SIT215 Assignment 2: Map Routing for Wheelchairs

Before you start

Due: April 29 2023, 8:00 pm (AEST)

This assignment contributes 20% to your final mark. Read the assignment instructions carefully.

Work in teams

In this assignment, you are encouraged to work in teams and to learn from your peers. You may form a "group" with up to 5 students. However, each of you needs to make a **separate submission** documenting your individual effort and learning, which will be **individually marked**. Highly similar reports will be investigated to ensure no plagiarism occurs.

What to submit

By the due date, you are required to submit the following files to the corresponding Assignment (Dropbox) in CloudDeakin:

- A brief report explaining the approach taken, any challenges faced during implementation, and any improvements or extensions made to the project. You may structure your report in the following sections:
  - Introduction and problem description
  - Approach and implementation
  - Results and analysis
  - Conclusions and lessons learned
  - Acknowledgement of any external assistance, if applicable
  - References
- A code file of your implementation `[YourID]_assignment2_solution.ipynp` or `[YourID]_assignment2_solution.py`. If you submit an IPython notebook, you also need to submit the HTML export, named `[YourID]_assingment2_output.html`.
- Extra files to complete your assignment, if any (e.g., input files used to reproduce your answers).

Please keep your report short and to the point. Clean up your code outputs to reduce unnecessary information (e.g., excessively long logs).

Marking rubrics

Indicative weights of various tasks are provided below, but your submission will be marked by the following criteria, adjusting for the overall quality.

P-level expectation

- Showing reasonable effort through completed tasks.
- Applying unit learning to design suitable solutions for the tasks.

C-level expectation

- Showing attention to detail through a good quality assignment report.

D-level expectation

- Demonstrating creativity and resourcefulness in providing unique solutions or contributions to a joint solution.
- Critically evaluating and reflecting on the pros and cons of various design decisions.

HD-level expectation

- Extending classroom learning to research and tackle previously unexplored theoretical questions or novel applications

(Warning: Highly similar solutions will be investigated for collusion.)

Your Task

In this assignment, you will implement a map navigation solution for wheelchair users.

Many people use a navigation program like Google Maps. However, no program currently supports planning a route specifically for wheelchair users. While Google Maps provides information on the accessibility of certain locations, it does not consider factors such as kerb ramps, sidewalk width, slope, and indoor lifts.

Objective: The objective of this assignment is to implement the A* algorithm for wheelchair map navigation.

Prepare a wheelchair-friendly map (P/C-level Task)

Create a map of wheelchair-accessible routes and locations in your local area, for example, your campus or your local shopping centre. You can use tools such as Google Maps or OpenStreetMap to mark wheelchair-friendly routes and points of interest such as accessible restrooms, car parks, and kerb ramps. The map may include information on the slope of sidewalks, the presence of obstacles or barriers, and any other relevant accessibility features. You may conduct an onsite inspection, use your own experience and online resources, or collaborate with local disability organisations to gather information for the map.

You should discuss the process of creating it.

Implement map navigation (P/C-level Task)

Implement the A* algorithm for wheelchair map navigation. You may create a simplified environment, with an adjacency matrix representing the connections between locations and the cost/speed of travel.

Justify the heuristic you choose.

Test the implementation with different start and end points. Verify that the path returned is valid and optimal.

Compare heuristics (D-Level Task)

Try different heuristics. Explain the pros and cons of each heuristic.

Record and compare the search efficiciency of different heuristics.

Fine-tune the performance (HD-level Task)

Implement another algorithm, such as Contraction Hierarchies or Transit Node Routing, and compare its performance to your A* implementation.

Characterise the conditions under which the alternative algorithm is more efficient.

Graphical user interface (Optional Bonus Task)

Create a graphical user interface that allows the user to input the start and end points of the path and to visualise the path on the map. The user interface should also display additional information about the path, such as the estimated time to traverse it and any obstacles or landmarks that may be encountered along the way.