

Sheet 1

Submission deadline: Fri 27.04.2007, 11:00 a.m. (before class)

General Notice

To be admitted to the final exam, every student has to submit solutions for at least 50% of the exercises and present one of the solutions in class. The exercises should be done in groups of two students. One bonus point for the final exam is granted for every reasonable solution of a complete exercise sheet.

Introduction

The JAVA program `mobrob` is the basic software framework used in this course. The source code can be obtained from the class website¹. The program can be compiled and run using the Sun JDK 1.5². At this point, no additional libraries are required. Eclipse³ is a free and convenient development environment we recommend you to use.

In this first version, `mobrob` will be able to read CARMEN's `.log` files⁴. These files contain simulated or recorded data of a robot's position, odometry readings, laser scanner measures and lots of other information.

Complete the exercises and submit your source code via the class website. Test your program with both log files supplied with the source code. Please consider in-code comments, self-explanatory names for variables and methods and well-organized source code in general. Make sure that your code compiles without warnings.

Exercise 1:

Complete the method `readMessage` in `Logger.java` and the `RobotLaser1Message` class to parse the tokenized laser range data. A `RobotLaser1Message`-object must be created and added to the `Logger` for each line in the log that starts with the tag `ROBOTLASER1`. The values of all fields from these lines (see header of `run.log` for a list) need to be parsed and stored in member variables of the message objects. Hint: the super class can take care of the last three fields in a line.

¹<http://ais.informatik.uni-freiburg.de/teaching/ss07/robotics/>

²<http://java.sun.com>

³<http://www.eclipse.org>

⁴<http://carmen.sourceforge.net>

Exercise 2:

Complete the method `paintComponent` in `VisualisationPanel.java` such that the method `paintRobotLaser1Message` is evoked for each message with the tag `ROBOTLASER1`. Complete the latter method to draw the pose given by the fields `laser_pose_x`, `laser_pose_y` and `laser_pose_theta`, as well as the range data, with one line for each beam. Your result should look like the screens in Figure 1.

Exercise 3:

Laser scanners are regularly used on mobile robots to obtain information about its surroundings, especially the location and shape of obstacles. Suppose a laser scanner that is mounted 30cm above the floor facing forward, and scans a plane parallel to the floor with an opening angle of 180° . In which cases will objects around the robot be missed on the scan readings? Write your answers into a file `answers.txt` and submit this along with your source code. Hint: Think of specific positions, shape and nature of the objects (6 answers).

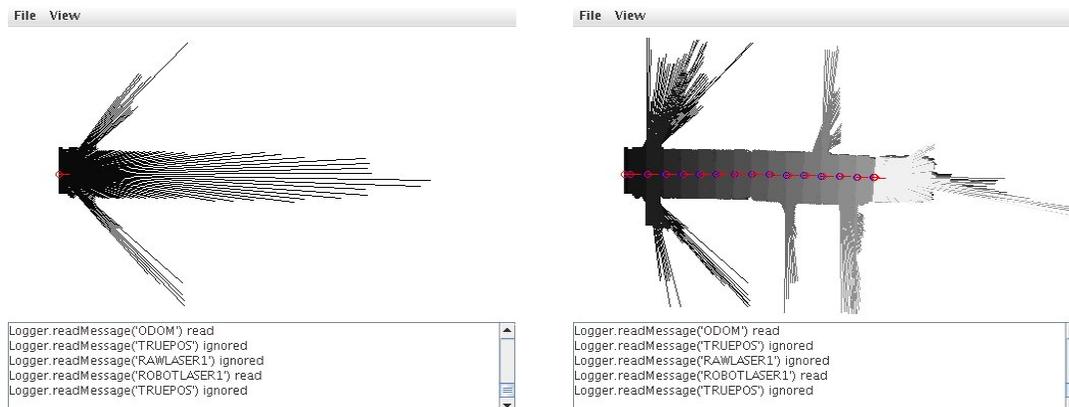


Figure 1: Screenshot of mobrob after reading the odometry data and range readings in `run.log`, for one frame (left) and after painting all frames (right).