



East West University

Department of Computer Science and Engineering

Semester: Fall 2023

Course Number: CSE366

Section: 04

Course Title: Artificial Intelligence

Assignment Number: Assignment-1

Assignment Title: Enhanced Dynamic Robot Movement Simulation

Submitted By,

MD. Ismail Bhuiyan

ID: 2021-2-60-063

Submitted To,

Dr. Mohammad Rifat Ahmmad Rashid

Assistant Professor

Department of Computer Science & Engineering

East West University

GitHub Repository Link:

<https://github.com/MdIsmailBhuiyan/CSE366-4--2021-2-60-063/tree/33bcdcae913cd960dc2ff23da329af4b76f8fe19/Assignment1>

Some important parts of the assignment

Class PriorityQueue:

It maintains a priority queue which is needed for both UCS search and A* search. Heap is used for its implementation.

Class Node:

Node Class represents a state in the search tree. It contains the current state and parent of that state. In this class there is a comparison operator for priority queue which will compare the path cost.

Class Environment:

The Environment Class represents the grid and handles state transitions. Its actions function will return possible actions that can be applied next from the state. Moreover, result changes the state with given action. Is_goal function is to check if the goal has been reached.

Class Agent:

This is the class where the algorithms are performed. The recharge_battery function is used for recharging the battery of the robot. When the battery level reaches a certain amount, the program needs to perform recharging. Here, the robot will lose 10% of its battery level for every move. **Uniform_cost_search** and **a_star_search** functions are used for performing Uniform Cost Search and A* Search respectively, to find the lowest cost path to reach the goal from the initial state. The **reconstruct_path** function is for backtracking from the goal to the start state.

Heuristic function:

It is needed for A* search. That function will return the Manhattan distance between points a and b.

visualize_grid_and_path function:

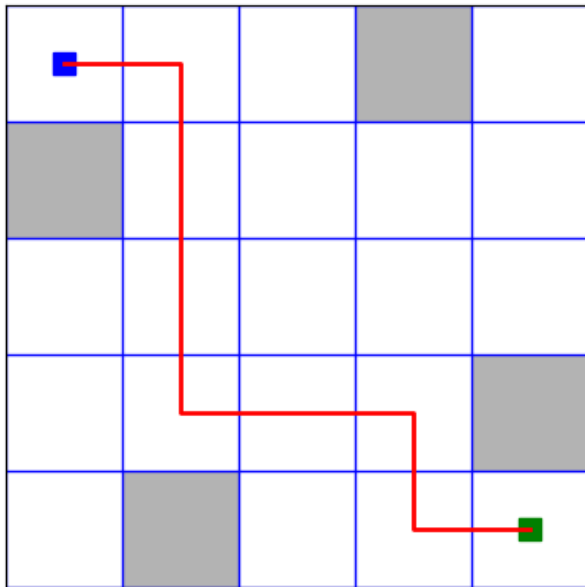
Visualization Function plots the grid and the found path. Here numpy is used to convert grid to numpy array for plotting easily.

Solution technique:

In this program we can define the grid or dynamically generate the grid for the environment creation. For the same environment, 2 agents are taken: one for Uniform cost search and another for A* search. So, here two pathfinding algorithms are implemented: Uniform Cost Search (UCS) and A* (A Star). The evaluation of these algorithms is based on the number of times the robot needs to recharge its battery while traversing the path to the goal. In the program, while traversing the path, the robot will lose 10% of its battery level. When the battery level reaches the lowest level, it needs to recharge again. The algorithm which needs less battery usage will be considered as the best algorithm for that particular environment.

Result:

The solution path and best algorithm will be shown in the output based on battery usage and number of battery recharge needed to reach the goal from start. A example of result is given bellow:



Issues:

In some environment can't reach the goal because the obstacles surrounded the goal. That type environment is given bellow:

