Project #2: Pathfinding in unity using A, UCS, BFS, and DFS*

Our team: **Taha Zaamoun**, Youssef Filali Benaceur.

foreword:

Youssef and I got inspired from SebLague's code and algorithm for acquiring the shortest path for a start to a target node using the Unity game engine and C# scripts. We have upgraded and added new algorithms to the code such as:

- A*–Search Algorithm using the Manhattan distance.
- A*–Star Search-Algorithm using the Euclidean distance.
- Depth–First Search Algorithm(or DFS).
- Breadth–First Search Algorithm(or BFS).
- Uniform–Cost Search Algorithm (or UCS).

In this project, We first established our own environment–scene using unity; The latter contained obstacles(where the algorithms could not cross), colour–coded algorithm–based paths, and finally a target and a start node to denote the boundaries of the path. Sebastian Lague implemented A* using his own heuristic. The strategies we were instructed to implement are: BFS, DFS, UCS, and the already implemented A* algorithm but through adapting it through both the Manhattan and the Euclidean distances' heuristic. The parameters that we used to compare between the above–mentioned algorithms are:

-Time it takes to reach the target node from the start node(in ms)

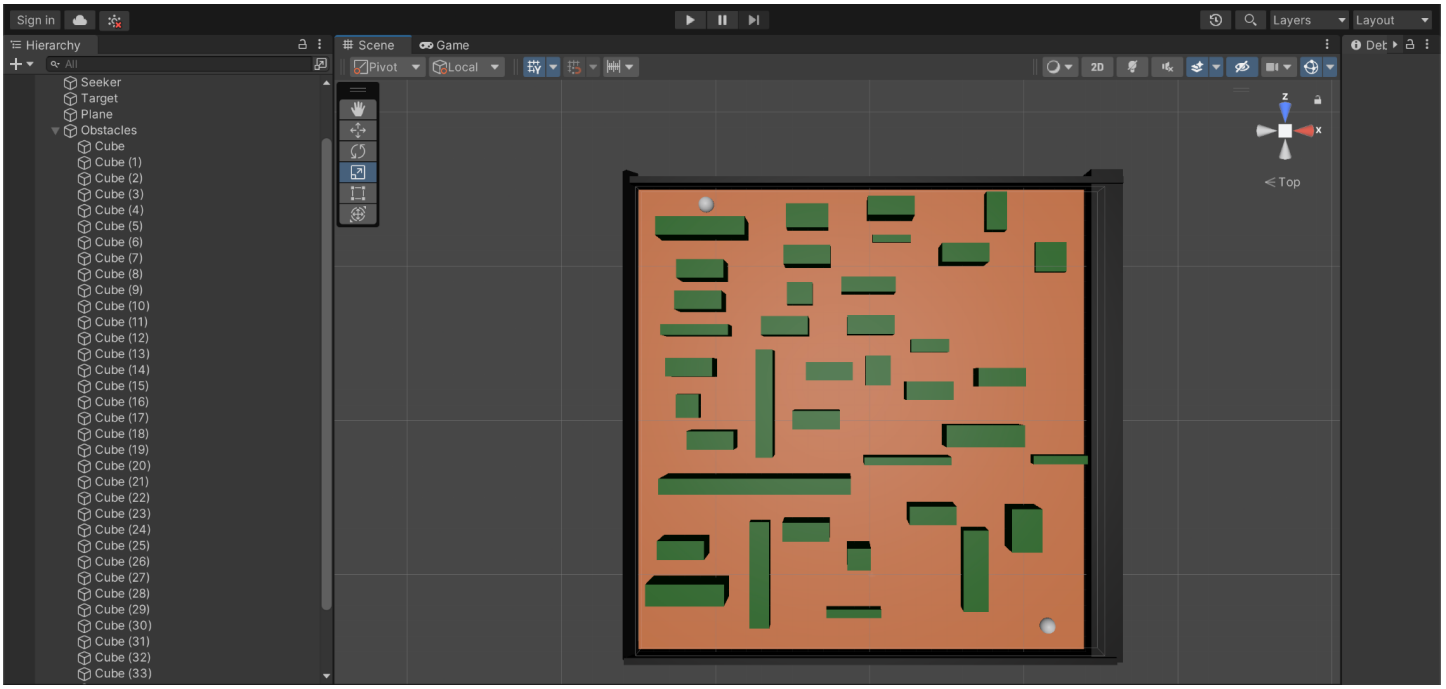
-Number of total nodes encountered to reach the target node from the start node

Acknowledgements:

Special thanks to Sebastian Lague for the inspiration and initiation code and visual immersion.
Sebastian's YouTube channel:
https://www.youtube.com/watch?v=-L-WgKMFuhE&list=PLFt_AvWsXl0cq5Umv3pMC9SPnKjfp9eGW
Sebastian's GitHub repository for Pathfinding:
https://github.com/SebLague/Pathfinding

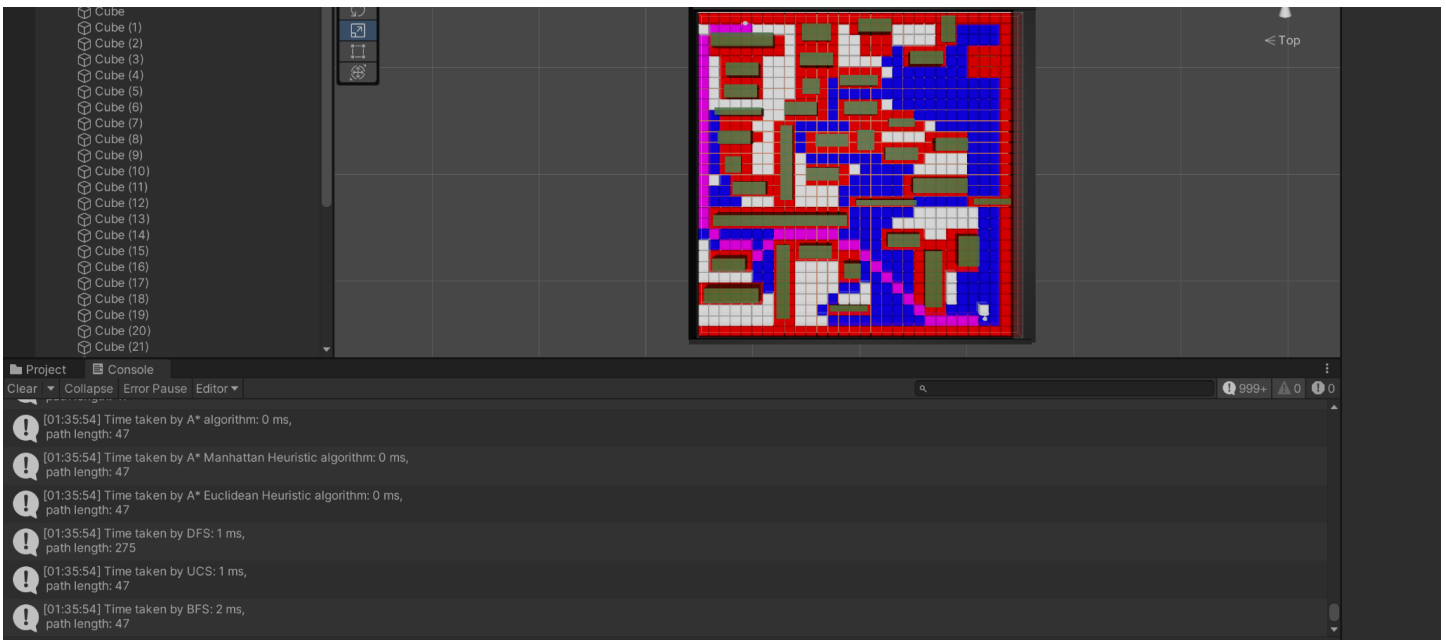
Environment:

Here is the environment we worked with:



As we can see, these are the different paths travelled with different algorithms. The best ones of these paths are the yellow and purple which belong to the uniform cost search(UCS) and Breadth–first–search(BFS) algorithm.

We can also see the execution time and retracement for all the algorithms which are displayed in the console when we run the game/scene.



Legend of the scene:

Blue: DFS

Purple: UCS

Yellow: BFS(Although it does not show a lot due to the constant overlap with the DFS path)

Teal: A*(Sebastian's heuristic)

Green: A*(Euclidean heuristic)

Grey: A*(Manhattan distance heuristic)

=> The colours that do not show up in the Game–view in Unity are due to overlaps with other paths.

We noticed from the console that DFS(Blue–colored path) was the slowest and gave us the longest path while travelling the maximum number of nodes(282 for example in the above–snapshot) which proves that it is not optimal. However, both UCS and BFS gave us the shortest paths in this case. A* with all heuristics performed in under a second but both UCS and BFS went through less nodes.

Afterword:

Performance comparison:

Time:

1st: A* with different heuristics(< 1 ms)

2nd: DFS(1 ms), UCS(1 ms)

3rd: BFS(2 ms)

# of nodes visited:

1st: BFS, UCS, A*, different heuristics; all averaging under 47 nodes

Last: DFS(275 nodes!)

Conclusion:

All in all, DFS is the least optimal path although relatively fast. BFS, UCS, and A* with different heuristics perform amazingly, but the absolute fastest and most optimal one is definitely A* with different heuristics.