Assignment 1b

In this assignment you should implement a Particle filter localization algorithm. Use thebag file provided from an actual run of a Turtlebot 2 robot moving through the environment; subscribe to sensor readings from the laser and the encoders (odometry); estimate the pose of the robot. Consider a known map of the environment and an initial unknown pose of the robot (global localization).

a. **Map:** Use the bitmap provided as your known map of the environment. Your first step should be determining the coordinate transformation between the bitmap image and the robots coordinates. Use the yaml file for the required information

b. **Number of Particles and Initial Distribution**: Experiment with different number of particles (1000, 10000, 100000?). Distribute the particles uniformly throughout the map. **Remember**, there is a known map so you know the dimensions of the environment; what is not known is the initial position of the robot. When generating the particles ensure that no particle falls inside an obstacle; use the transformation from world coordinates to image coordinates for checking if the position is occupied.

c. **Propagation**: Use the odometry readings to calculate the linear and angular velocities, and then propagate the particles forward accordingly.

d. **Updates**: Experiment with a varied number of laser readings to perform the update. Recommended values are 1, 2, 3, and 10. Compare the trade-off between efficiency of computation (1 reading is faster to compute than ten) and speed of convergence to the correct solution. For each laser reading use the Bresenham's line algorithm, or any other line drawing algorithm, to identify the respective reading for each particle. In other words, find for each particle what would have been the return using the map. Use difference between the real and the expected measurement to update the weights, you can use a Gaussian weighting function with a small standard deviation.

e. **Synchronization**: Please ensure that the laser data are synchronized with the odometry. Remember that each message is time-stamped, as such you should interpolate from the last known pose to the pose laser data was recorded using the velocities of the robot.

f. **Resampling**: Is already implemented for you for this assignment.

In your written report comment on the trade-offs between speed of convergence to a correct solution and computational cost.

Notes:
Your task is to implement the uniform initialization of the particles over the free space of the environment in pf.cpp, the propagation step in pf2dlocalizer.cpp, and the update step in pf.cpp for a particle filter with all related necessary functions if not present in the provided source code.

To be compiled, the source code should be in the source folder of the catkin workspace.

Once implemented, to test the particle filter:

1. modify "image: ...." with the correct path to the bitmap file pf2dlocalizer/maps/amoco_hall/amoco_hall.yaml
2. in one terminal, run the command "rosbag play assignment_1b_no_map.bag"
3. in another terminal,
    a. export ROS_NAMESPACE=robot_1
    b. run the command
    "rosrun pf2dlocalizer pf2dlocalizerNode _environment:=<path to the yaml file>"
4. you may open rviz for verifying the result wrt the motion of the robot

You may tune some of the parameters that can be found in the constructor in pf2dlocalizer.cpp.