AI Project 1: Pathfinding Using AI: Informed and Uninformed Search Algorithms

Mohamed Moumou & Ouwais Zlaigi

12/06/2022

Definitions of the algorithms that we will simulate

A* (video heuristic): A* algorithm with a heuristic that is a combination between Manhattan and Euclidean distance. This heuristic is admissible (explanation below). This heuristic basically describes the shortest path between two nodes assuming that there is no obstacle.

A* Euclidean distance heuristic: Using A* with Euclidean distance that represents the distance between two points. To find the two points on a plane, the length of a segment connecting the two points is measured. We derive the Euclidean distance formula using the Pythagoras theorem.

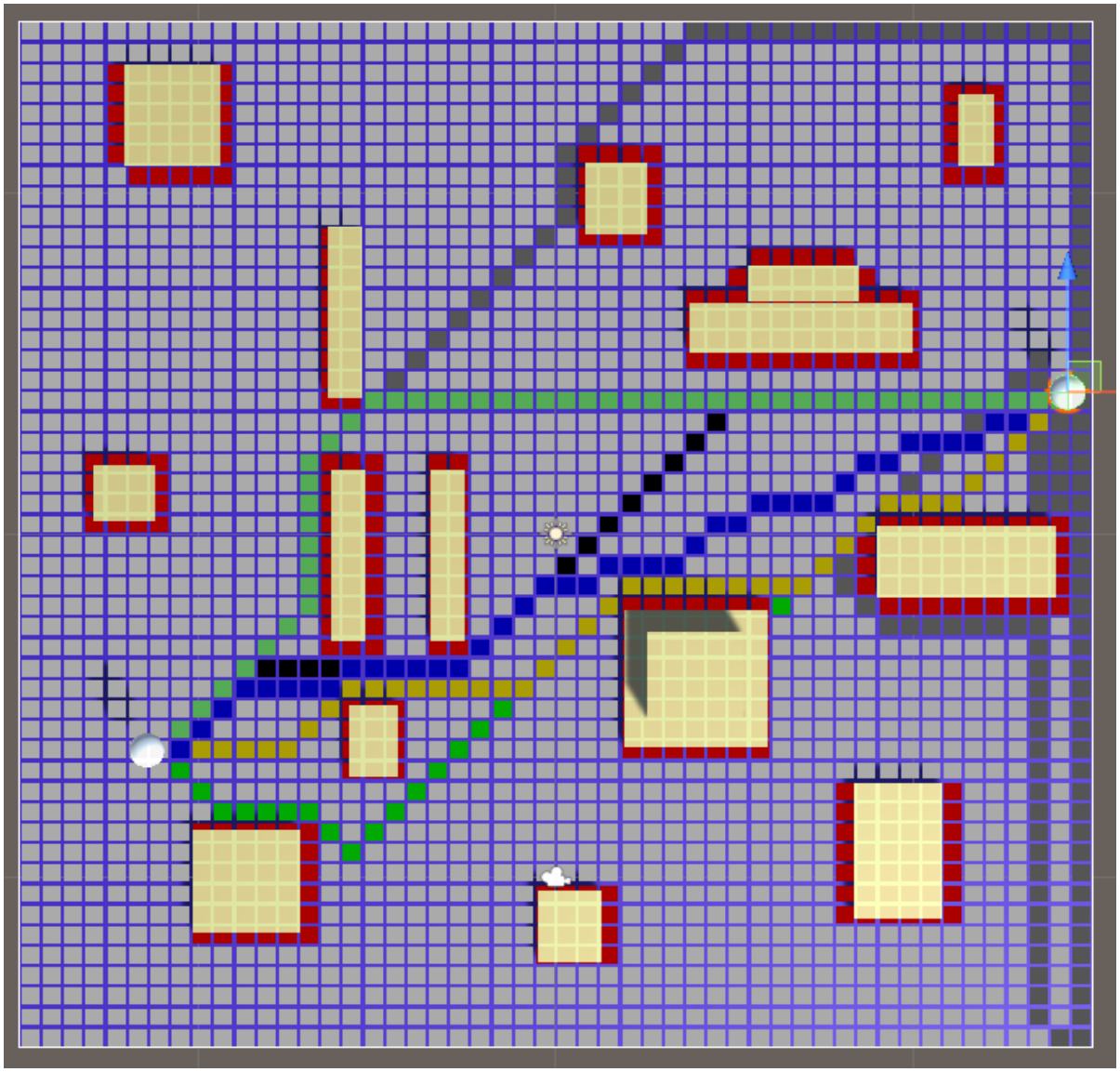
A* Manhattan distance heuristic: Using A* with Manhattan distance that represents that is the sum of absolute differences between points across all the dimensions.

DFS: A recursive algorithm for searching all the vertices of a graph or tree data structure is called depth first search or depth first traversal. Traversing a graph entails visiting all of its nodes.

BFS: Breadth First Search is a vertex-based method for finding the shortest path in a graph. It employs a Queue data structure that follows the first in, first out principle. In BFS, one vertex is visited and marked at a time, then its neighbors are visited and stored in the queue.

UCS: Uniform-cost search is an uninformed search algorithm that finds a path from the source to the destination by calculating the lowest cumulative cost. Starting at the root, nodes are expanded according to the lowest cumulative cost.

The simulation



Using Unity, we could simulate a game in which there is a start and end point, and the aim is to find the shortest path from the start to the end. The playground contains obstacles which does not make the game very intuitive (Euclidean distance). In the image above, we see 6 different colors. These colors represent 6 different algorithms. Some of them produce the shortest path, but others do not. The following is a map of colors and their respective algorithms:

A* (using the video's heuristic) => Black

A* (Euclidean heuristic) => Blue

A* (Manhattan distance) => Light green

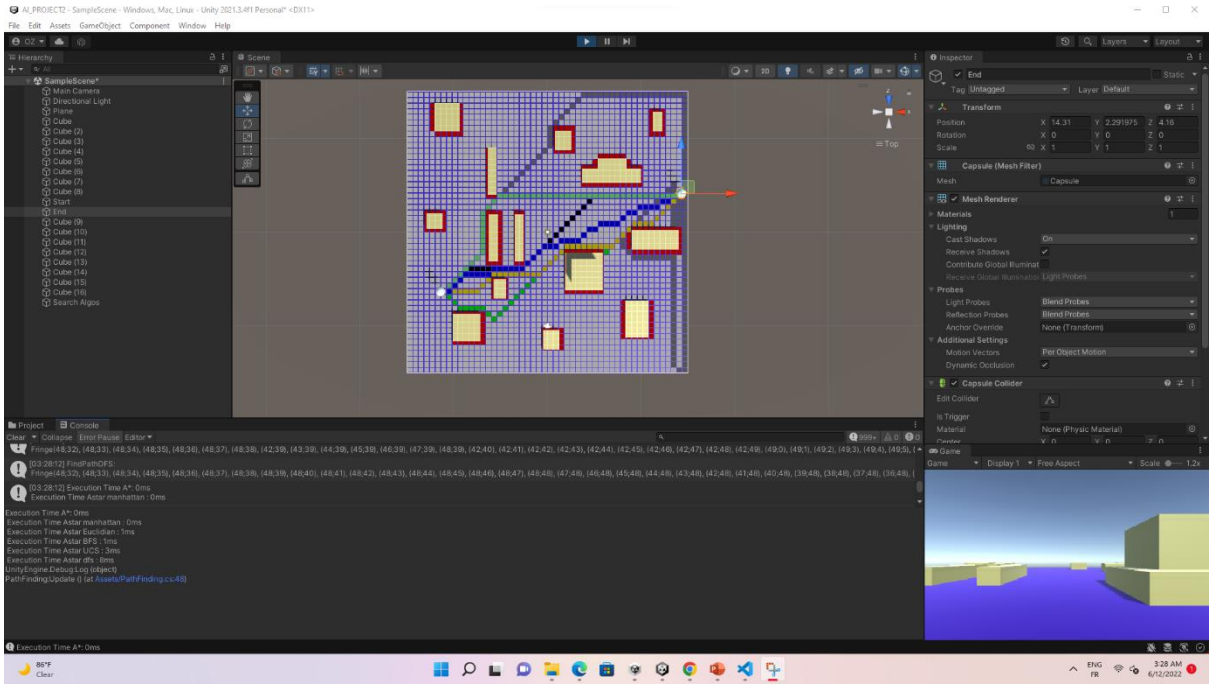
DFS => Gray

BFS => Green

UCS => Yellow

As we can also see, the grid walkable nodes are marked in white, whereas the grid unwalkable nodes are marked in red.

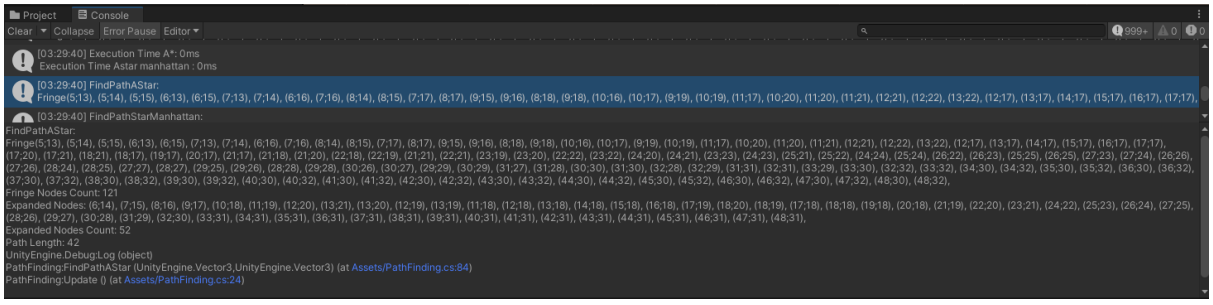
The following is the view of the whole screen.



As each algorithm is printed we got to analyze it, by printing its fringe, fringe length, expanded nodes, expanded nodes length, and the path length. These parameters will help us analyze each algorithm.

Analysis of each algorithm

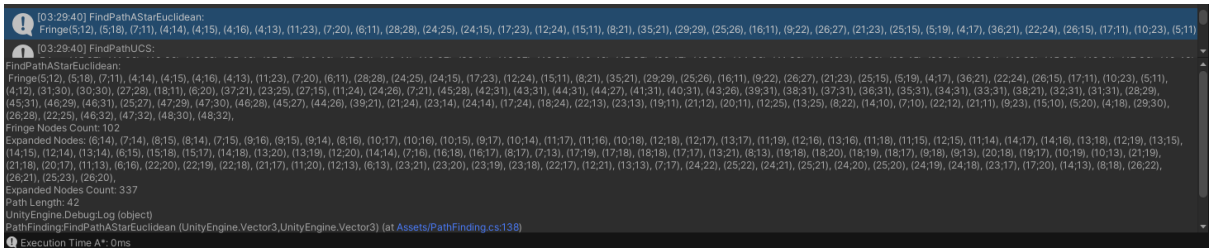
- A* (video's heuristic)



The video's heuristic is a combination of the Manhattan and Euclidean distance, The heuristic gives the shortest distance between two nodes on a grid, using only moves allowed by the grid, and as such combines both horizontal and diagonal paths. The heuristic is admissible because if there is an obstacle or another path, the distance will be greater to reach the target node. As such, this algorithm provides the correct answer, i.e., the shortest path length is 42.

We also notice that the fringe length is only 121, whereas the length of the expanded node is 52, 10 more nodes that the path length. The latter is a good indicator, because it means that our heuristic is good and we did not need to explore that much to find the answer.

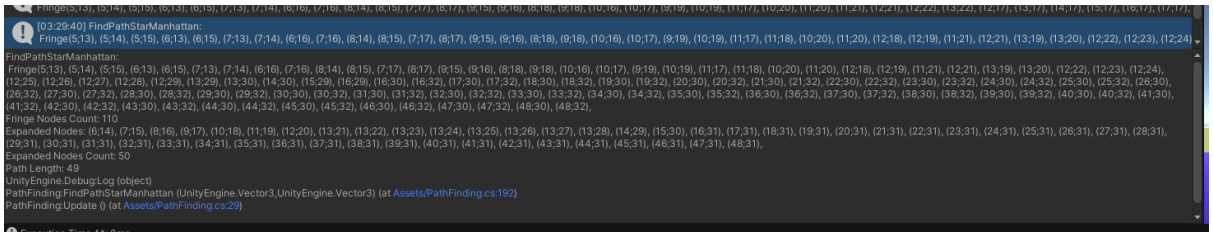
- A* (Euclidean heuristic):



First, the Euclidean heuristic is an admissible heuristic, because it gives us the shortest path between two nodes, regardless of the allowed moves on the grids. The latter means, the Euclidean heuristic will be underestimating the distance to the goal state more that the heuristic we discussed earlier. So, we proved again that this heuristic is admissible. Indeed, it provided the right answer, i.e., the shortest path length is 42.

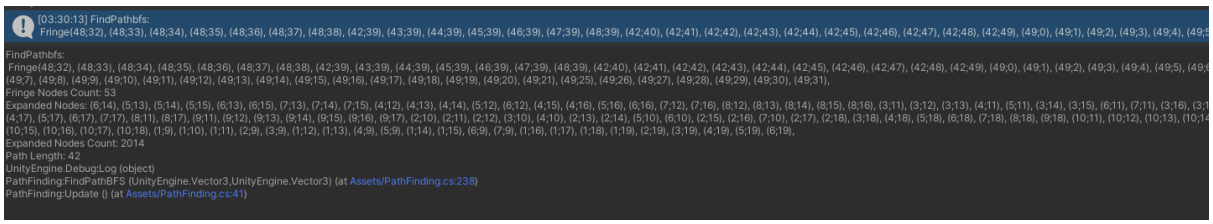
In addition, we notice that the fringe and the expanded nodes length in this algorithm are bigger, 102 and 337 respectively, which proves again that this heuristic is underestimating the length of the path to the goal more than the previous one.

- A* (Manhattan heuristic):



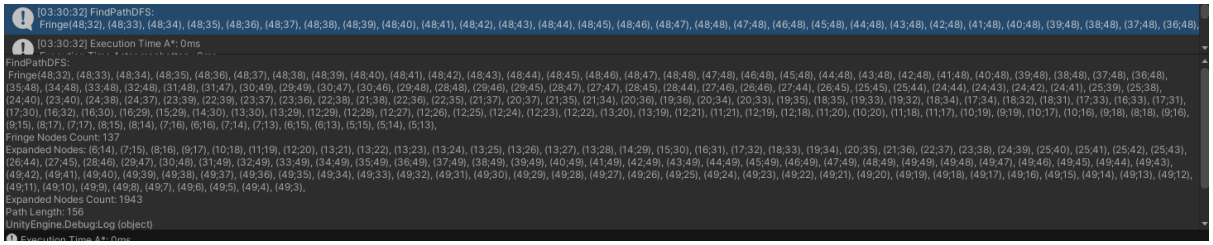
Looking at the path length, 50, we can advance that the Manhattan heuristic is not an admissible heuristic, because it provided a wrong answer. We can clearly see why, because in a grid with no obstacles, the shortest distance will be a combination of horizontal and diagonal paths, and not a rigid Manhattan distance (in most cases). Regarding the sizes of the fringe and the expanded nodes, they are 110 and 50 respectively.

- BFS



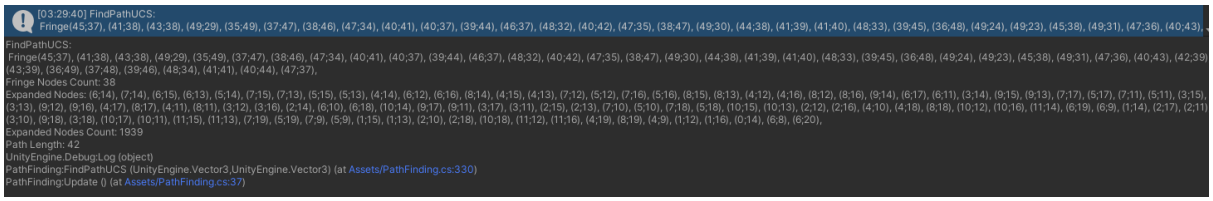
From this data, we can clearly notice that the BFS gives the right answer, 42. The latter makes sense, because the BFS algorithm explores all the surrounding nodes, until it can find the goal. However, this algorithm is very expensive. It is expensive to the extent that the expanded nodes almost contain all nodes that we have in the grid, 2500. Indeed, the expanded nodes are 2014, and the fringe only contains 53 nodes.

- DFS



From the simulation, the DFS algorithm does not provide the right answer. In fact, it is way off in many cases. The length of the path in this case is 156. The fringe and the expanded nodes have a size of 137 and 1943 respectively, which signals that this algorithm is not efficient does explore and visit many nodes, and most of all does not find the right answer.

- UCS

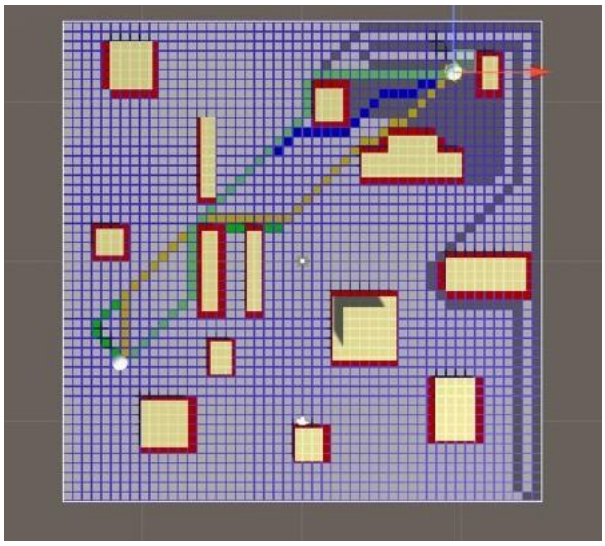


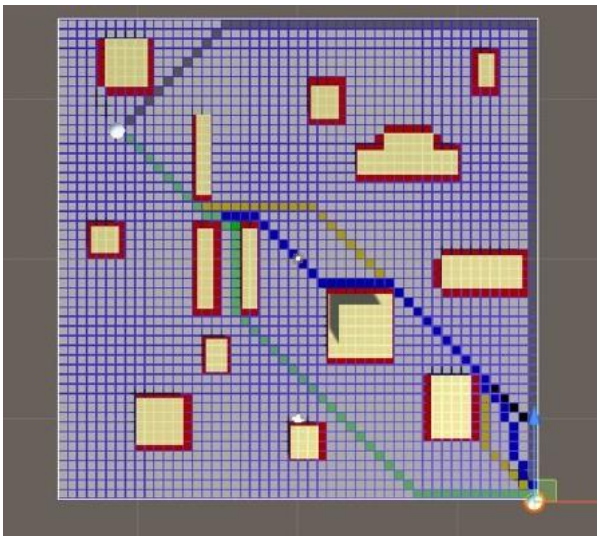
We notice that the UCS gives the right answer, and that is because it explores the grid in a fashion very similar to BFS (although the logic is quite different. As such, we notice that there are a lot of expanded nodes, 1939, and automatically a small fringe, 38.

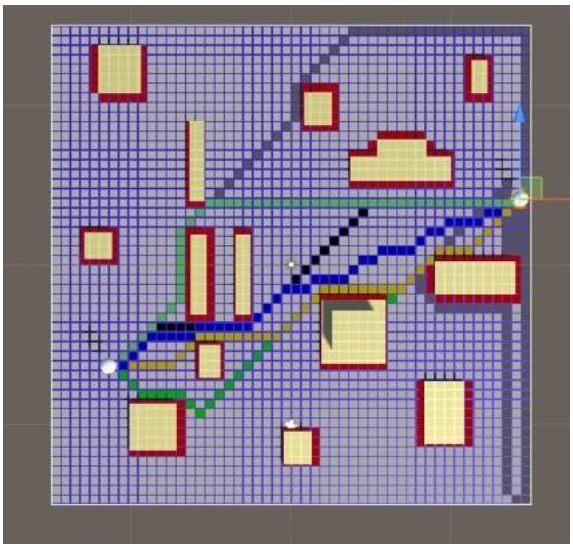
After exploring and analyzing each algorithm, let us look at the time it takes to execute these algorithms to better understand the benefits of informed search.

We can clearly see that informed search algorithms (A* algorithms) perform better (between 0 and 1ms), whereas the other algorithms, the uninformed ones, have a worse performance (between 1 and 8ms). Although, these measures might not be accurate all the time, they at least give an idea about the performance of informed search algorithms against uninformed search algorithm. To get an idea about the performance of an algorithm, it helps first to have a more complex grid, so that the performance of the algorithm is clearer. Also, by taking these measures (for the same configuration) multiple times and averaging them, we can get very close to the right elapsed time for each algorithm.

Images for different positions of the start and end node







Conclusion

In this project we reproduced the pathfinding and experienced different search and heuristic strategies. First, following the instructions of the A* (video heuristic) we got to simulate the 5 other algorithms, A* Euclidian distance, A* Manhattan distance, BFS, DFS, and UCS. As stated previously, we got to analyze each algorithm by printing its fringe, fringe length, expanded nodes, expanded nodes length, the path length, and time. These parameters helped us to compare between each algorithm. So far, we got the appropriate simulation of all algorithms and relying on the previous parameters we could reach to the project goal by understanding the algorithm behind each strategy and use the parameters to check whether our algorithm is working well or not.