Project 1: Drones
Computer Science 4013/5013: Artificial Intelligence
Project groups: Due 8:00AM February 5, 2018
Project: Due 11:59PM February 12, 2018

Introduction

This project is intended to mirror the Project 1 for SpaceSettlers, with the same learning objectives focused on knowledge representations and simple reflex agents and model-based reflex agents.

For this project, you have been provided a Parrot Mambo FPV drone. Your first step is to download (either clone or fork) the pyparrot library from GitHub.

https://github.com/amymcgovern/pyparrot

You will use this code throughout the projects. Since this code is in alpha mode, a fork is probably your best answer as it will allow you to request changes through pull requests on GitHub. However, please make sure your forks are private so that only your group members can see your repository!

Your task for this project is:

- Download the pyparrot library and ensure you can run the code by following the directions in the documentation (if there are documentation errors, please let me know).
- Run the demos (at least demoMamboTricks.py) to ensure you can control your mambo.
- Ensure that you only fly your mambo INDOORS! This drone is not designed for outdoor flight and you are not allowed to fly drones outdoors on OU's campus.
- Create a simple reflex agent that will fly your mambo from 0, 0, 0 (where it starts) to a specified (x, y, z) coordinate. Ensure that your user can enter the coordinates and assume they are in meters. Note that the simple reflex agent cannot access a model of the state, only instantaneous state variables. It will probably stink at getting to the location but you can make a good guess with math!
- Using the sensors available to the agent, build an effective knowledge representation so that the drone can estimate its x, y, z, and orientation.

- Build a model-based reflex agent that uses the sensors and will fly your mambo from 0, 0, 0 (where it starts) to a specified (x, y, z) coordinate. Ensure that your user can enter the coordinates and assume they are in meters. This likely will perform better than your first agent.
- Note: since we are not yet using vision, please ensure that your agent doesn't run into any obstacles and especially not at high speed! Also note that your accuracy will be worse at high speeds! And crashes will be much more costly!

Extra-credit opportunities

To keep grades within a fair range, all grades are capped at 110.

Wanted dead or alive: bugs or new features

The pyparrot library is brand new and likely has some bugs to work out. We want to know about them and fix them! If you find a bug, you can receive extra credit according to the following scale:

- **1 or 2 points:** General bugs are likely given how new this code is. Finding a bug and reporting it can get you 1 point. Fixing the bug and giving us the fix (you can't check it in directly but you can give it to us in the bug report) can get you two points. Both bug and fix must be verified for any extra credit to be awarded. Bug reports should be verbose and state exactly what happened. Simply stating "it broke" will not count. Feel free to report them on GitHub using issues.
- **3-10 points:** If you a particularly clever way to implement a feature that is not currently implemented, please submit a pull request with your code. If I accept it, you will gain extra credit worth 3-10 points, depending on the feature and the code you submit (one line quick fixes are worth less than more complicated features).
- **10+ points:** Anyone who can get opencv to work reliably across platforms will be on my list for extra credit and lots of good karma too! It needs to read RTSP and RTP streams and it needs to work on Linux, Mac, and Windows. Our current solution involves the VLC library but I would prefer to have opencv working directly.

Wanted (alive please): creative individuals

Creativity is highly encouraged! To make this real, there are up to 10 points of extra-credit available for creative solutions. This assignment is ripe for creativity! If you choose to implement anything that you consider creative, please do the following:

- Document it in your writeup! I can't give extra credit unless I know you did something extra.
- Remember that by being creative I am referring to the algorithm and *not* to the ability to creatively download code. **All project code must be written exclusively by your group except for the sample players that we provide.**

Those pesky details

1. Before Feb 5 at 8am (when I open the competition ladders for the other projects): put yourself and your partner into a group with the team name that you chose in canvas under "Project Groups".
2. Ensure you have checked out the pyparrot library from:

   https://github.com/amymcgovern/pyparrot
3. Create a simple reflex agent that will fly your mambo from 0, 0, 0 (where it starts) to a specified (x, y, z) coordinate. Ensure that your user can enter the coordinates and assume they are in meters. Note that the simple reflex agent cannot access a model of the state, only instantaneous state variables. It will probably stink at getting to the location but you can make a good guess with math!
4. Using the sensors available to the agent, build an effective knowledge representation so that the drone can estimate its x, y, z, and orientation.
5. Build a model-based reflex agent that uses the sensors and will fly your mambo to 0, 0, 0 (where it starts) to a specified (x, y, z) coordinate. Ensure that your user can enter the coordinates and assume they are in meters. This likely will perform better than your first agent.
6. ONE member of your team should submit your final code to canvas, along with your writeup.
7. Bring your drone to class on the day listed on the schedule for project 1 demos. Be prepared to demo!

Point distribution

- Reflex agents
  - 25 points for designing and correctly implementing a reflex agent that correctly moves to the specified location (it doesn't have to be super accurate but try your best). It must also make effective use of your knowledge representation. Full credit requires FULL code documentation explaining what each function does. The top of each python file should also contain a 2-3 sentence description of everything that python file does.
  - 20 points for one minor mistake. An example of a minor mistake would be an off by one error or incorrectly choosing your direction or choosing to buy something when you had no money. You can also lose 5 points for not documenting your code well (but at least somewhat).
  - 15 points if there are several minor mistakes or if your code is missing a lot of documentation.
  - 10 points if there is a major mistake. An example of a major mistake will be causing your ship to die by sending it into an asteroid.
  - 8 points if you can move around the world in a manner more intelligent than random and it is not a copy of one of the heuristic agents already provided to you.
  - 5 points for an agent that at least does something other than random movements.
- Knowledge representations
  - 25 points for correctly implementing a high level knowledge representation. this representation will store information the current estimated location and orientation and anything else you think you need to know for the mambo to do its job for this project. The top of each python file should also contain a 2-3 sentence description of everything that python file does.
  - 20 points if there is only one minor mistake. An example of a minor mistake in knowledge representation would be an off-by-one error in calculations. You can also lose 5 points for not documenting your code well (but at least somewhat).
  - 15 points if there are several minor mistakes or if your code is missing a lot of documentation.

  - 10 points if you have one major mistake. An example of a major mistake in knowledge representation would be incorrectly storing information for a team other than yourself and not storing your own information (e.g. making choices on the wrong team's data).
  - 5 points if you do not make an official knowledge representation but still somehow use knowledge to intelligently move around the world.

- Model-based reflex agent

  - 20 points for correctly implementing a model-based reflex agent effectively uses your knowledge representation to move to the specified locations. Full accuracy is not required but get as close as you can given the 2Hz update rate. Full credit requires FULL code documentation explaining what each function does. The top of each python file should also contain a 2-3 sentence description of everything that python file does.
  - 15 points for a minor mistake (see above)
  - 10 points for several minor mistakes
  - 5 points for a major mistake (see above)

- Code: We will randomly choose from one of the following good coding practices to grade for these points. Note that this will be included on every project. Are your files well commented? Are your variable names descriptive (or are they all i, j, and k)? Do you make good use of classes and methods or is the entire project in one big flat file? This will be graded as follows:

  - 10 points for well commented code, descriptive variables names or making good use of classes and methods
  - 5 points if you have partially commented code, semi-descriptive variable names, or partial use of classes and methods
  - 0 points if you have no comments in your code, variables are obscurely named, or all your code is in a single flat method

- Writeup:

  - 5 points: Describe your knowledge representation in detail
  - 5 points: Describe your reflex and model-based agent that you created

  - 3 points: Why you think this knowledge representation and reflex/model-based reflex was a good choice.
  - 5 points: Briefly describe how you worked together as a group. In the case of groups, full points requires full collaboration and not just splitting the project into parts.
  - **Your writeup is limited to 2 pages maximum. Any writeup over 2 pages will be automatically given a 0.**
  - **Turn your writeup AND your code in to canvas**.

- Live Demo: You can have up to 5 points of extra credit for a live demo of your best agent in class.