Ahmed AL Hilal

74869

CSC 4301 (01) – Intro. to Artificial Intelligence

Dr. Tajjeeddine RACHIDI

Spring 2022

Pathfinding Project Report

In this project, we have been asked to experiment pathfinding algorithms and different heuristic strategies assuming they all have the same cost and implement them in a game engine where a seeker agent will try to find the target. The algorithms used are: (A*, UCS, BFS, DFS).

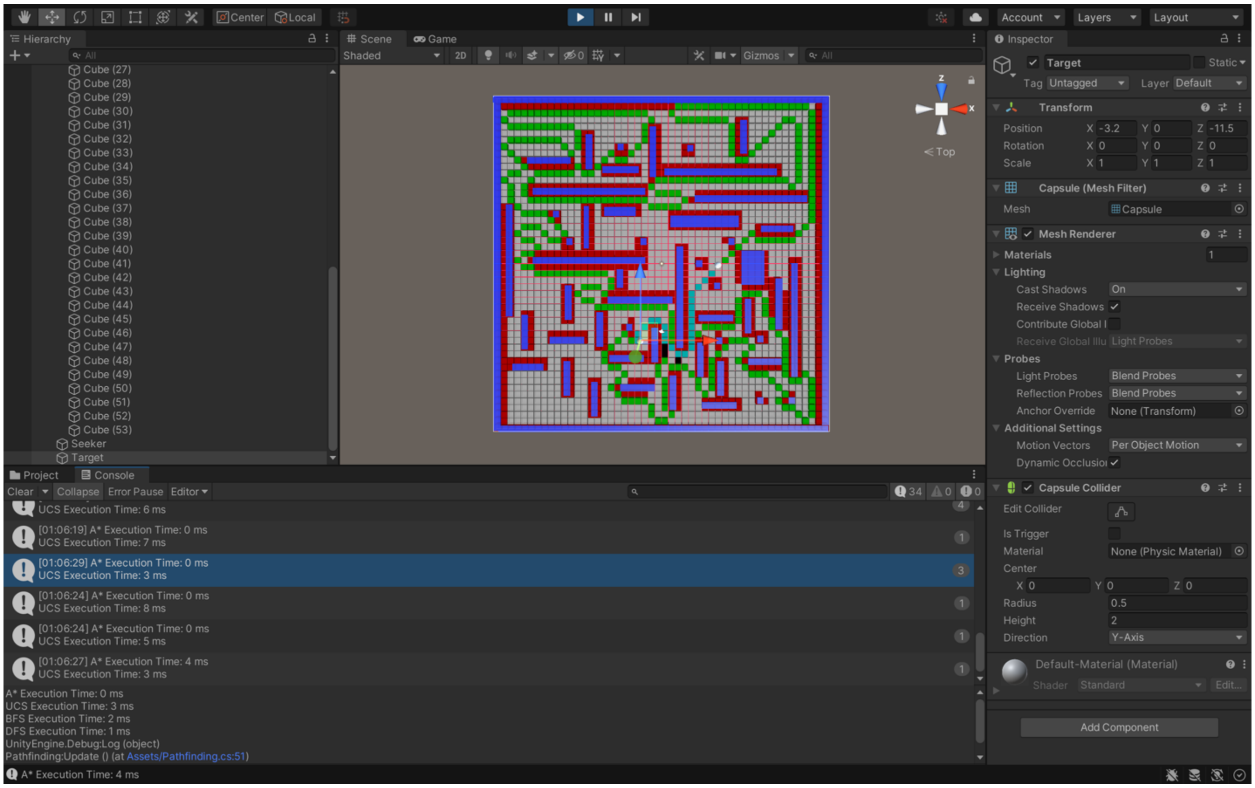
First of all, I have created an environment in Unity that resembles a maze. I tried to use a big plane where the obstacles will be placed in a tricky way to make it harder for the algorithm to realize its purpose. Another reason of doing so, is to make it easy for us to see the difference between the algorithms since the bigger the plane the more processing power needed to reach the target. I have placed the seeker and the target nodes in many positions to examine the performance of each algorithm. On average, the A* is by far the best performing algorithm between the four algorithms in terms of efficiency and time. It can allocate the target in the minimum time and with the fewest steps possible. DFS and BFS are both competing for second place, or at least for my case since it depends on the position. And finally, UCS comes last. I have used many positions to calculate these times in ms and the outcome didn't vary much. The most noticeable thing is that A* is locating the target in a quicker time compared to the other algorithms which is interesting. I have used multiple colors to differentiate between the search algorithms and to make it easier to compare. The BFS algorithm's color somehow seemed to not work. I suspect that since BFS and DFS are similar in terms of performance, one of the two colors overlaps the second algorithm, however, in the console the output seemed to be fine.

I have summarized the performance of the algorithms of the pictures posted in Github into the following table:

Algorithm Name	Position 1	Position 2	Position 3	Position 4
A*	2 ms	0 ms	0 ms	2 ms
BFS	6 ms	5 ms	2 ms	1 ms
DFS	4 ms	0 ms	1 ms	0 ms
UCS	8 ms	9 ms	3 ms	4 ms

Although DFS seems to win against BFS, in other positions BFS was actually winning too against DFS. That is why I think they both deserve the second place since the position is the sole element that can tell us which algorithm will win. Moreover, as we can see A* is winning in all positions on average with a big margin and UCS is in last place in terms of time.

On the other hand, when we talk about efficiency in terms of steps to reach the target, A* is again winning reaching the target in the fewest steps possible. However, what is interesting is that although UCS is slow compared to the other algorithms it does compete with A* in terms of steps since it reaches the target in the fewest steps too. DFS and BFS as we can see from the picture bellow are quick, but not efficient since they can traverse the whole plane sometimes only to reach the target.



Therefore, to summarize. I have learned a lot from this project. The visualization implementation can give us a sense on how the four algorithms will behave. A* is the ideal algorithm to use in all positions. However, there might be other variations where the other algorithms can outperform it. I am sure that each algorithm is specific to the environment dedicated to it. Eventually, what we can conclude from this project is that the obstacles and the plane both play a big role in the performance of the algorithm. At first, I have tried a smaller plane and the calculations were almost 0 ms for all algorithms. This means that the algorithms didn't use many processing power to reach the target. After I have increased the size of the plane, I started noticing the difference and also the steps started getting narrower for some and wilder for others. We have seen how some algorithms can beat others in terms of time and also in steps to reach the target. Finally I would like to give credits to Sebastian Lague for his source code to create the environment in Unity and the A* algorithm.