

Ok, I will start to introduce my work. My final year project is Development of a C++-based user-interface for a plasma simulation tool. I will quick review the background and objective of this project first. Plasimo is a toolbox that provides support for the numerical simulation of plasma sources of various degrees of equilibrium. But there is some problem make user hard to research data. First is data are saved in 68 text files and it's hard to match them especially for new users, and then the simulation process is irreversible and cost lots of time. So, my project is build a user-friendly interface to solve these problems and provide some functions to dispose of related data.

Now, I will introduce the developing process and run the program. If you don't mind, could u set a md2d model first and we can simulate it and when we finish the introduce of developing process, we can test my program by this output file directly. (I already had two different output, one is pdpdemo_md2d model and another one is medo.md2d model)

First step is prototype design because when you use an operational interface design to communicate with your supervisor, it easy to give both a directly image of what the final program looks like.

Second step is learning related knowledge of MFC and openGL, I can't only use Year 1's knowledge to build this program, it needs me to do lots reading and practice.

Then I will introduce the main logic of my program design:

1. First is draw the dialog of main page, as you can see, this frame is based on the prototype design, when you draw this, the visual studio will generate related code atomically, and then you will need a command to put real picture into these widgets. In my project, I have use opencv(Cstring), but we can't directly use it, we will need to change the format. So we need this command.

After this, we will write the response command, like we will judge the mouse's location, if it in widgets, we can continues writing following command, like click the mouse, help button or QR code.

Then we will move to next dialog, in this case, when you click the load button in the first page, it will display the folder chooser, you need find the address of where the output file in. and then this system will display the final information of the simulated data.

And then is the main interface, two function will be display here, first is generate 2D image generating and data displaying. Another is using line chart to display the data in history.out file.

To realise these two function, we need match the properties of plasma to the specific text file. For having a more using range, not only in pdp_md2d model, I use CSV file to save the corresponding properties and text files. User could change this csv file to let my program apply to other models. This program will read the csv file first and then put them into a vector. When user click specific property, program will find text file name and then loading them. For generating 2D image, we need do some processes. First is divided the data to independent data based on time, and then separate one time's data to 7 areas. In this case, we find the max and min number in one time, and use their difference value divide 7, we can get 8 numbers by using min value add different magnification of this difference value. And we will transfer these areas to number 1~8, give different colours to them, and we will get the 2D image. Like this, the number of time will be saved in another vector and display here. Then the user could know the time of image and find the data on the right side.

The second function: line chart, it also need some processes, first is loading the history.out file and then save the data column by column. The first column will be used to generate the x-axis, other columns will be used to generate the y-axis information. First we need to set the coordinate system; this program will find the difference value of max and min number in one column and then divided into 5 areas. Write these values to coordinate first, and then the line chart can be draw by lot of points. The value of X and Y-axis will divide the ratio of coordinate and then we obtain the value of x and y. So we can draw the point.

This project has an abandoned version which is using video and image to show the output data from Plasimo. It's inapplicable because it needs lots of time to record videos. The Second version is not perfect either. The running speed will slow down with the increasing number of data and there is not an effective way to locate specific time.

Use one sentence to express my feeling in this project is that Keep learning to solve unfamiliar problems are the big challenge but also the most exciting thing.