

Project 1, 2018

Deadline: Thursday 30 August 19:00

This project counts towards 10% of the marks for this subject.
This project must be done individually.

Aims

The aims of this project are to improve your understanding of the various search algorithms and to experience how to derive heuristics, using the Berkely Pac Man framework.



http://inst.eecs.berkeley.edu/~cs188/pacman/project_overview.html

Your task

Your task is to complete the assignment at <http://inst.eecs.berkeley.edu/~cs188/pacman/search.html>. You can download the zip with all the necessary files to complete the project, the description of the task is contained within that.

NOTE: You should not change any files other than `search.py` and `searchAgents.py`. You should not import any additional libraries into your code. This risks being incompatible with our marking scripts.

Marking criteria

We will follow the marking criteria specified in the instructions <http://inst.eecs.berkeley.edu/~cs188/pacman/search.html>. There are a total of 26 marks available in that marking criteria, including one bonus mark. For the 25 'standard' marks, we will take a normalised score out of 10, rounded up to the nearest half mark. The bonus mark will be worth 0.5% of your final subject mark.

Submission

You must submit solution electronically as two different files as a zip file called `assignment-1.zip` containing two files: `search.py` and `searchAgents.py`. Your zip file must contain only these two files, must not be within another folder, and it must be named `assignment-1.zip`. Do not submit a tar or rar file. Breaking these instructions breaks out marking scripts, delays marks being returned, and more importantly, gives us a headache.

Note: Submissions that fail to follow the above will be penalised.

Submit your zip file using the following command:

```
submit COMP90054 1 assignment-1.zip
```

Please note the following:

- **Only include the `search.py` and `searchAgents.py` files in your zip file.** Do not modify the filenames, otherwise our scripts will fail to execute your code.
- You will need to login to the server `dimefox.eng.unimelb.edu.au` using a secure shell connection using port 22, using your university login credentials. Once logged in, copy your files over and use the ‘submit’ command above.
- The `dimefox` server is only available either on-campus machines or using a VPN connection (instructions on unimelb website for how to setup VPN <http://studentit.unimelb.edu.au/findconnect/vpn>).
- Note that the script for project 1 will not run your files, so using ‘verification’ will not work. It will merely provide you with some strange output.
- Test that your code runs on dimefox using the autograder.

Academic Misconduct

The University misconduct policy¹ applies. Students are encouraged to discuss the assignment topics, but all submitted work must represent the individual’s understanding of the topic.

The subject staff take academic misconduct seriously. In the past, we have prosecuted several students that have breached the university policy. Often this results in receiving 0 marks for the assessment, and in some cases, has resulted in failure of the subject.

Important: As part of marking, we run all submissions via a code similarity comparison tool. These tools are quite sophisticated and are not easily fooled by attempts to make code look different. In short, if you copy code from classmates *or from online sources*, you risk facing academic misconduct charges.

But more importantly, the point of this assignment is to have you work through a series of foundational search algorithms. Successfully completing this assignment will make the rest of the subject, including other assessment, much smoother for you. If you cannot work out solutions for this assignment, submitting another person’s code will not help in the long run.

¹See <https://academichonesty.unimelb.edu.au/policy.html>