UDACITY

‹ Return to "Intro to Self-Driving Cars" in the classroom

Implement Route Planner

REVIEW

CODE REVIEW 5

HISTORY

Meets Specifications

Correctness

✓

Running test.py shows "all tests pass".

Good job, your code passed all tests

✓

The student implements all required methods.

Congrats! you have implemented the classic A* search algorithm, please take a look at the code review section for some suggestions to improve your code for future implementations.

✓

The heuristic function used to estimate the distance between two intersections is guaranteed to return a distance which is less than or equal to the true path length between the intersections.

Well done, the euclidean distance is an admissible heuristic.

✓

Student answered all question correctly.

Good, please note when thinking about the differences of the search algorithms, you may think in the way they guide its search in the case of the UCS it uses only the cost of the path, in the case of BFS it uses only the heuristic, and in the case of A* it uses both.

Choice and Usage of Data Structures

✓

Code avoids obvious inappropriate use of lists and takes advantage of the performance improvement afforded by sets / dictionaries where appropriate. For example, a data structure like the "open_set" on which membership checks are frequently performed (e.g. `if node in open_set`) should not be a list.

Nice use of sets and dictionaries which gives a good performance to the algorithm.
Please visit this resource to know more https://wiki.python.org/moin/TimeComplexity

✓

This item is a judgement call. Student code doesn't need to be perfect but it should avoid big performance degrading issues like...

...unnecessary duplication of lists
...looping through a large set or dictionary when a single constant-time lookup is possible

Please keep up the good work!

DOWNLOAD PROJECT

5 CODE REVIEW COMMENTS ›

RETURN TO PATH

Rate this review
☆☆☆☆☆