# Report of My Final Project

General introduction

In my program, I mainly completed three parts of functions: the first function is to load the data into the program and show the loading progress with the progress bar; Secondly, I completed the mandatory part of data analysis and comparison; Thirdly, I have completed map-related functions, which are respectively to draw the specific user’s motion trail map and the heat map of POI.

Implementation details

**1. Part of loading data:**

**1.1** I created a dialog box called “LoadDialog*”*. Users can choose the data they want to import by choosing the starting and ending time, the longitude and latitude of the location, etc. I will filter the imported data based by the class “InputFilter”, and then save the data they want.

**1.2** I created a c++ class called “InputFilter” to process and store the data item by item. This object of it is built in the “LoadDialog” and moved into the object of QThread named “thread1” to prevent the program from getting stuck in the process of processing a large amount of data.

**1.2.1** As you can see in “InputFilter”, I have customized a structure called cknode. It is used to store information about each access record, including user\_id, location\_id, check-in time, latitude and longitude.

**1.2.2** In the process of reading data, I used three arrays respectively to store access records with different standards. The “cklist (or list)” stores records according to the reading sequence, “user” stores records in stripes according to the ID of users, for example, user[i] stores all the access information of the user with the user\_id of i.Similarly location[i] stores all the access information of the ith location. After the data is loaded, these three arrays will be returned to the main window. Then they will also then be passed to the “AnalyzeFunction” to be analyzed and the “usermap” to be visualized.

**2. Part of analyzing data**

In this part, I create “AnalyzeDialog” to receive the corresponding parameters selected by the user from interface and then pass them to “AnalyzeFunction” through signals and slots. “AnalyzeFunction” is responsible for processing the data and passing the processing result to “AnalyzeDialog”, so it is moved into the QThread “thread2”.Finally, “AnalyzeDialog” draws images and tables based on the results sent from “AnalyzeFunction”. Below I will respectively introduce the implementation process of the four functions "TOP 10 popular visited POIs", "Most frequently visiting 10 users of a POI", "Compare 2 users' activity levels", and "Compare 2 POI's daily active users".

**2.1 The realization process of the “TOP 10 popular visited POIs”**

In the “slot1” function of the AnalyzeFunction, I use the array “count” to record the number of times each location has been visited. I use vector “user[i]” which was built during the loading process to traverse the number of visits the user with id of i chosen by user. If the visit record’s visiting time is within the user's selection range, the corresponding element with the location\_id of x in “count”, which is “count[x]”, will be added “1”. Finally, I find the 10 indexes corresponding to the ten largest numbers in “count”, which is the 10 POI ids with the most visits. In the function named “paint1” of “AnalyzeDialog”, I drew a histogram of the names of these ten locations and the number of visits based on the analysis results. Also, I took the ranking from 1 to 10 as the horizontal axis, and the number of visits as the vertical axis. Meanwhile, I used the result of ”Interpolation“ to fit 200 points and drew a smooth curve.

**2.2 The realization process of “Most frequently visiting 10 users of a POI”**

The realization idea of ​​the function "slot2" in “AnalyzeFunction” is very similar to 2.1. But this time I changed the traversed object from the “user” to the “location”. "Count" records the number of visits by different users this time. The result is to find the ten most visited users, and the function named "paint2" in “AnalyzeDialog” draws the corresponding histogram and smooth curve.

**2.3 The realization process of "Compare 2 users' activity levels"**

In this process, I roughly analyzed the similarities and differences between the two users' checking information. The similarities include the number of places they have both visited and the place where the two users are most likely to meet. The differences include their total number of visits and the total number of places they have visited. In “AnalyzeFunction”, function "slot3" implements this step. I use the array "user[i]" to analyze the access records with the two user ids selected by the user interface. In the process of retrieving the visit records, I used the arrays "check1" and "check2" to record the corresponding visit times of the two POIs visited by the two users, and synchronously calculated the data listed above, and returned the results to the "paint3" of” AnalyzeDialog”. Which finally fill in the results in the form.

It is worth mentioning that in the of "the place where two users are most likely to encounter", the number I presented in the table is the id of the place that two users have visited the most frequently. Because I think it is of representativeness in a certain degree in the enormous data of access records.

**2.4 The realization process of "Compare 2 POI's daily active users"**

In this implementation function "slot4", I respectively traversed the visited records of two places, user[i] and user[j] chosen by the user, and recorded the number of visiting users on the date selected by the user as its DAU during the process of traversing.

**3.Part of painting map**

In order to realize the function of drawing on the map, I chose the Java Script API of Baidu Map Open Platform as the tool for implementation. On this platform, I applied for a Baidu map AK, and then called the offline map provided by this platform. Then I save the map API I need in a file in html format, and then include this file in the resource file section of my program. In this way, I can use QWebEngineView to load the offline map in qt, and use the "runJavaScript()" function provided by the class to call the map's API.

This part of the data processing volume is not much. I have completed the compution of data processing required for the user motion trail map and the POI heat map in the function "usermap". "Usermap" was moved into "thread3".

**3.1 The part of drawing the user’s motion trail map**

In the part of drawing the user's motion trajectory diagram, I imitated the demo on the Baidu platform and modified the functions in the API so that its functions can meet my requirements in qt. In the corresponding function in “usermap”, I sort the access information of the user to draw the trajectory graph in order of access time. The sorting result is returned to “PaintDialog”, and then PaintDialog calls the function named "paintUserMapAdd" through “runJavaScript()” I wrote in the file named "baiduMap" to add the locations to the map in the order of visits one by one. After the points of locations is added, “PaintDialog” calls the function "paintUserMap1" to draw the curve in turn according to the points. To draw the curve, I used several dynamic drawing effects provided by Baidu Maps to draw lines between points.

**3.2 The part of drawing the POI heat map**

I connected the signal of the "paint" button to the function that draws the entire map, which is, to add all the labels of POIs on the map. The implementation method is similar to that of 3.1, adding the specific latitude and longitude coordinates in a vector. After adding the elements of the vector, the "addLabel" function will be called to mark the labels of the corresponding information on the map.

The second step is that after the user selects the POI\_id he wants to see and presses the "find" button, I use “runJavaScript()” to call the function "centerAndZoom" in the API to set the map center fixed on the location selected by the user.

Results

I managed to implement the mandatory task and the part of map in elective task.

In the function analysis interface, the user can freely choose parameters and get results visually from the chart, as well as freely choose the locations or the users to compare their checking information. In the map section, the user can clearly observe the movement track graph of a particular user and the heat map of the points of interest, and he can drag and zoom with the mouse.

Discussion

1. I had a lot of trouble completing the map feature. The first is that the compilation environment is very problematic. The qt library I originally downloaded didn't have "QT WebEngineWidgets", and I started compiling with the "minGW" version of the compiler. But the library needs to run in "MSCV". So I got an error when I added the library to the ". Pro "file. Finally, I re-downloaded qt of 5.14.2-2017mscV-64bit and successfully called "qtWebEngineWidgets".

2. In the beginning, I tried to create a "QWebEngineView" to load the offline map, but the application crashed unexpectedly. To fix this problem, I changed the functions in the API several times, but again without success. Then I tried to open the file in my browser, and it worked fine. After a lot of searching, I realized that it might be the compiler. I tried to fix the problem by using a Release compiler instead of a Debug compiler. In addition, programs compiled with the Release version run much faster than that with Debug compiler. This is evident from the loading speed of the progress bar.

3. Finally, I successfully realized the loading data, analyzing function and data visualization in this program. I was satisfied with the speed of the program in terms of loading and analyzing data, but there was still a lot of room for improvement with the loading speed of the offline map.Besides, I store "user" and "location" as " QVector<QVector <cknode>>", but if I store the data as "map" in QT, the application will call the data faster, because the data structure of the type “map” is b tree. In addition, the speed of the function of drawing maps as well as the design of ui also has a lot of room for improvement.

Conclusion

In the process of finishing this project, I gained a lot of knowledge and improved my ability to apply the knowledge learned in the course to solve problems in practice. I would like to thank Mr. Jin and Ms. Zhao for their help during the course of problem solving and practice, as well as Teaching assistant Zhaoxing Yang for his detailed answers to my questions about project requirements.