**Assignment 2:**

**Unity maze shortest path using (UCS, BFS, A\*, DFS)**

Team Members: Ayoub Maimmadi, Hamza Ammad

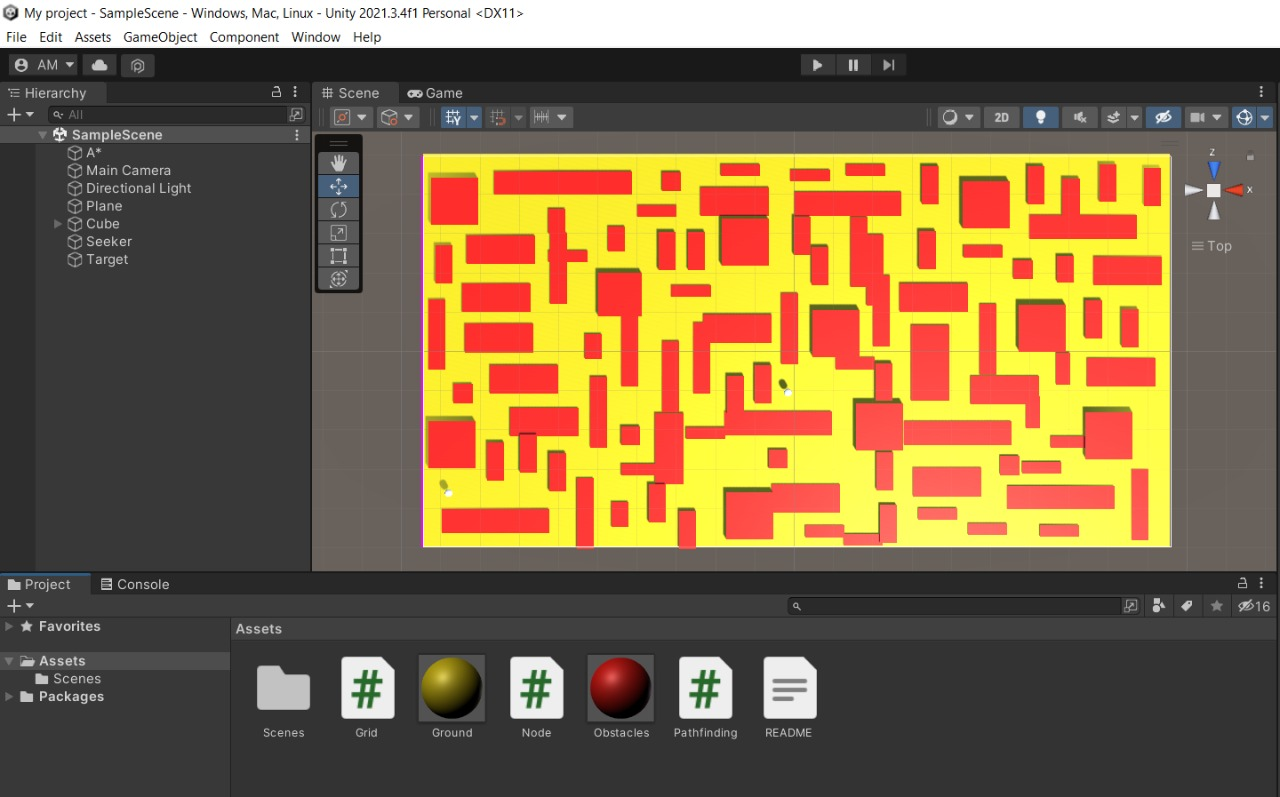
Me and my teammate Hamza Ammad have improved the code of @SebLague algorithm for finding the shortest path between a target and a start node in the Unity game engine. We have upgraded and added new algorithms to the code such as:

* A-Star-Search-Algorithm using the Manhattan distance **A\***
* A-Star-Search-Algorithm using the Euclidean distance **A\***
* Depth-First Search-Algorithm **DFS**
* Breadth-First Search-Algorithm **BFS**
* Uniformed Cost-Search-Algorithm **UCS**

@SebLague's YouTube channel: <https://www.youtube.com/watch?v=-L-WgKMFuhE&list=PLFt_AvWsXl0cq5Umv3pMC9SPnKjfp9eGW>

@SebLague's GitHub: <https://github.com/SebLague/Pathfinding>

1. The environment:

We first created our own environment with complex characteristics as shown in the figure below

1. Comparison of strategies

The screenshots below show the different paths using the algorithms simultaneously with different starts and goals (shown as white dots) as well in every screenshot.

* A\* Euclidian Search: White
* A\* Manhattan Search: **Red**
* Breadth First Search: **Green**
* Depth First Search: **Blue**
* Uniform Cost Search: **Black**

As we can see, these are the different paths our character can take with different algorithms. The best one of these paths is the white one which belongs to the A\* search algorithm.

A screenshot of a video game

Description automatically generated**case1:**

we can see that the white line is the optimal path for this specific map, followed by the red line. Both these lines belong to the A\* search algorithm. After these ones, we can see that the black line for the UCS is the 3rd optimal path, then BFS, and finally the blue line is the least optimal, which belongs to the DFS 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.

The execution time for the A\* Euclidian algorithm on average in 6ms with 144 retracements, and it’s 8ms on average for A\* Manhattan, with 147 retracements.

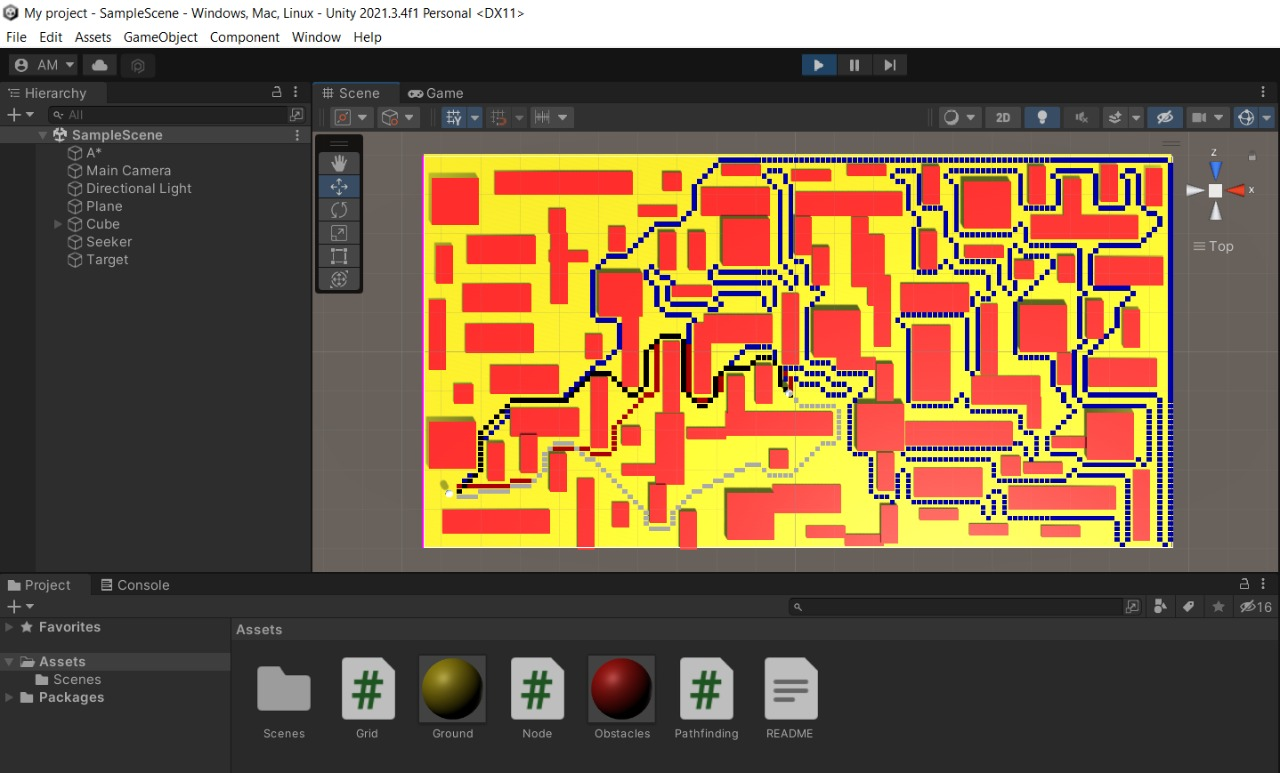
Text

Description automatically generated

A screenshot of a video game

Description automatically generated

**Case 2:**



For this case, the execution time for the A\* Euclidian algorithm on average in 4ms with 93 retracements, and it’s 3ms on average for A\* Manhattan, with 94 retracements.

