

Warmup Project progress:

- Familiarity with the ROS platform (topics, nodes, simulators, different commands for using ROS within terminal) and writing object-oriented code for ROS nodes
- Used a finite state machine to plan a wall following strategy/control flow
- Implementation of wall follower

Wall Following Strategy:

Wall Follower Code Architecture:

(relevant script: `comprobo2014/src/warmup_project/scripts/wall_follower.py`)

Learning Goals for the Warmup Project:

- Become familiar with ROS tools
 - resulted in tradeoff of time devoted to project: spending time on tutorials/poking around vs. implementing something complicated with the behaviors)
- Become familiar with neato platform (debugging, visualizing results in simulator or printing msg content, etc)
- Write code that was “flexible” for future work/well documented

Future iterations/project options...

- meanshift for determining the center of an obstacle to avoid/person to follow

Insights

(good things to do with comprobo projects; didn't necessarily do all of these in the warmup project but will remember them for all assignments/projects from now onwards)

- Object oriented programming. write classes for ROS nodes
- Finite state machine for figuring out and communicating decisions/thought process!
- Draw lots of pictures! geometry diagrams are very very helpful
- Start with naive implementation and list of assumptions, then make it more robust
 - Once you get to something testable in the real world, the real world results will make it very obvious what needs to be fixed/worked on next.
 - As you tackle each assumption, the implementation become more robust and work better in different situations.