ASSESMENT 3 CAI104 REPORT

Overview

This project is aimed at studying how the agent designs the real-time AI system based on the use of search algorithm methods.This project is done with the help of C++ language in visual studio.In this project the goal is to draw a maze in such a way that the robot finds a way to reach the path destination.Using A* algorithm as a base search method the robot visualize the paths and makes sure it reaches to the target using a closest path possible ignoring the obstacles found on the way.A* search algorithm is used here because it is often used in many fields of computer science due to it's completeness, optimality and optimal efficiency. Also it uses cost path g(n) and a cost estimate h(n) to reach the goal.

$$I.e. f(n)=g(n)+h(n).$$

The maze is designed in 2d array such that there are 12 rows and 24 coloumns;288 blocks on the map.Since it is represented in 2D array ,'1' represents wall,'0' represent walkable space 'A' represents the robot and 'B' represents the target.Here,The agent uses a Global path planning that requires prior knowledge of the robot's environment and using this knowledge it creates a stimulated environment where the methods can plan a path.

The envrionment created is static where the environment is unvarying, the source and destination position are fixed, and obstacles do not vary location over time.

The robot adapts to real time changes in the environment including algorithms that stops a robot when it approaches an obstacle and takes in information from the surroundings and creates a plan to avoid obstacles.

Where it went right

After completion of the code and running the program there were many errors. After many trials and errors the program ran successfully achieving our main goal i.e creating a path to the target from a robot intital position.'*' star notation has been used to mark the shortest path obtained by the algorithm in the 2D array map. The path created by the robot seems to be the shortest path in the algorithm. Since it was programmed to find the only shortest route ,there were no secondary routes in the result which proved the completeness of A* algorithm. The program ran sucessfully showing all the details such as wall, walkable space, the robot and the target in 2D array. Every aspects were visaulized properly and shown so the user can understood too.

Most of the things were in right order. The node list (closed list and open list),the algorithm explored and evaluated the loacation and made a record of all location that already has been explored and took a path closest to the target. The surrounding nodes were also checked and the search were inside the rectangular map scope only ;inside 288 blocks. Depending on it's completeness ,the heuristic algorithm also did a good-quality solution in a shorter time peroid. The algorithm differentiated the location of the robot and the target so that the robot follows the target not vice versa.

The performance of vector in the program was optimal too. The vector was used in to futher add a node in the program when the array becomes full. It helps to add aa extra node each time the current position of a node doesn't break from the original parent source position.Also the movement of the robot were fluent as it could move upward,doward or diagonally.

What went wrong

Even the best- thought-out program can have unexpected issues. Some are easy to track down ;others can drive you crazy trying to figure out the issue. We might tempted to blame the robot as being faulty but that is not the case. The robot is doing exactly what it is told to do by the program.

The main error in the program is the robot passes through the wall which are present inside the mapArray scope.The (*) sign is used as a path created by the robot(A) to the target(B).Though the robot ultimately reaches the target and achieved it's goal the obstacles are being neglected. Maybe this was due to the wrong step in the program ,the robot seems to passes through it as a walkable space. In my conclusion the robot doesn't count the inner walls as obstacles but as a free space.

Also the program has other errors like console engine was too big.The colors representating the wall, target and the source was not used so it is little difficult to visualize.

What you are Not sure about

Though In theory we think to know about everything but after implementing in real life world we found ourselves in a dilema.A dilema of what will happen if?or which we are not sure ?. So following are the points which I didn't know or I am not sure about:

1.The first thing I was not sure about why did the robot passed through the wall obstacle which was meant to be avoided.Though it was second objective of the robot it neglected the walls and thought as a free space.

2.Since the target was in a static position the algorithm was able to locate it but if the target was movable in the free space and constantly moving how will the program perform.

2.The first thing which I though was we do not have any information about the motion of the robot with respect to time and if it was provided how the algorithm would behaved compared to this program.

3.There are many techniques to path planning and the agents chooses particular technique for a specific problem and implementing a particular algorithm .So,what results would have been there the agent chosed different methods.

4.The degree of freedom of the robot's motion.Since the robot ignores the current coordinates and moves to next coordinates and doesn't go out of the map scope what results would be there if there was no finite map.

5.Checking accuracy. The result for heuristic in A*search should return shortest paths. So to prove that it should be done by hand in a graph but what results would have been seen if it was compared with the other search methods.

6.If either the source or target was enclosed inside a wall and since the program seems to ignore the interior obstacles will the algorithm find

the target ?or either if the program avoided the obstacles and the target was enclosed in a wall then what would the result be?

Conclusion

In this work project, the path planning strategies for a robot path planner was proposed and implemented sucessfully on a real life activities using different mathematical equations, heuritsic algorithm and a uniform search method I.e A -star algorithm.In a*, we determine the order by a lower bound on the total length of the path,which includes the distance from the robot,to which is added to a lower bound on the distance still to be traveled.Thus,A* is suitable search algorithm for the proposed path. The proposed path planning are realized by appropriate code using visual studio c++.

The fundamental goal of this project was to find a path from a source to the target location. The robot(target)finds the target location using A* search and heuristic values. The path planning was sucessful as the trajectory was found as a shortest path to the target.

Though there seems to be the problem where robot doesn't recognize the interior wall as obstacle,it passes right through them.This may cause the error result.

Proper use of search method and practical work in the real life ,the agent can learn a lot from it.

In summary,this project work is carried out for the development of path planning strategies which are to be implemented for mobile robot task like pick-and –place, material distribution, automated storage and retrieval operation etc using heuristic algorithms.This project was also shows the ethics of an AI in real life,and potential benefit as how it helps the man kind and its application in the future world.