# Problem Set 4

## Download the pset files in a .zip

Download here a .zip file with the python files that you will need. (pset4/pset_4.zip)

## Introduction

There are three types of files in this lab.

- Some `.py` file(s).

  These files contain the implementation of the planners we will be considering. You will be completing/modifying certain parts of these files.

- A test script with a name like this: `test_pset_4.py`.

  This is the same test script on Gradescope that determines how many points you score. The test script is provided to you so that you can run the tests locally and quickly get feedback/confidence on how well you do. **Do not modify the test script**.

- Model/resource files.

  `.pddl` files are used to encode symbolic planning problems.

## How to do this Assignment

This assignment is a little different from the previous psets and lab. It doesn't use Drake or jupyter notebooks, so most of your testing and development doesn't need to run in a docker container (although it all can if you wish). Also unlike previous assignments, in addition to the code to submit for passing the Gradescope tests, part 2 asks you to describe your development process in the comments.

The assignment has two parts. In part 1 you will complete the implementation of an RRT and a biRRT planner; in part 2 you will complete the heuristics for a STRIPS planner. The zip file defines two directories `rrt_planner` and `pddl_planner`. See the following two pages for a more detailed description of each part:

- Part 1: Motion Planning (pset4/rrt_planner.html)
- Part 2: Tasking Planning Heuristics (pset4/pddl_planner.html)

The only time you need docker is to run the Gradescope tests. Run `test_pset_4.py` in a docker bash container. For example, if you are on Mac, run

```
docker/docker_run_bash_mac.sh drake-20181030 pset_4/
python test_pset_4.py results.json
```

## How to submit this Problem Set

We will use Gradescope (https://www.gradescope.com) for collecting and grading your problem sets, since it allows us to auto grade your code against our software tests, as well as look at your code and other text answers. If you did not already do this, make an account on Gradescope, and add 6.881 with the class code emailed to you. Although all course content is open, we only do grading for officially enrolled students who are taking this class for credit. Please contact the class staff if you are taking the class for credit but have not received the access code.

Please note that you need to make **only one** submission for the problem set.

This submissions is:

1. **Autograded submission**
   - Upload your `rrt_planning.py` and `pddl_main.py` to "Pset 4" on Gradescope.
   - In our testing, Gradescope will give you a grade within 2 minutes. You can resubmit as many times as you'd like before the deadline.
   - Note that for your RRT and biRRT implementations, we are testing your random algorithm. This means it might not always converge. Each test runs your implementation 5 times and only requires a solution to be found once to pass. However if your implementation fails the Gradescope tests, you can resubmit the same code to re-run the tests.
2. **Manual submission**
   - Make sure your `pddl_main.py` contains comments describing your testing strategy. You **do not** have to upload this file separately from your autograded submission.

Good luck and have fun!