Masters in Computer Vision
Software Engineering Project
2013-2014

Ozan

Oksana

Klemen

Natalia

January 2014

Contents

CHAPTER 1

Introduction

As a project at Software Engineering class we had to develop an application that to some extent mimics well known Google Earth. The main idea is to give an user who is unfamiliar with Le Creusot the tool that enables him to search for shortest path between two points on the map, enables him to create an itinerary containing different types of points of interest. We had to do all this in C++ and in Matlab.

CHAPTER 2

Project management

In this chapter we want to present how our group organized and managed this project. As we all had a bit of software designing experiences, we knew how important a proper plan and research before the implementation is. Unfortunately, we were also aware that no matter how good the plan is, we will be forced to adjust it during the implementation, because of the things we did not take into account while planning or because some things turned out to be different to what we assumed. Because of that we decided to follow the iterative and incremental development model, which enabled us to adjust our plans after each of the implementation iterations. A schematic representation of the iterative and incremental development model can be seen on figure 1.

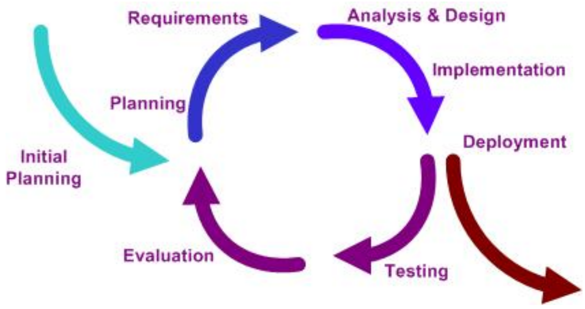
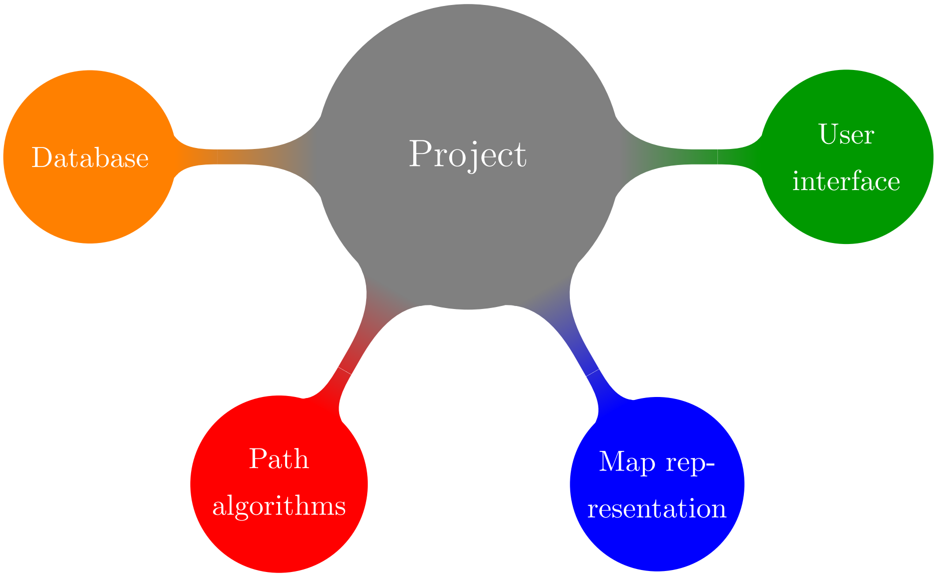


Figure 1. An iterative development model

1. Basic building blocks

To be able to fully manage the project at all times we decided to split the project into four main parts:

- User interface
- Database
- Map representation
- Path algorithms



1.1. UI - User Interface.

It is the part of the application that is responsible for user interaction with the application. Main goal is to make the interface as user-friendly as possible. In the beginning we made a rough sketch of how the interface should look (figure []), but we, as expected, changed it a little bit during the development. The UI is presented more in details in chapter [UI USER MAUNAL].

1.2. Database.

Our database consists of two parts. The static part, having the information about the roads in Le Creusot (described in []) and the dynamic part with the information about the points of interest. From the beginning we were deciding between using XML based and relational database. During the consideration we took into account the following facts:

- Most of the data is static - information about the roads in Le Creusot will not change;
- Dynamic data will change rarely - user will not often update information about the points of interest;
- Relatively small amount of data - Le Creusot is a small city, so there is not a lot of information about the roads. This gives us the opportunity to read all the information to memory at the beginning, reducing the latency caused by queries to the database;
- Easiness of install and mobility - using relational database, requires installation of different software (sql server, connectors, etc.) on the clients computer, which we wanted to avoid.

Because of all these reasons we decided to use XML based database. We have the information about the roads in one osm file, while the information about the points of interest is in xml file. We will describe both of them more in detail in later chapters.

1.3. Map representation.

TODO

1.4. Path algorithms.

This part consists of all calculations about paths, distances, travelling times and optimizations of the paths. Our main goal is to make the algorithms work as quickly as possible, taking into account all the restrictions road networks has (oneway streets, footways etc.). We also want to make the algorithms as reusable as possible, to reduce the redundancy in development and minimize the possibilities of errors.

2. Softwares used for project management

Github, skype etc.

3. Meetings

some bullshit about that

CHAPTER 3

Initial planning

As already mentioned in the previous chapter, we decided to use the iterative and incremental development model. Before starting the iterations we spend quite a lot of time on the initial planning, with special attention to user and general project requirements analysis. From the project's instructions we were able to identify the following major user requirements and construct use case diagram:

- user should be able to enter start and end point either by:
  - mouse click
  - specifying latitude and longitude
  - selecting a point of interest
- find shortest path from point A to point B (by foot or by car)
- find all points of interest in a certain radius from point A (by foot or by car)
- construct an itinerary from point A to point B with points of interest in between (with max distance limit)
- view, edit and add points of interest (POI)

For better

1. Shortest path A -¿ B

2. title

!!!ADD image of the use case diagram

CHAPTER 4

Plan of implementation

After having the user requirements, we decided to make the global plan of implementation. Here we had to take into account, that we have to develop the application in C++ and Matlab. As show in the figure[] we have already splited the project into four main parts. Because the user interface and map representations require completely different approaches in C++ and Matlab, we decided to split them into two parallel projects, one almost independent from another. On the other hand, we decided to use the same algorithms for both projects. The main reason was to minimize the redundancies in the research and development stage and minimize the possibility of errors. This way we could have the same algorithms for both, differing only in pure implementation details.

Each of was delegated to oversee one of the parts (as shown in figure []). We have to emphasis that this does not mean he/she developed that part by him/herself, not even that he is solely responsible for it. No matter the delegation, we all worked on all parts of the project, either in research, design or implementation stage.

CHAPTER 5

Path algorithms

1. Street data

What did we use...open street map...shot introduction into nodes, relations etc.

2. DB class structure

3. Algorithms - shortest path

3.1. A*.

4. Radius search

5. Bicycle search

CHAPTER 6

GUI

1. C++

1.1. UI.

1.2. OpenGL.

2. Matlab

2.1. UI.

2.2. library for maps.

CHAPTER 7

c++ vs matlab

Bibliography

[1] J. D. Bovey, M. M. Dodson, The Hausdorff dimension of systems of linear forms *Acta Arithmetica* (1986) 337-358.

[2] J. W. S. Cassels, *An Introduction to Diophantine Approximation*, Cambridge University Press, 1965.