# Project 3
# CS4013/5013: Artificial Intelligence
# Due date: 11:59pm April 4, 2018

## Introduction

This project focuses on machine learning and getting you real experience at creating and machine learning methods. The goal of this project is to have you create a forward model for your mambo, meaning you build a machine learning model to significantly improve your predictions of where you are in space (x, y, and z).

As we discussed in class, all projects should collect data and share the data across projects so that you minimize individual data collection. You should collect data on every sensor available and report your reported landing position as well as takeoff and the commands that you sent.

All groups should implement decision/regression trees. Graduate students should implement a second ML method Note that regression using the code I handed out in class does not count as a second method but it could certainly count as a comparison method for EC.

Note, as with the spacesettlers projects, if you want to implement a ML method that is not something we discussed in class, you need to talk to me first to get it approved! I want to ensure you keep your approaches within a feasible range. For example, I will not approve any deep learning based projects as they are simply too hard within the time span of the project.

## Extra-credit opportunities

To keep grades within a fair range, all extra credit will be capped at 10 points. This means that no project can receive a grade higher than 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).

## 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.

- Document it in your writeup! I can't give extra credit unless I know you did something extra.
- You must still be doing learning (speak to me if you want to verify that your approach is still learning).
- 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 Visual SLAM, which I am explicitly giving you permission to download. You are also welcome to use any functions in the opencv libraries.**

# Those pesky details

1. Update your code from project 1. You can update your code at the command line with "git pull". I have released fixes for the codebase since project 1.
2. All students: Create a drone program that improves the estimation of position using machine learning as described above.
3. CS 5013 students: Create a second machine learning approach to position estimation.

4. ONE member of your team should submit your final code to canvas, along with your writeup.
5. Bring your drone to class on the day listed on the schedule for project 3 demos. Be prepared to demo for EC!

## Grading rubric

The points for trees are broken out below. NOTE: CS 5013 students must implement a second learning method. To be clear: undergraduate groups will be graded out of 100 points. If they choose to implement a second learning method, it would count towards creativity and extra credit. Graduate groups will be graded out of 130 points. Groups with both undergraduate and graduate student will be graded separately. The undergraduate will be graded out of 100 points and the graduate student will be graded out of 130 points. They will share the points for decision trees. Graduate grades will then be re-scaled to be 100%.

- Decision trees
  - 30 points for properly implementing a regression or decision tree learning algorithm that improves the estimation of your position.
  - 25 points if there is only minor mistake.
  - 20 points if there are several minor mistakes or one major mistake. This also includes NOT including a graph of the learning in your writeup.
  - 15 points for several major mistakes
- Scoring the splits for decision trees
  - 10 points for implementing a working score for choosing the best split for either decision trees (entropy or gini) or regression trees
  - 5 points for errors in the splitting criteria
  - 0 points for not sharing data
- Using the tree
  - 20 points for using your trained decision tree to improve your position estimation and demonstrating the improvement in a graph (see the writeup for the description but the points go here)

  - 15 points for errors in your usage of the tree
  - 10 points for using the tree correctly but not demonstrating using a graph or for multiple errors

- Data collection
  - 10 points for properly collecting data and sharing it with all of the drone groups in sufficient time that everyone can use it for learning
  - 5 points for sharing late or improperly sharing data
  - 0 points for not sharing data

- Second learning method: CS 5013 students only
  - 30 points for CS 5013 students only: for implementing a second learning method correctly and in a way that is useful for your agent AND demonstrating the learning in the writeup
  - 25 points if there is only minor mistake.
  - 20 points if there are several minor mistakes or one major mistake. This includes a state representation that crashes some of the time or that simply is not useful for learning at all. This also includes NOT including a graph of the learning in your writeup.
  - 15 points for several major mistakes

- Coding practice: We will randomly choose from one of the following good coding practices to grade for these 10 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
  - 7 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 (not sure you can do that with A$^*$ anyway!)

- Writeup as described below (totals 20 points)
    - 1 point for the name and class level (CS 4013 or CS 5013) for each student in your group at the top of your writeup document
    - 2 points for describing your method of collecting data (how you measured, what you measured, where you collected data, etc)
    - 4 points for describing your choice of learning representation (which features you chose) and why you chose that representation. This applies to both learning methods if you are in CS 5013.
    - 4 points for describing your choice of learning parameters. This applies to both learning methods if you are in CS 5013.
    - 5 points for a proper learning curve. To get all of the points, the axes must be properly labeled including units.
    - 4 points if your learning curve demonstrates actual learning (meaning performance improves over time) and you discuss why you think learning worked. 3 points if you can explain why you think it is not learning. Note, additional curves measuring different forms of performance are welcome.
    - CS 5013 students MUST have learning curves for both of their learning methods in their writeup for full credit. Points are deducted under the graduate 2nd learning method for missing the 2nd learning curve.
    - **Your writeup is limited to 2 pages maximum. Any writeup over 2 pages will be automatically given a 0.**