

Computer Seminar Final Assignment

Car Navigation System

Chua Tzong Lin
B7TB1703

January 28, 2019

1 Introduction

The Car Navigation System is written using C Language by utilizing the OpenGL libraries. The program is compiled using the GCC compiler in LINUX Ubuntu environment. The added features are mainly referenced from Google Maps.

2 Basic Requirements

2.1 Load Given Data

The program should be able to load the given data file, *map.dat*, into the program.

2.2 User Data Input

The User should be able to input their desired starting crossing and final destination via ID, Name or Coordinates.

2.3 Shortest Route Calculation

The shortest route connecting the Starting Crossing and Final Destination should be calculated using Dijkstra's Algorithm.

2.4 Map Display

A map should be displayed in a Graphical User Interface by using OpenGL libraries.

2.5 Animation

A point should be animated to move from the starting position to the final destination.

3 Extra Features

3.1 Error Messages

The program will terminate automatically if an invalid input is inserted.

3.2 Stop Addition

Allows the user to add a rest stop or visit specific place during navigation.

3.3 Real-time Features

3.3.1 Traffic Simulator

The *random function* is called to give a crude simulation of traffic. This is done by swapping the buffers at random frequencies by utilizing both the *random* and *sleep* functions.

3.3.2 Traffic Light Simulator

At each junction, there is a possibility of being stopped by a red light. Similarly a random function is called to simulate the possibility of being stopped by a red light. A junction with more intersecting points will have a higher possibility of being stopped.

3.3.3 Cruising Velocity

Since the frequency of buffer swapping is unconstant, by using the *clock* function, the cruising velocity is calculated and displayed in the GUI.

3.3.4 Incoming Action

The distance until the pointer changes it's moving direction is calculated and displayed together with the command of turning left or right.

3.3.5 Current Route

The name of the current is displayed for reference during navigation.

3.4 Route Display

3.4.1 Highlights the Entire Route

The shortest route taken by the moving point is highlighted in green.

3.4.2 Highlights Current Route

The current route is highlighted in red.

3.5 Text Display

Instead of displaying all the crossing names, only the names of the crossings that will be passed through are displayed.

3.5.1 Text Font

The font *Japanese Gothic* is used to display the texts in the GUI.

3.5.2 Text Colour

The name of the incoming crossing is coloured in red while the rest of the crossing names is displayed in green

3.5.3 Text Sizing

The size of the text displayed will be rescaled together when the map is zoomed in or out.

3.5.4 Text Rotation

The displayed text will always be displayed upwards even if the map is rotated in the first person view or using the rotation function.

3.5.5 Text Translation

Text displaying real time information will be translated to retain their positions when the map is panned or resized.

3.6 Point of View

Two views are available in the system, 1st person view and map view. The keyboard key 'T' is used to toggle between the modes.

3.6.1 1st Person View

The default view of the system. The map is rotated so that the direction of the motion is always facing upwards of the screen.

3.6.2 Map View

The map is not rotated, the global axis is always pointing in their respective directions.

3.7 Animation

3.7.1 Traffic light

The sign of the traffic light is indicated by animating a red marker when the moving pointer is stopped at the junction vice versa.

3.7.2 Position Indicator

Instead of using a dot to indicate the current position of the animated moving point, an arrow is used to indicate position of the current position. Not only does this allow the user to perceive the current location of the point, the arrow also carries information on the moving direction.

3.7.3 Moving View

In both views, the displayed portion of the map will move according to the moving point to observe the motion of the animated arrow. However, if the map is being panned, this animation ceases.

3.8 User Interaction

The user is able to reset all changes made to their view by pressing the keyboard key "R".

3.8.1 Map Sizing

The view of the map can be zoomed in or out by pressing the keyboard key "Z" and "X" respectively.

3.8.2 Map Panning

The user is able to pan around the map using the keyboard keys "A", "S", "D" and "W". During panning mode, the parameters are set so that the user is able to pan normally in the 1st person view and map view. The moving view function is turned off to provide the user full control of the viewing location.

3.8.3 Map Rotation

The user is able to view from any angles they prefer in this mode, the moving view function will still be turned on.

3.8.4 Combination of Panning and Rotation

In this mode, the panning parameters are set so that the user will be able to pan normal even after rotating the map to the orientation they prefer. The moving view function is switched off to provide the user a full rein to manipulate their desired view.

4 Conclusions

This project improved my understanding of C language and the OpenGL libraries. I managed to use the knowledge from other subjects when adding extra features to my GUI. Overall, this project has shown me that programming is not just about the language and syntax, it covers many fields of studies in a whole.