Software Engineering Project Report

Belal Hmedan & Deng Jianning
University Bourgogne

Sunday 8th December, 2019

Contents

Introduction

BUILDING an Offline Map Software is a challenging project, because it requires a good knowledge in Geography, Mathematics, Graph Theory, and Computer Science.
At the very beginning of such project many questions must be asked, such as:
Which places should the Map covers? For which application the Map will be used?
Which Information is necessary, and Which Information is useless so it can be
Filtered, so total amount of Data will be reduced.
This Project is a try to answer some of many questions that any computer science Student will ask while starting this kind of projects. Maybe we didn't answer all The questions because of time limit, but we hope that this project will be useful For Other students to get some answers , and to get the fastest way between two Locations, That's why we selected 20 Locations carefully to satisfy the daily needs Of Students like Universities, Cool Restaurant, Pharmacy, Hospital, Shopping Centers, Students Residences, etc.
This project with all cool features and nice looking objects, is nothing more than "Hello Map", because Cartography is a great science developing everyday for both Civil and military use, and everyday there is changes on our planet leads to changes On the Map, starting from Environmental changes, and not ending by Human-Made changes.

Chapter 1

Project Description

1.1 Goal of the Project

The goal of this project is to develop a software to locate (and rate) various buildings, such as schools, university buildings, major offices, hospitals, various shops, cool restaurants, streets, roads, parks, and other interesting points in Le-Creusot. The key idea is to develop a software to visualize Le-Creusot, its streets, some buildings, roads, and parks, in which the user can perform several actions, such as asking for an itinerary between two (or passing through more) points.

1.2 Completeness of the Project

1.2.1 Place Locating

Twenty locations were provided on this project, and there is ability to add many other places due to requirements, but user can't add places, only programmer. User's privilege on this stage of the project is limited to chose only from a list of those twenty places.

1.2.2 Place Rating

This Feature isn't added yet, but it maybe available in next releases.

1.2.3 Routing

Routing in this stage only by foot, expanding travel Methods to car, bicycle, and Train is possible in the following releases.

1.3 Dependencies

Apart from STL, I'm using Boost Graph Library (BGL), which in turn depends on other Boost Library: Property Map, which in turn depends on other Boost Libraries, so I included the whole Boost Libraries at once.

1.4 Data Sets Introduction

Chapter 2

Code Design and Structure

2.1 Algorithm and Shortest Path Design

This part of code organized in 3 different Classes, each of them contains Header, and Source File:

2.1.1 MyGraphBuilder Class:

This Class mainly handles data provided from our Data Structure represented by the Model which depends on the OSM file in its compressed format PBF. This Class mainly takes the nodes included inside ways only from the Model as an input, then builds Vertices and Edges Between Vertices to construct the Graph, which in turn will be as an input to the Next stage which is Dijkstra Algorithm. The Graph in my approach is Adjacency List proposed by Boost Graph Library. The need for this procedure comes from the fact that: finding shortest path will be implemented through an algorithm, and the algorithm deals with graph, so I wrote this class to build the desired Graph from the Data Structure we imported from OSM file. The input for this Class is the Model (Data Structure) extracted from OSM file, And the Output is a Bidirectional weighted Graph. Bidirectional, because it's easy to implement, in case we want improve it to Directional Graph, we must do a complicated constrains, and check each Tag, to see the type of Transportation, and if the way was one direction, or bidirectional. Weighted Graph: because we are calculating Euclidean Distance between the Nodes depending on Longitude, and Latitude.

2.1.2 MyAlgorithm Class:

This Class takes the Graph produced on the past class MyGraphBuilder, and the Source Node which we will start from, and runs Dijkstra Algorithm proposed by Boost Graph Library to Find the Shortest Path Between the Source Node, and all other Nodes in The Graph. Special Function in this class called: getShortPath takes the destination Node, then returns the shortest path between the source and destination only. In case that there was no path between nodes, the path (Vector) will contain only one default node, so we don't face issues due to empty vector.

2.1.3 ShortPath Class:

This Class is just to reduce amount of code in the main window, it doesn't do much, it just takes Source Node, Destination Node, and Model, then it builds the Graph From the Model using MyGraphBuilder Class, and Runs MyAlgorithm Class, and its special Function getShortPath, finally the output is our shortest path.

2.2 Data Manipulation

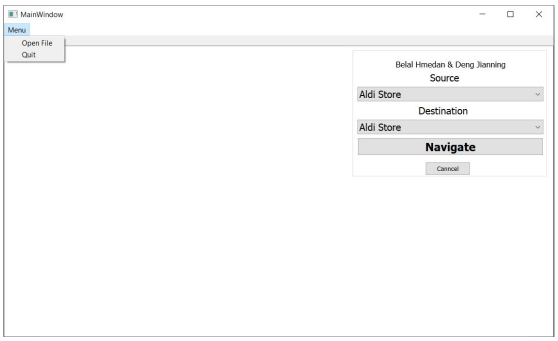
2.3 Rendering

2.4 Routing

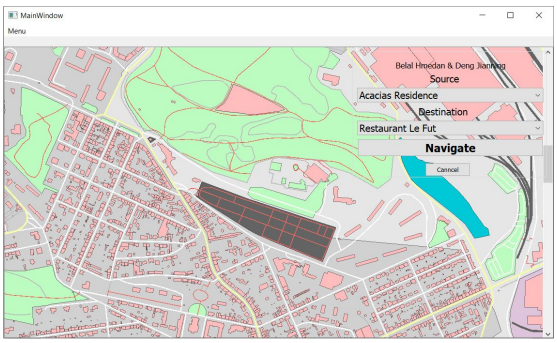
Routing on this approach is just on foot, it's possible to add more transport methods later using Tag filtering, building different transport maps, Vertices, and Edges, but unfortunately time was short for this project ! . Isolated Nodes which doesn't belong to any way are not included in building the Graph, because our purpose is to find a way between two nodes, so I didn't build the graph with all nodes, only connected nodes which belong to way. Euclidean Distance between the Nodes is our way to assign weight to Edges between Nodes, and of course the Way or Path is nothing except group of sequential Edges between connected Nodes.

2.5 GUI

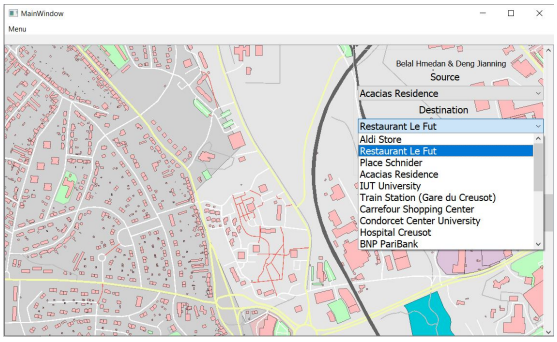
Running the Program that will show a window with a Menu and four buttons Source, Destination, Navigate, and Cancel.



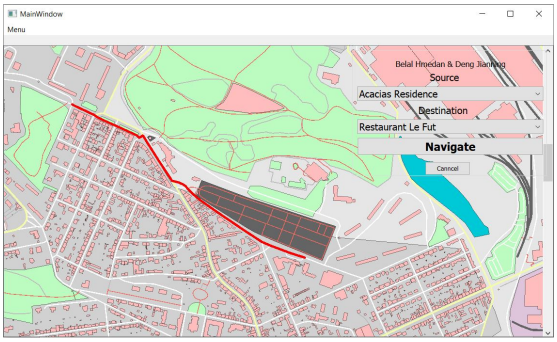
To start working, you should select (Menu - Open) to open the map file, which should be with (.pbf) extension. After you select your Map, the Software loads the map File showing the Map.



The Drop-down lists Source, and Destination contains 20 places for each of them, All on Le Creusot City, where you can select your start point, and destination.



by clicking on Navigate Button, the Shortest Path will be drawn by Red , so you can see the way you should follow.



Cancel button is to hide that path, and you can choose again your start point and target, so your path again will be drawn for you. To Finish, you can select (Menu - Quit) to exit the Software.

2.6 Difficulties during implementation:

There is two kind of Difficulties here:

2.6.1 Difficulties we got rid of it:

First Difficulty was CMake didn't find some libraries:

specially Boost Library, later we excluded Cmake, we depended on Headers only to implement our plan.

Second Difficulty was that Edges can't be built using idType Nodes:

the solution for that was mapping the idType Nodes to unsigned int indexes through a map, so each node has ID, and Vertex Number.

Third one is Redundancy of Nodes in different Ways:

the solution was to compare each node of way to the map we built as solution to problem1.2, so if the node already found, we don't add it to our map, instead we just call it and get Node index.

2.6.2 Difficulties we can solve, but it needs more time:

1.Supporting Different Travel Method.

2.Building Directional Graph instead of Bidirectional one.

3.Adding Reviews, and stars options to the GUI.

4.Adding locations on the Map.

2.6.3 Unexpected Difficulties

Losing member of the team at critical time of the project doubled the responsibilities to do more than what we have already planned to do, and make us work under pressure in some parts of the project which we didn't have ideas about it before.

Chapter 3

Project Management

3.1 Overview of the Project Planning

We Received the Project on 30Th of September,it took us about 2 weeks to team up in a group of 3 students, then we started The Research for the Project since 15Th of October, Research for this project took 3 weeks until 6Th of November, There was many choices, and due to strict conditions such as: the Project should work under different operating systems, we excluded promising solutions such as "Libsmscout" which was easy to use Library with advanced features, but it doesn't work on Windows, only Linux, that lead us to more complicated choices such as "Libosmium" and "Boost" Libraries. The Implementation started on the 6Th of November, each student worked on different part of the project: "Data Manipulation and Map Rendering" for Deng, "Shortest Path Algorithm" for Belal, and "Graphical User Interface" for Our Third Partner. Ten Days later At 16Th of November Deng Proposed Three Solutions for dealing with the Project:
Solution 1: Using one-for-all library.
Solution 2: Using OSM data access library.
Solution 3: Display the map using Image. At that Point Belal and Deng Agreed on the Second Solution, because we want to propose something different and more advanced than the Image. Our Partner preferred to go on different Direction with Image Solution, so we had to work on the project as a team of two members only !. Following the advice to "Expect what is Unexpected " we reduced the project size and features so we can finish it before Deadline. A week later, on 24Th of November we had our first working project with Minimum features available, a week later on the 1st of December We did a simple GUI to deal with our Map. Writing The Documentation took a week until 7Th of December, and Writing the Report in Latex took another week up to 14Th of December, so We did it on time.

3.2 Work Schedules

3.2.1 Plans VS Actual

3.3 Conclusion

Chapter 4

Summary

Bibliography

[1] Jeremy G. Siek, Lie-Quan Lee, Andrew Lumsdaine. *The Boost graph library : user guide and reference manual.* , “Book," *BGL*, pp. 161-322,Pearson Education, Inc., Reading, Massachusetts, 2002.