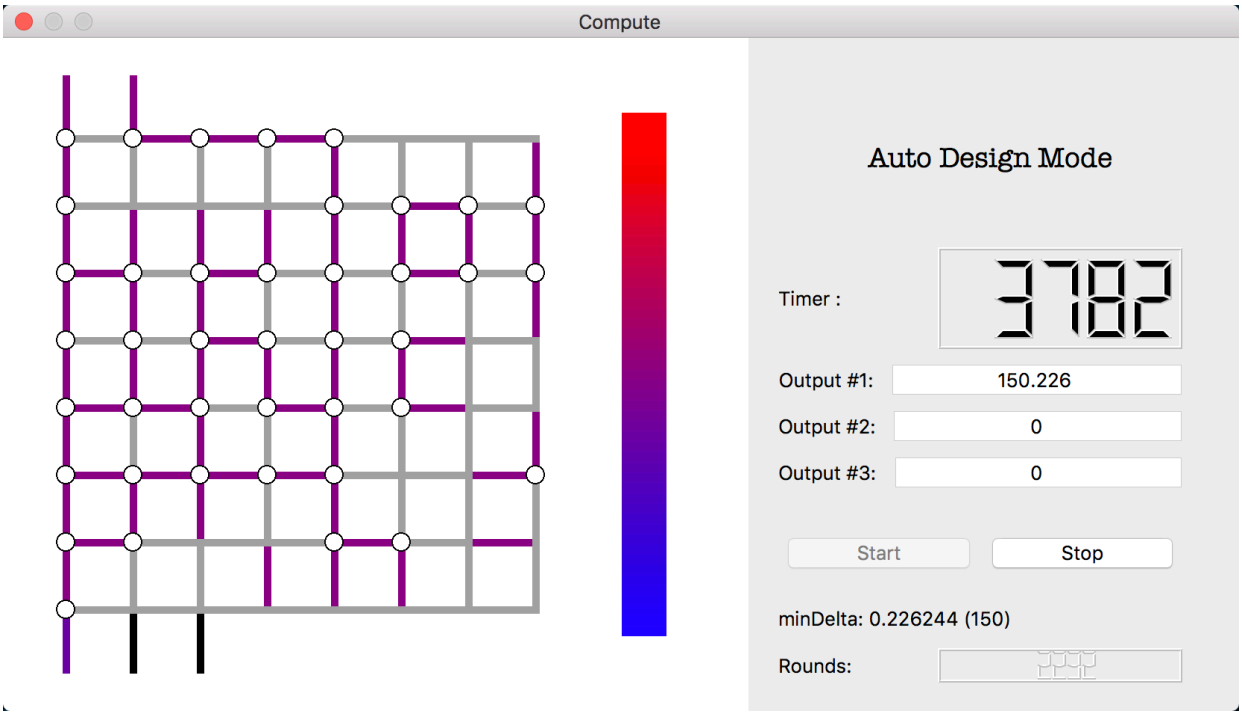
- Computing Interface



- The pipeline is draw by QPainter.
- The color of the pipeline is decided by its flow speed.
- The output includes both speed and concentration.
- The timer is implemented by inheriting the QThread, so that it will not have an impact on the UI fluency.
- Just click on the pipe to change its existence and press Space to

change its width, which will be strictly limited in a range.

Auto Design Mode



- Here I implement a simple Chip Auto Design using Simulated Annealling Algorithm.
- Click the Start Button to start iteration, when reaching the convergence point, the process will be stoped by itself.
- The SA is implemented using QThread.

About the code

- The code is designed in OOP Style, there are many codes reusing, such as the computing mode and auto design mode are using the same class, the only difference is the parameter.
- The class diagram is as the following one.

- about.cpp/h
  - AboutDialog is inheriting QDialog.
  - This file implements the dialog of the About window.
  - (Just for fun...)
- ComputeMain.cpp/h
  - This file implements the main computing window (including Auto Design Mode).
  - Class ComputeMain is inheriting QDialog and has the status of all the pipes, a simulated annealing thread, a timer thread.
  - When it comes into the computing part, it will create a computing instance and deliver the data into the computer, all the computers(including the speed and concentration) are implemented in independent classes, the design uses many OOP knowledges.
  - Of course, this part has its own thread, but it is only used to display, all the computing work is finished by other threads.
- main.cpp
  - Entrance of the program.
- mainwindow.cpp/h
  - A simple welcome interface.
- newcomputewin.h/cpp
  - Here I make class NewComputeWin to get the basic information of the user's requirements.
  - It will also check whether the input is legitimate.
- sa.cpp/h
  - The implement of Simulated Annealing
  - The class SA will emit signal AutoStop() to tell the main window

that the computation is ended, and using roundEnd() to transmit the computing result.
    - If the main thread wants it to stop, the main thread will set a variable to false (of course, there is a mutex), and the class SA will check the variable when a round is finished.
    - It used about 3k+ rounds to get the result.
    - The state of beginning is randomly generated.
- solver.cpp/h
    - Here I packed TA's code into a structure.
- timer.cpp/h
    - This file implements a multi-thread timer.
    - Refreshing the clock is also by this thread (you should tell the timer your QLCDNumber pointer).

Happy Creating Your Chip!