# Homework Assignment 5:
# *Open-Ended*

**Overview**

This assignment is not as structured as the previous assignments. You have a choice of what you want to implement. The three options are described below. You should implement and test your project in a similar manner to the previous assignments. I will assign up to 70 points for the implementation and 30 points for the evaluation. You don't need to test on the actual robot.

1. Monte Carlo Localization on black and white tiles

In this option, you will modify your particle filter code to use a different sensor. You will simulate the "line" sensor on the bottom of the Sparki robot. The line sensor detects whether the surface beneath the robot is black or white.

You will need to create a new map class (similar to ObstacleMap) which contains a random grid of black and white tiles. The class should simulate whether the line sensor would return black or white given its position.

The particle filter then should use this measurement to weight and update the particles.

2. Grid Localization on black and white tiles

In this option, you will use the same idea as in Option 1, which is to localize based on a random map of black and white tiles. However, you will use the Grid Localization method to track the position of the robot. For the motion model, you can simply "blur" the map using cv2.blur() – this will spread out the belief in the robot's position at each timestep.

3. SLAM using a Rao-Blackwellized Particle Filter

In this option, you will combine the mapping and localization algorithms from previous homeworks to create a Simultaneous Localization and Mapping (SLAM) system using a Rao-Blackwellized Particle Filter (RBPF).

The RBPF should maintain a set of particles representing the location of the robot, as before. Each particle should have its own OccupancyGrid. Before the particle filter update, update each OccupancyGrid using the rangefinder reading and the particle's pose hypothesis. The particle weight should be computed as in

HW4, except the particle's OccupancyGrid is used to generated the expected rangefinder reading, after rounding the probability values in the grid so that they are either 0 (free) or 1 (occupied).

Another alternative is to do SLAM RBPF with the black-and-white tile map.