LAB 3: EXTENDED BEHAVIORS

Due: Thursday, February 16th 4:30pm

The objective of this lab is to establish a strong foundation for the robot's motion and behavior systems before we begin adding planning and localization. Keep in mind that you will continue to build on this code over the course of the semester, so try to follow good coding practices. The lab consists of three parts:

Continuous driving [30 points]: Implement continuous driving using Cozmo's `drive_wheels` command. We saw a wide range of solutions to Lab 2, majority of which did not use continuous driving. In order to enable the next component of the lab, we need the robot to be as reactive as possible and have the ability to turn and drive at the same time in order to keep a moving ball in view. See excerpt from the Mobile Robots textbook posted on Piazza, as well as lecture slides from Feb 6th for reference.

Pursuing moving target [35 points]: Extend your `goto_ball` solution to pursue a ball that is rolling across the arena. Your robot should have the ability to reliably track a ball rolling at medium speed approximately 12" from the camera. If the ball rolls out of the robot's view, the robot's search behavior should appropriately take into account in which direction the ball was last seen in order to maximize the chance of finding it.

Finite State Machine [35 points]: Adapt your code to encode all behaviors as a finite state machine. You may adapt an existing FSM library of your choice, such as this one, or implement your own version. Either way, the FSM must be a stand-alone class, not just a series of if-statements embedded within your code. Your implementation must use a minimum of three states, although you may want more. For demonstration purposes please encode the following features:

- The robot's screen should display the current state (as a unique symbol, word, picture, letter, number, etc.). In other words, by looking at the screen you should be able to tell which state the robot is in.
- Each FSM transition should result in i) a print statement that lists the old and new state on the terminal, and ii) auditory feedback in the form of a short sound played by Cozmo or your laptop.

Evaluation: The following grading rubric will be used to grade the assignment:

30 points	The robot successfully uses `drive_wheels` to continuously adjust its distance and angle to the ball.
20 points	The robot tracks a moving ball and is able to catch up to a slowly rolling ball
10 points	The robot searches in the correct direction when a rolling ball passes out of view
35 points	The behavior implementation correctly utilizes a finite state machine. Points will be deducted if the robot frequently oscillates between states or gets stuck in an loop.

You will demo your code for grading during class on the day the assignment is due.

Submission: Submit a zip file containing all of your code, with the file name: `Last1First1_Last2First2.zip` (the first and last names of partner 1 and 2, respectively). Also make sure you enter the names of both partners in a comment at the top of the file. Only one partner should upload the file to T-Square. If you relied significantly on any external resources to complete the lab, please reference these in the submission comments.