User Manual for Demo Application 1

1. Introduction

The demo application is implemented to show some features of EgeNav. To run the demo application, just compile and run the *tr.edu.ege.cs.egenav.demo.DemoApplication* class.

Internet connection is required for downloading maps and getting directions. The demo application comes with some preloaded map tiles.

Also note, this application uses Google Static Maps service for getting maps through EgeNav framework and there is a limit for downloading maps. If the user is connected to the internet which is shared by many people (such as a university campus) this limit might be already exceeded.

2. Demo Application

In this section, the usage of the demo application will be described. A screen capture of the application at start-up is shown in Fig. 1.

The demo application supports

- setting map type and zoom level,
- dragging map to explore other regions,
- traveling from one point to another point,
- getting textual, audio and visual direction instructions for travelling from the current location to a destination location,
- simulating a real-time travel, passing from many locations which are defined in a file.

To change the map type and zoom level, *Map properties* panel should be used. When the map type and zoom level is changed to *Hybrid* and *16* respectively, the map shown in Fig. 2 is shown to the user. Note that, this requires an internet connection.
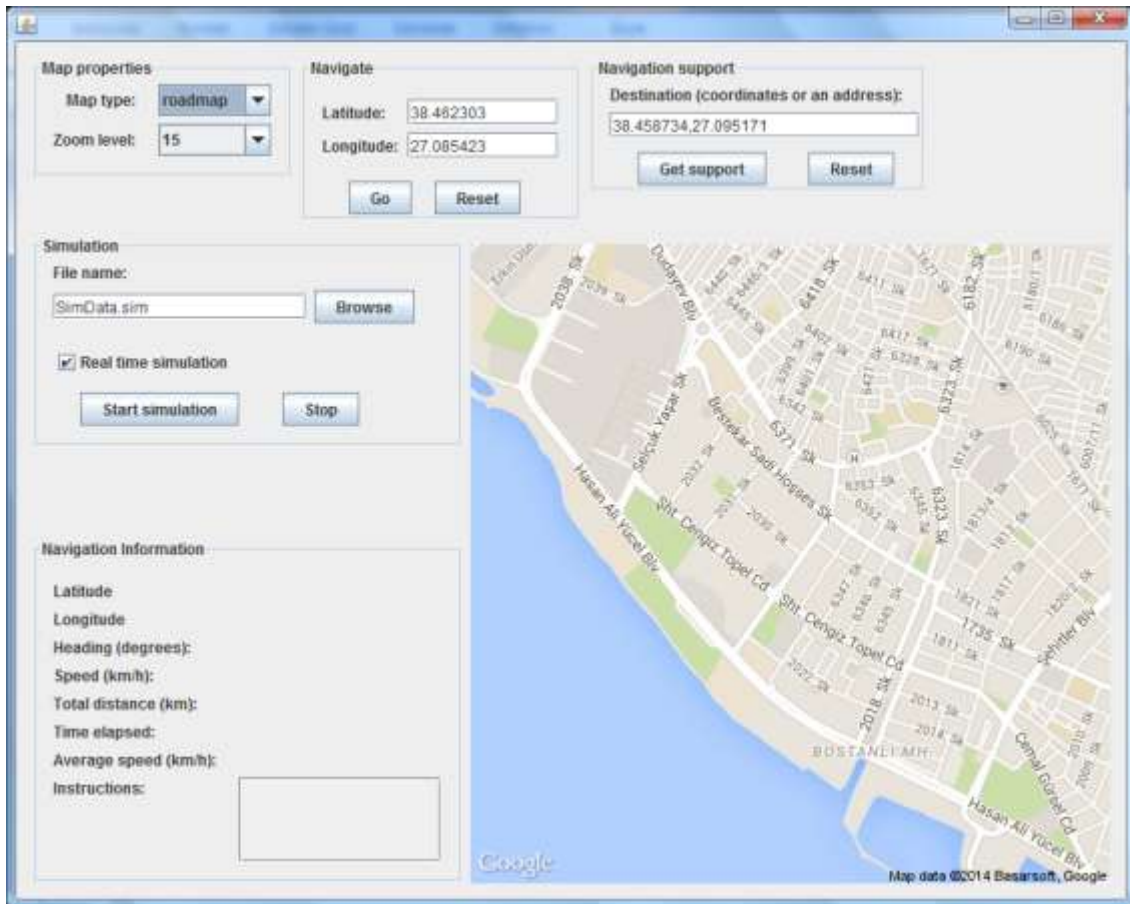
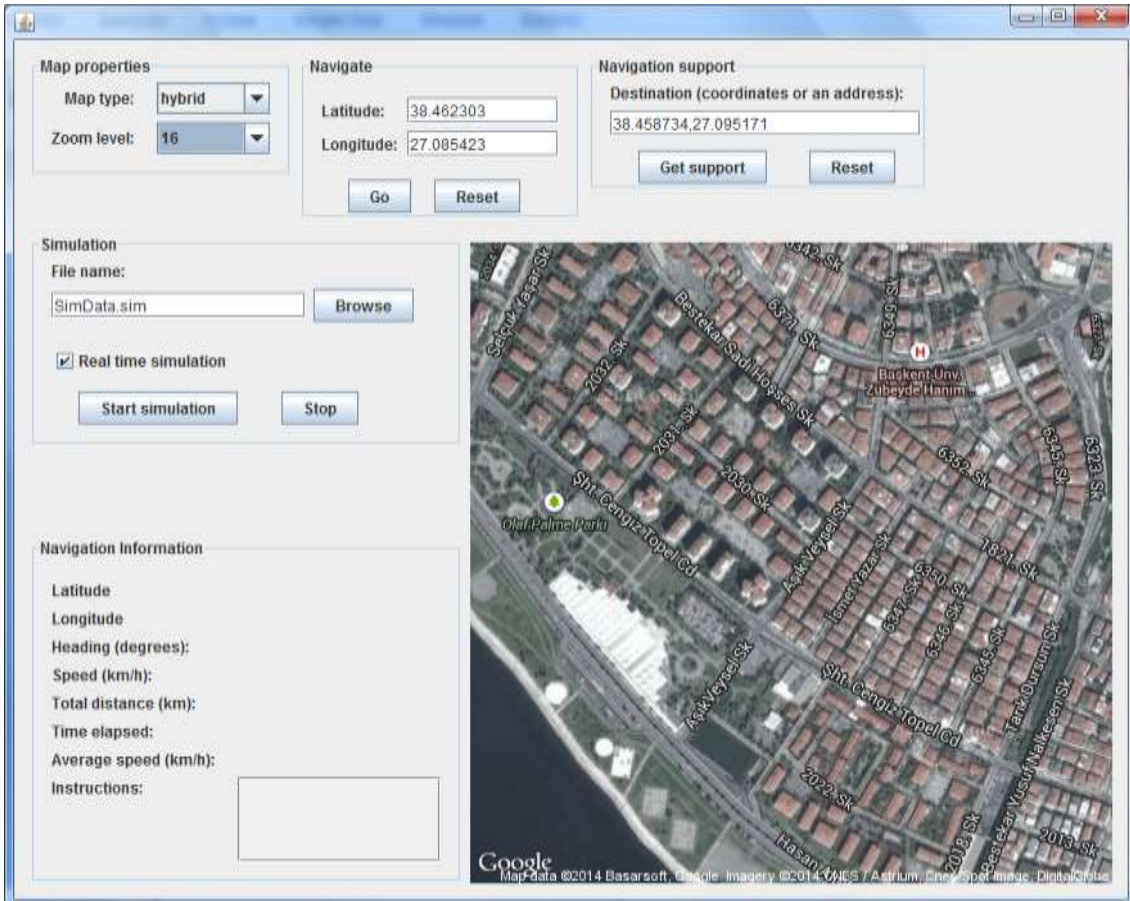Fig. 1. Screen capture of the application at start-up

Fig. 2. Changing map type and zoom level

Now, let's go back to Fig. 1, the figure which shows application start-up. If you have changed map type and zoom level, please change it back to the original or you can close and restart the application.

By dragging the map, users can explore other regions of the map. The demo application comes with 9 previously cached map images. The 9 maps contain the map which is shown at application start-up and 8 neighbor tiles of this map (neighbor tiles to the north, south, west, east, north-east, north-west, south-east and south-west). These regions can be explored without an internet connection, provided that map type and zoom level does not change, because of the employed caching strategy. If the borders of the combined 9 maps are exceeded, then required tiles are downloaded from the internet to show the map to the user. This caching strategy is useful, when the user usually goes by the same route. For example, a person going from his/her home to work and vice versa every day. In this situation, no maps are to be downloaded, because they are cached previously.

By using the *Navigate* panel, the user of the application can go to desired coordinates on the map. When the application first starts, there aren't any passed points. Now, restart the application and press the *Go* button with pre-entered coordinates in the *Navigate* panel. Nothing is shown on map because this is the first location. If we change only longitude value to 27.095423 and press the *Go* button, then traveled route and heading direction is shown to the user in blue by using a line and an arrow. This is shown in Fig. 3. Note that, in the *Navigation Information* panel, coordinates, heading, current speed, total distance, time elapsed and average speed are shown to the user. If the new location is outside of the map region, then the map is refreshed to show the new location.

By using *Navigation support* panel, users can get direction instructions from the current location to a destination location. Destination location can be an address or coordinates. In this demo application, Google Directions service is used through the EgeNav framework. To get navigation support the user should have a current location. If we get navigation support for pre-entered coordinate values by pressing the *Get support* button, the destination route in red will be shown to the user on the map as shown in Fig. 4.

The demo application can also simulate a journey composed of many waypoints. For this purpose *Simulation* panel should be used. Coordinates of the waypoints and time difference (in milliseconds) to the previous point should be entered as a comma separated list in a text file. A sample simulation data file is shown in Fig. 5. This file is included in the root directory.
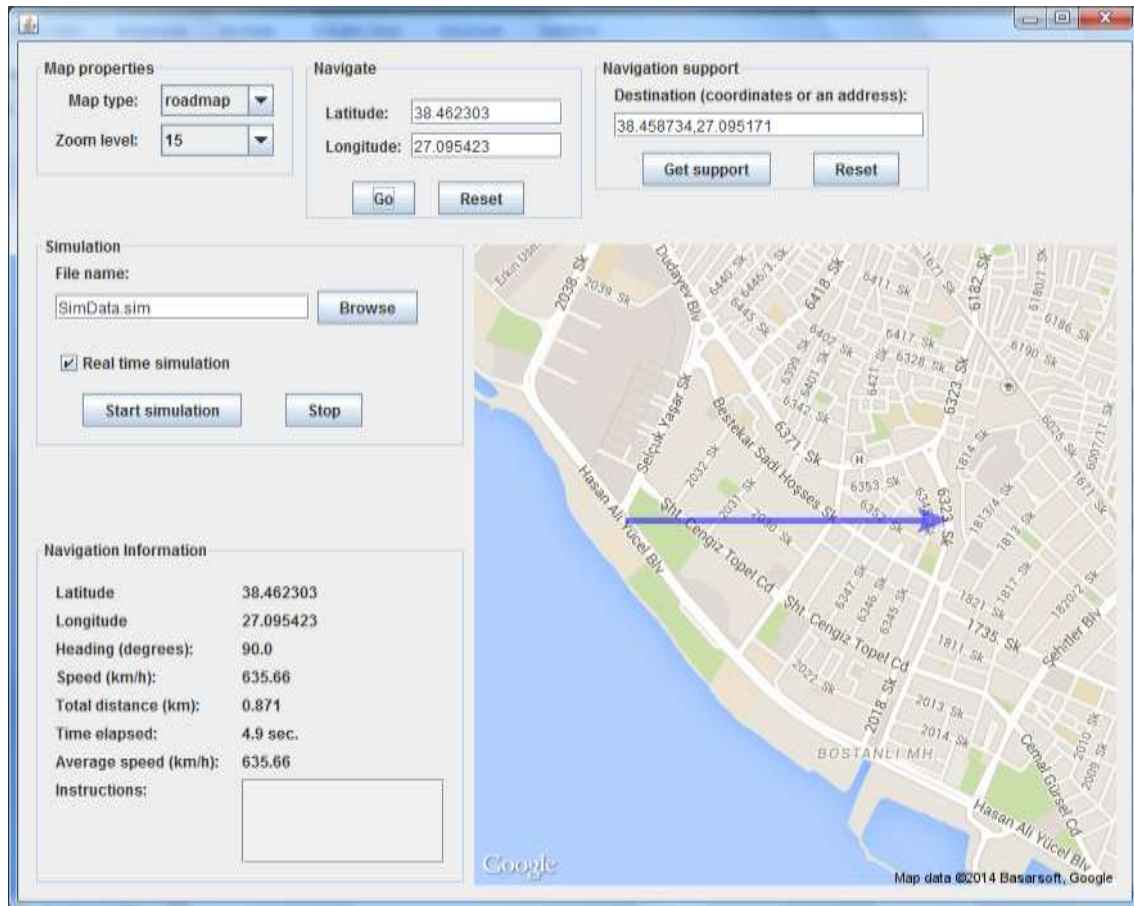
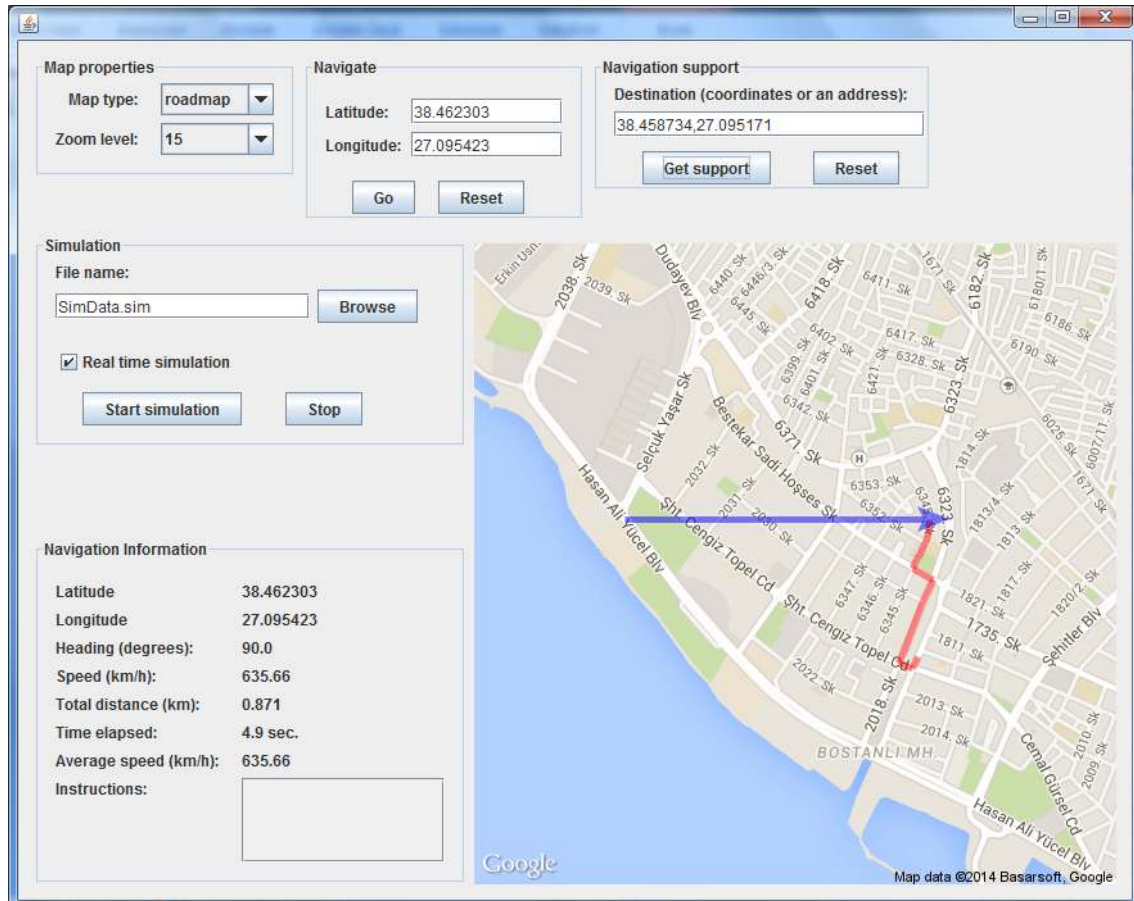Fig. 3. The navigated route and heading direction

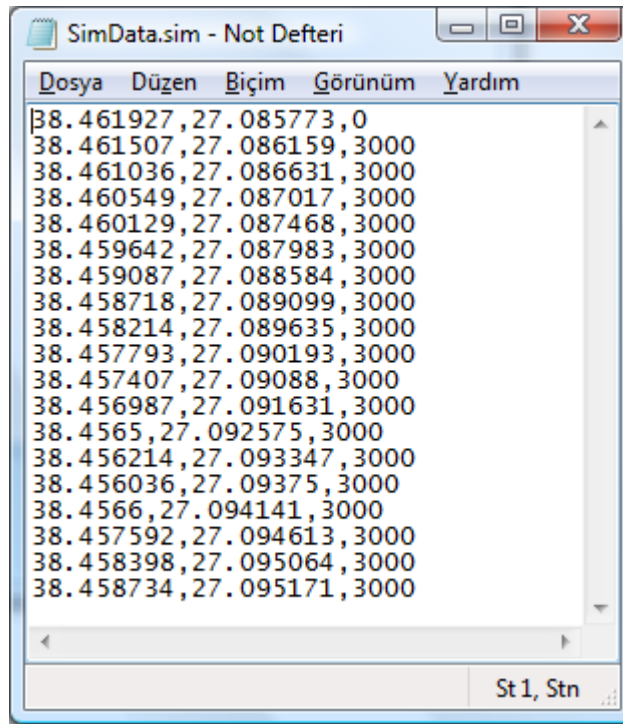

Fig. 4. Getting visual navigation support

Fig. 5. An example simulation file

To start a simulation, a simulation file should be entered in the *Simulation* panel and there should be a navigated point. Then the simulation can be started by clicking the *Start simulation* button. At any time the user can stop the simulation by pressing the *Stop* button. Simulation can be performed as real-time or non-real time.

Simulation task is performed using a thread. This thread reads navigation data from the file and feeds them to map panel to simulate a GPS. If the simulation is real-time, the thread sleeps for a given amount of time between waypoints.

Now do the following steps to start the predefined simulation:

- Close and restart the application.
- Press *Go* button in the *Navigate* panel to go to the specified location.
- Then press *Get support* button in the *Navigation support* panel to get directions and instructions to go to the specified destination location. The route to the destination point is shown in red.
- Next, press *Start simulation* button.
- Now the predefined simulation should start.

In Fig. 6, 29 seconds after the simulation was started is shown. The followed route is shown in blue. Note that, in the *Navigation Information* panel, coordinates, heading, current speed, total distance, time elapsed, average speed and textual direction instructions for

reaching the destination point are shown to the user. Visual direction is shown in red to the user. In addition to these, to provide audio support, textual directions are read out loud by a robotic voice.
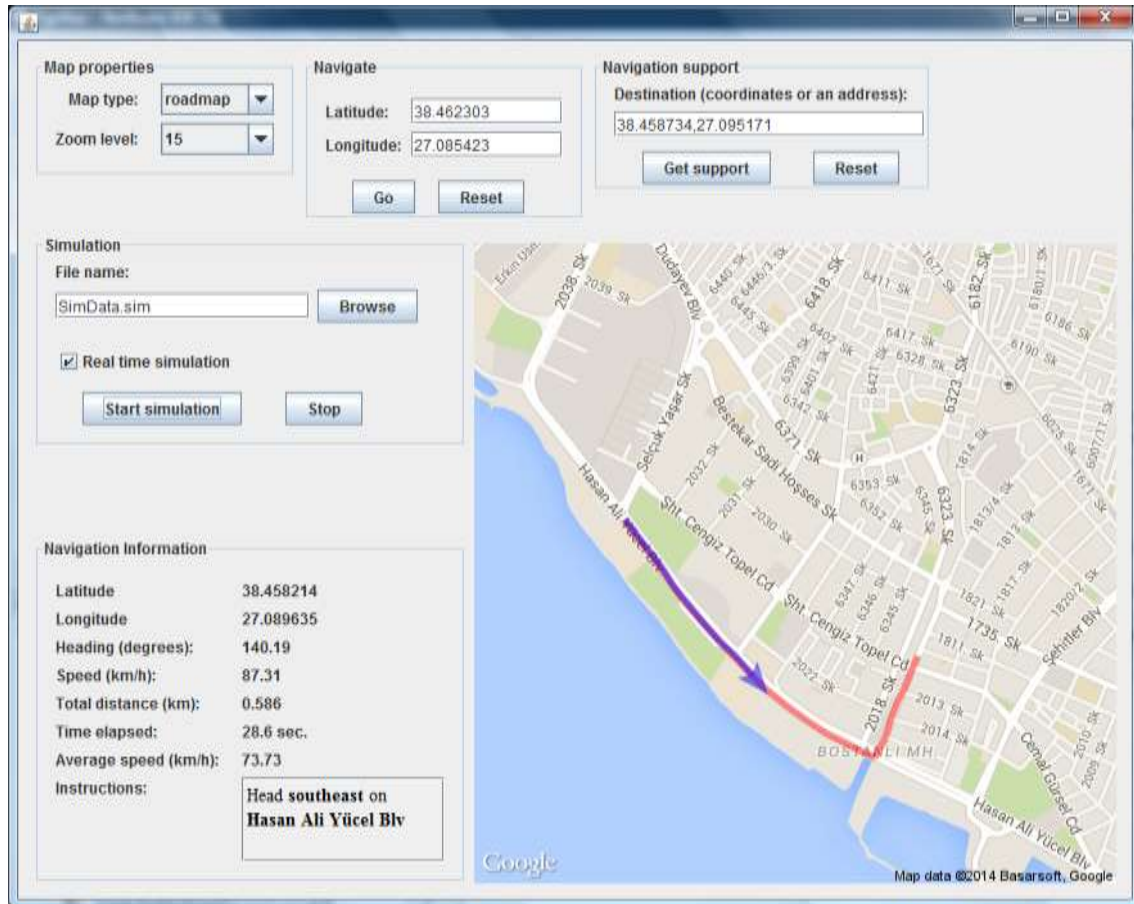

Fig. 6. Getting navigation support after the simulation was started