Assignment 3, Due November 4th, 2011

(Worth 10%)

1. 40%) Create a program that would implement Occupancy Grid Mapping. Use the random walk implemented in Assignment 2 to have a single robot wander around the environment. Collect the laser data and use them to implement an occupancy grid. Use the provided code as a guideline on how to store and display an occupancy grid in the form of an image. In addition display the path of the robot. Remember to consider *Occupied*, *Unoccupied* and *Unknown* Cells.
2. 40%) Based on question 1, implement a frontier based exploration scheme. See the class notes on exploration. Search through the occupancy grid and find a frontier cell. Remember a frontier cell is an unoccupied cell that has unknown cell(s) as neighbours. Set the coordinates of the frontier cell as the new destination (instead of random) and use the potential field approach to navigate to it. As the robot approaches the destination recheck that it is still a frontier cell. Continue selecting new frontier cells and navigate to them until there are no more frontier cells. In addition display the path of the robot.
3. 20%) Use the provided logged data publisher instead of a stageROS simulation to construct and display an occupancy grid. The data are from an early experiment with a hokuyo laser range finder mounted on top of a Nomadics scout robot. The robot pose as published comes also from real data and thus is corrupted by odometric error. In addition display the path of the robot. (This part would be available on Tuesday 25/10/2011).

Note:

There is a difference between image coordinates and Stage coordinates. See sample code and implement accordingly.

You need to have opencv 2.x and some boost packages installed. For electric users, these packages are: libboost-thread-dev, libboost-date-time-dev, libopencv2.3-dev. (For diamondback user, it's slightly more complicated...)

For the sample code:

- press SPACEBAR to save snapshot of occupancy grid canvas into folder where user opened the executable in.
- press X or x to quit.
- read over the code, and pay attention to the 2 TODO comments especially

Other new things in the project, i.e. why did I have to include a zip of the entire project:

- CMakeLists.txt has some lines needed to link project with Boost and OpenCV libraries
- manifest.xml has ros_dep line that asks project to depend on EXTERNALLY INSTALLED OpenCV library (recommended by ROS, see http://www.ros.org/wiki/opencv2)

For all questions prepare a written report with a screen (images saved in your program; see attached code) that document the progress of the mapping. Compare the two approaches

What to submit:

Submit the source code together with a written report through WebCT.