Software Construction 2 (IE-B2-SO2)

Lab Exercise 4:
Tracking

HAW Hamburg
Fakultät Technik und Informatik
Department Informations- und Elektrotechnik

This exercise serves as a motivation for further exercises and does not have to be completed. Instead, you will do an old exam.

1 General Notes

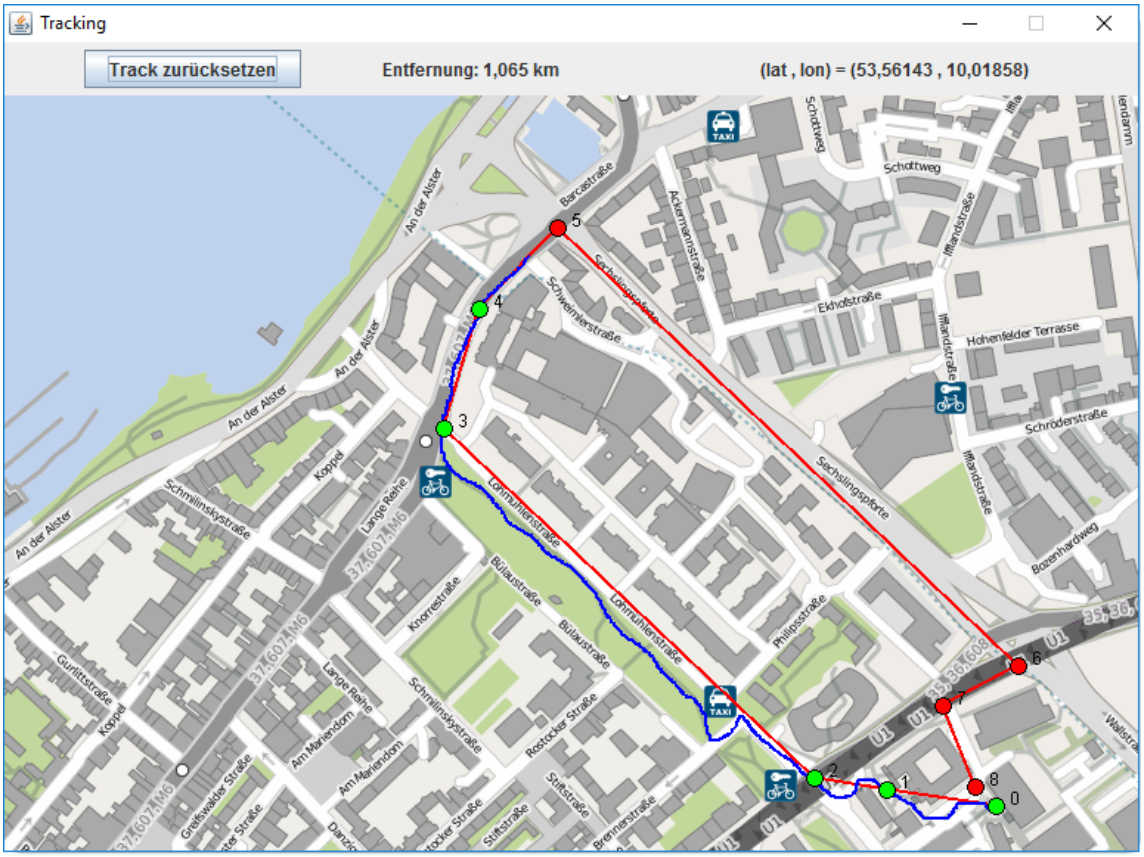
The instructions given in lab exercise 1 apply. Please note also:

- Primary Learning Objectives: Reading text files, Graphical User Interfaces (GUI)
- For the source code, please create a package called *lab4.geoPosition*

2 Overview

Get some fresh air! - Well, go first purely "virtually" to the fresh air. Your way leads you from the HAW along the Lohmühlenpark around the hospital St. Georg.

In this lab you will simulate tracking data (e.g. recorded while jogging or cycling) by moving the mouse along a route on a map. The route must be read from a text file. The following figure shows an example of a possible solution for the application to be created. The route with its waypoints is drawn in red, while the virtual route is shown in blue. Waypoints already passed are filled in green.



3 Exercise

Create a program with the following functionalities.

3.1 Mandatory

- The application uses the map provided in file *OSM_BerlinerTor.png*[1].
- The application reads the coordinate pairs stored in the *Route.txt* file and displays the route consisting of these coordinates on the map.
- Users can draw a route from waypoints by moving the mouse over the map while holding down the left mouse button.
- Users can delete the current route.

3.2 Optional

- Use geographic coordinates (*lat, lon*) instead of image coordinates (*x, y*). The corresponding coordinate pairs of the route are stored in *RouteGeo.txt*.
- The application displays the total length of the rout being tracked in kilometers.
- The application recognizes at which waypoints of the route you have already passed and displays them in a different color - as shown in the figure.

3.3 Hints

- Map: Please follow the hints given in lab 3 (display, file paths).
- Text file: Use *BufferedReader* to read the text lines and separate the values in the text lines using the *split(",")* method call..
- When the mouse is moved with the key pressed, the method *mouseDragged()* of a connected *MouseMotionListener* is called..
- Conversion to geographical coordinates: The file *OSM_BerlinerTor.txt* contains the latitudes and longitudes of the upper left and lower right corners of the map.
- Recognize waypoints of the route: For example, add a method to the *GeoTrack* class from lab 2 that gets a route and returns the number of waypoints passed. To do this, go through the distance of the *GeoTrack* object and first compare the distance to the first waypoint of the route. As soon as this is "small enough" (e.g. 25 m), compare the other waypoints with the second waypoint of the route until either the entire route or the tracking route has been processed
- Please also follow the hints given in lab 3.

[1] Free OSM-map (http://www.openstreetmap.de/), created with MOBAC (http://mobac.sourceforge.net/)