Milestone 5

Please note: this has been split into two deliverables. The majority of it is due on the evening of May 14th; however, the "Experimental Evaluation" section of the writeup can be incomplete at this point (but does not have to be). The completed version is due the evening of the 17th. See Step 2, below.

This milestone is the final version! You will turn in your final code base for your robot, a *descriptive* video, and a complete, research-paper style writeup. Your goals are the same as Milestone 4: Have a robot that elegantly handles maze navigation in a way that is robust to error, makes full use of sensor information, has a good strategy for robotic maze solving, and incorporates what you've learned in class. You should not change your core approach.

Although this turnin gives you an additional week and a half to do big hunting and minor improvements, your basic strategy should be nailed down and implemented at this point. The biggest block of your time should be spent on the writeup.

Step 1: Finish Robot Navigation

The code you turn in at this stage is what will be running on your robot at the final maze-running demo. There are no specific additional guidelines; at this point you know what you want your robot to do. However, here is some ***advice***:

- **Concentrate on basics.** Because this is the final turnin, before experimenting, make sure the baseline is solid, because your actual strategy depends on the core working. Can your robot:
  - Move forward in a straight line? Backwards?
  - Move forward a fixed amount (say, 28cm)? Backwards?
  - Turn 90° either direction?
  - Reliably tell how far it is from side walls?
  - From the end of the corridor ahead?
  - Center itself between two walls?
  - Turn a corner? Without bumping anything?
  - Notice an opening in a side wall?
  - Navigate through an opening?
- When that's working reliably, **practice working through simple mazes**.
  - Can your robot reliably—say, 5 times in a row—navigate through the four example mazes given, or some similar variant?
  - If you simulate noise (for example, nudging the robot to one side while it's moving, or briefly passing a hand in front of the sensor), what happens? (No, I won't do that. You're just testing out noisy cases.)
- **Don't worry too much** about weird cases.
  - Most of the mazes will be pretty simple. You're getting graded on your strategy, use of everything we've learned in class, and general robotic robustness.
  - You aren't getting graded on your ability to figure out what curveballs we might throw at you; the more complex mazes are mostly for fun.
  - It's not a good use of your time to worry about it. It *is* a good use of your time to be able to handle basic mazes.
- **Practice good software engineering**.

- Even though (actually, especially when) it's close to a deadline, it is a really good idea to break what you need to do down into functions, write unit tests, and pay attention to architecture.
- Meet with your group consistently and when you have agreed to. Use email. Make sure everyone knows who is doing what pieces. This ***will*** affect your grade.

Step 2: Write Up Work

The final iteration of your paper is a project report. This will probably be on the order of 8-10 pages; it shouldn't be shorter than 6 or longer than 12 unless you have a very specific reason.

- The majority of what you write will probably be in the Approach and Evaluation sections. Very broadly, the first contains the description of strategies, plans, architecture, and decisions you made, and the second contains the actual results of trying to make it work in the real world.
- Feel free to include pictures and figures such as architecture diagrams. In fact, please do—it generally makes your paper a lot more readable and informative. Always use meaningful captions.

Your paper should contain (at least) the following sections:

1. **Introduction:** Describe the project (very briefly), then a high level, conceptual overview of your approach and your experimental results (that is, how well it worked out). This is typically about a page.
2. **Related work:** This should contain a short (1–2 sentence) description of sources you accessed, including any libraries, github projects, our textbook, useful web pages, videos you watched, etc. Please note that this is not just a list—it is written text, and reading it should give me an idea what these sources are and what they contributed to this project. The actual links or sources should be cited in this section, and formatted as references in the bibliography. (See example below.) Don't forget to include libraries like easygopigo.
3. **Approach:** This is the section where you tell us all about everything interesting you did and all the design choices you made, as well as your thought processes and other considerations. This should build on all the points from Milestone 4. What did you end up doing? What decisions did you make and why? What other approaches did you consider, and why did you do them? (If you tried them and it didn't work in practice, that goes in the next section.)
4. **Experimental evaluation/Results:** A description of how you're testing the robot and what happened. Did you build a maze? (Include pictures!) What did you have to do to make it work? Did you calibrate the robot? How? If you elect to include the final demos in your writeup, how were the mazes set up?

   An explanation of what actually happened during testing. Did the robot navigate successfully? If not, what happened? Did you try something and have to switch approaches because it didn't work when you tested it?

   This is a good place to expound on the real-world difficulties and decisions you made. Did your robot consistently bump into walls? In practice, did your strategy perform poorly in a maze where all the openings were on the left? What went well?

   NOTE: *If you wish,* you may turn in an updated version of your paper with a results section that talks about how the robot did in the actual maze demos. This is intended to give you the opportunity to describe your results in the actual final "working" conditions, as well as in your own testbed.
5. **References:** These should be in a separate section at the end. Any standard citation format (APA, MLA, or IEEE) is fine; just be consistent. (679 students should use the template format.) For specific guidance, see: `http://tiny.cc/ieee-citations`

Example related work:

> For this project, we explored several different ways of implementing SLAM [1]. We initially intended to use FuzzySLAM [2], which is designed to handle mapping in a maze, but it turned out to be unsuitable because it made our robot grow actual fuzz, which interfered with the servos (see Figure 3). We also considered using an existing particle filter-based localization implementation [3], but decided to build our own implementation of mapping when we remembered the environment is just a bunch of squares.

Example bibliography:

[1] Dissanayake, M. G., Newman, P., Clark, S., Durrant-Whyte, H. F., & Csorba, M. A solution to the simultaneous localization and map building (SLAM) problem. *IEEE Transactions on Robotics and Automation*, 17(3), 229-241, 2001.

[2] Matuszek, C. and Smith, T. FuzzySLAM: Using Fuzzy Logic on Completely Inappropriate Problems. *Journal of Improbability Distributions in Robotics*, 2011.

[3] Smith, J. J. J, Particle Filters on A Tiny Robot. GitHub repository, https://github.com/smithjjj/ros_gopigo3_particles. [Retrieved May 1, 2018].

Turnins for Milestone 5

- Your final, complete code, in a form that can be put on our robot and tested, with a short, useful README ***text*** file describing how to run it.
- A video of your robot navigating a more complex maze. This should be less than five minutes and doesn't need to show the entire solving process. If there is informative console or audio output, include it.
- A PDF writeup with the sections described above.