# Warmup Project Write-Up

*The original one I wrote is Wall_Follow.py, and the one I wrote after that is Wall_Follow_etc.py. The second is what I did after the actual assignment because I felt could do a much better job after what I had learned. Anyhow, that's why that is there.*

## Behaviors

The implemented behaviors are Wall Follow and Obstacle Avoid. For Wall Follow, I …
Obstacle Avoid was made very simply; if the robot saw any data point within 0.5 meter, it would turn 180 degrees and go in the other direction. This is not, of course, a great way to implement this behavior but it was what I had bandwidth for at the time.

For a finite state controller, I had both Wall Follow and Obstacle Avoid running at the same time, but always dealing with different ranges of data. Wall Follow only interacts with numbers between 0.85 meters and above, while Obstacle Avoid interacts with data points within a half meter.

For the second iteration, I started, I attempted the same two behaviors but did them very differently. I wrote functions to detect clusters, so it would know what to avoid (or not).

## Structure

I structured my code very simply, keeping both behaviors in the same script. When I started on the second iteration it was much more complicated so I chose to organize the functions as below. I could have had two scripts, but I wanted to try to do it all in one function.

top section- functions which are not specific to either behavior
second section - functions which relate to Wall Follow
third section - functions which relate to Obstacle Avoid

## Reflection

Really the greatest challenge I faced along the way was my unfamiliarity with ROS. I have never used it before, so learning the syntax really made this project take a long time (read - about 15 hours) because I spent a lot of time doing many tutorials every time I encountered a problem. The most time intensive was figuring out how dictionaries work.

If I had more time I would have changed a lot of how my code was written. It is far too simple to do intelligent maneuvers. I realized I should have done it really differently once I reached the end, which is why I started doing it again for practice. I included that file so you can see what I decided to change, if you are curious. When I rewrote it the biggest difference was deciding to have the robot able to recognise what were walls and obstacles, instead of assuming any points near it were viable.