Project 1 Description and Submission Policy

In this project, you need to model the problem of finding the optimal way to get from one place on ZC campus to another place as a search problem.

Your tasks:

1. Define the abstraction of the problem using the state space representation, successor function, set of actions.
2. You need to mention any modeling assumptions you use such as for example, artificially created bumps to slow down cars, pedestrian crossings, waiting areas, sun shields, etc.
3. Find the best route using the following uninformed search algorithms

    - BFS
    - DFS
    - IDS

4. Reflect on the results of the three algorithms above.

5. Define TWO reasonable heuristic functions and justify your choice, then use them to find the best route using:

    - Greedy best-first
    - A*

6. Apply hill climbing (or any of its variants) and simulated annealing to find the best route.
7. Compare between the results of all used methods (time needed to find a solution, is it complete and optimal?)
8. For each case, you need to present a graph or search tree of your solutions.
9. Two-point Bonus part: show the actual path on top of ZC map using Google Directions API or Leaflet (https://leafletjs.com).

Submission Policy

Number of students per team:

- Up to 3-student groups are allowed.
- Only one team member from each group needs to fill out the form

Project Phases

Phase #	Project Phase	Deliverables	%	Due
Phase1	Requirements collection and modeling	- Team group names and emails (submitted at classroom using the announced form link) -a 3-min Demo for the initial design-video recoded	15%	Week 8, Friday 11:59 pm
Phase2	Final Report	Report	20%	Week 11, Friday 11:59 pm
Phase3	Final Demo	Presentation and a 5-min Demo, video-recorded	20%	Saturday of week 12, 11:59 pm
Phase4	Code submission	Commented and well-organized code	30%	Saturday of week 12, 11:59 pm.
Phase4	Individual Discussion		15%	Week 12, tutorial time

Code Grading Rubrics

- Code Functionality (50%)
  - The program shall be complete and includes all the points described in the project description file
  - The program shall perform the functionality described in the project description file
  - The program shall run without errors and without crashing
- Code Style (25%)
  - The code shall be organized into appropriate functions

    - The code shall not be repetitive, with effective use of helper methods
    - Variables shall be named appropriately, using appropriate scopes
    - The code must be readable with comments (following standard indentation)
- Code Comments (25%)
    - The program shall include detailed and precise comments
    - Your comments shall explain the "high-level" functionality/purpose of the code

Late policy

- 25% of the phase grade will be deducted for every late day by maximum 2 days. If a phase is submitted after the deadline by more than 2 days, no grade will be given.

Submission

- One team member should submit a compressed file (.zip, .rar, ….etc) containing the required files for the selected phase.
- The compressed file should be named as follows < Phase#_TeamName.zip>
- Submission should be done on Google classroom by only one member in the team.

- You are requested to deliver the following:

    1. Report: It should contain how work was distributed among the team, your introduction for the problem, how your program managed to solve it and how to use your program.
    2. Powerpoint presentation: a well-organized and concise ppt of your project with conclusions and challenges you faced. In your demo, you must adhere to the announced time limit.
    3. Project files: All project source files in addition to any other related files.

- you will be requested to have your individual discussion with your TA. Your evaluation in this part is based on whether your role in the project is clear, you communicated effectively about the project, provided cogent responses to questions, and defended the design choices that were made in the project.