

Project -2 Writeup

Environment: We took the Co-operative mode which will focus on gathering resources, so that here we'll know what we are competing against in the environment.

Search (): This is the main method to search a path using Astar Search algorithm. Takes a start vertex and finds the path to the goal vertex and returns the path.

SearchByAstar (): Executes A star algorithm by exploring vertices until the goal is reached or the max number of steps is exceeded. We took max steps as 10 ,it replans if the steps are exceeded, or the target is moved.

CalculateHeuristic ():

- ☐ Calculates the distance between two vertices.
- ☐ $g(n)$ - shortest distance from the root to n .
 - Its calculated based on the shortest distance between two vertices (edge distance)
- ☐ $f(n)$ – shortest distance from the start to goal.
- ☐ The Heuristic is both admissible and consistent because it uses the shortest distance between vertices which will be optimistic cost to reach a goal and it doesn't overestimate the cost at any given vertex in the path.

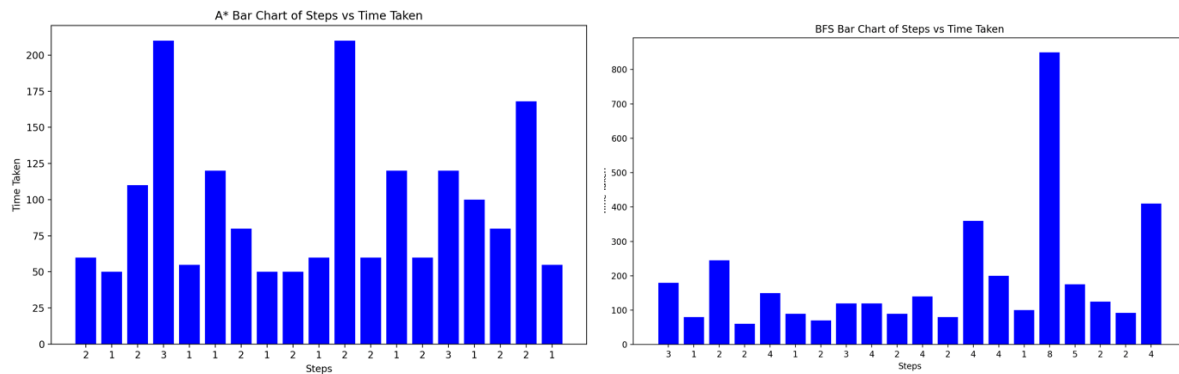
FringeContains (): Check if the given node with lower cost is already present in the fringe or not.

FindNode (): Recursively searches for a tree node from a given vertex to a specified goal vertex.

Choosing Target:

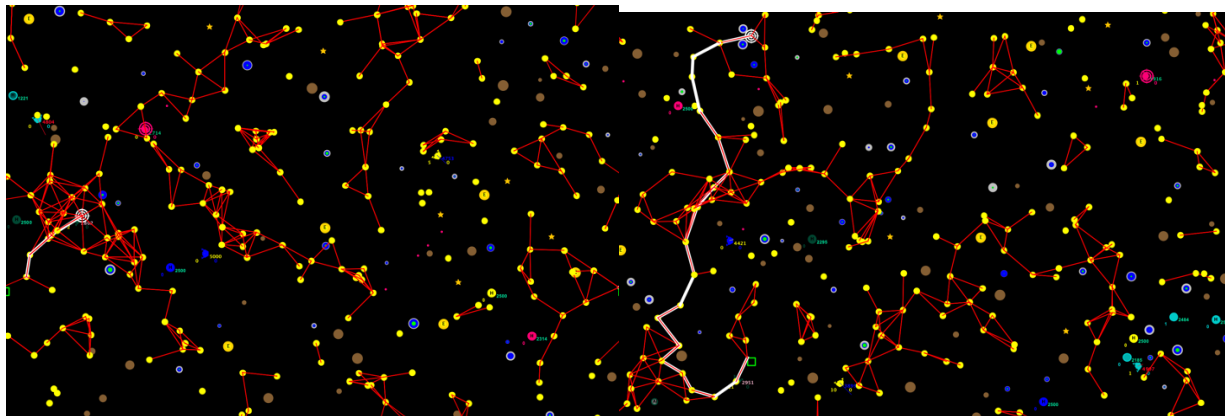
If the energy is less than 2000 for ship it goes to Energy beacon, and resources of the ship are more than 3000 then it'll go to base, and generally it utilizes the Astar algorithm to determine the optimum route to the nearest star or asteroid(based on whatever is nearby).

Bar graphs representing the number of steps taken to path and time:



1. In the above bar graph we took 20 examples where the ship reaches the goal vertex and note the timesteps required for the ship to reach the goal.
2. Here the Astar search is taking a smaller number of time steps when Compared to the BFS search algorithm.

Below the left one is the path assigned using the Astart search and the right side one is using the BFS.



1. The first diagram above shows the A* search algorithm path, and second diagram shows the BFS path where A* search performs better than BFS by taking a path with smaller distances and reaching the goal vertex.
2. Since the A* search uses Heuristics & priority queue, it does not explore any node that makes the ship travel in longest available paths unlike BFS path.