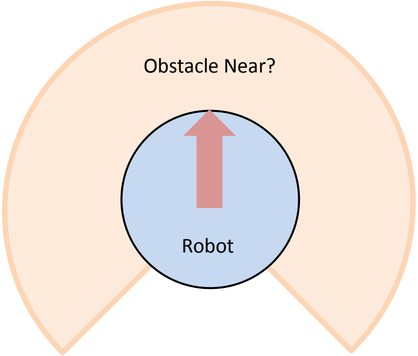
Finite State Machines for Behavior Coordination
Programming Exercise #3
February 8, 2017

This is what we will work on in class. If you are trying this on your own, after you complete step 3 you should be caught up with the class. If you run into trouble please email Dr. Archibald (ca861@msstate.edu). These instructions assume that you have completed the in-lecture coding from before and have the necessary `ai_lectures` package setup.

1. First, we will get the code in the correct place to use for this lab
   - Open a new terminal in Ubuntu and ROS-change directory (`roscd`) to `ai_lectures`
     - `roscd ai_lectures`
   - Unzip programming exercise (PE3) files from blackboard into `ai_lectures` folder
   - Then copy the python script into the `scripts` folder by running
     - `mv *.py scripts/`
   - We need to make the script executable, so now type (and don't forget the *)
     - `chmod +x scripts/*`
2. Now we will test that this is all working. Run the following command from the terminal. You should see the simulator window pop up and a robot (circle shape) sitting there.
   1. `roslaunch ai_lectures bouncer.launch`
3. Now you can modify the code in `bouncer.py` to complete the rest of the exercise.
   A. First, ***design*** a Finite State Machine to coordinate behaviors so that the robot drives forward until it senses a wall, then rotates in place until the wall is not in its view, and then drives forward again (in essence driving around bouncing off the walls). You have access to a single sensor variable, which tells whether or not something is close to the robot (as shown below). Do this design on paper.



   B. Implement your design and make sure it works
   C. Rerun and play with parameters until your robot bounces in the way you would like
   D. Change something about one or more of your behaviors to change the overall behavior of the robot.

- Tips:
  - To drive the robot forward/backward, set `twist.linear.x` to +/- values.
  - To spin the robot, set `twist.angular.z` to +/- values
  - If you have trouble getting ROS to `roscd` to your directory, try running the following command from the `catkin_ws` directory
    - `source devel/setup.bash`
    - Add the line "`source <path to your catkin>/catkin_ws/devel/setup.bash`" to your `.bashrc` file to have this happen automatically when you log into a terminal. (make sure that `<path to your catkin>` is replaced by the path on your system to the `catkin_ws` folder. You can get this path by typing `pwd` from the terminal when you are in the directory.