**Assignment 2:**

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

Team Members: Ayoub Maimmadi, Hamza Ammad

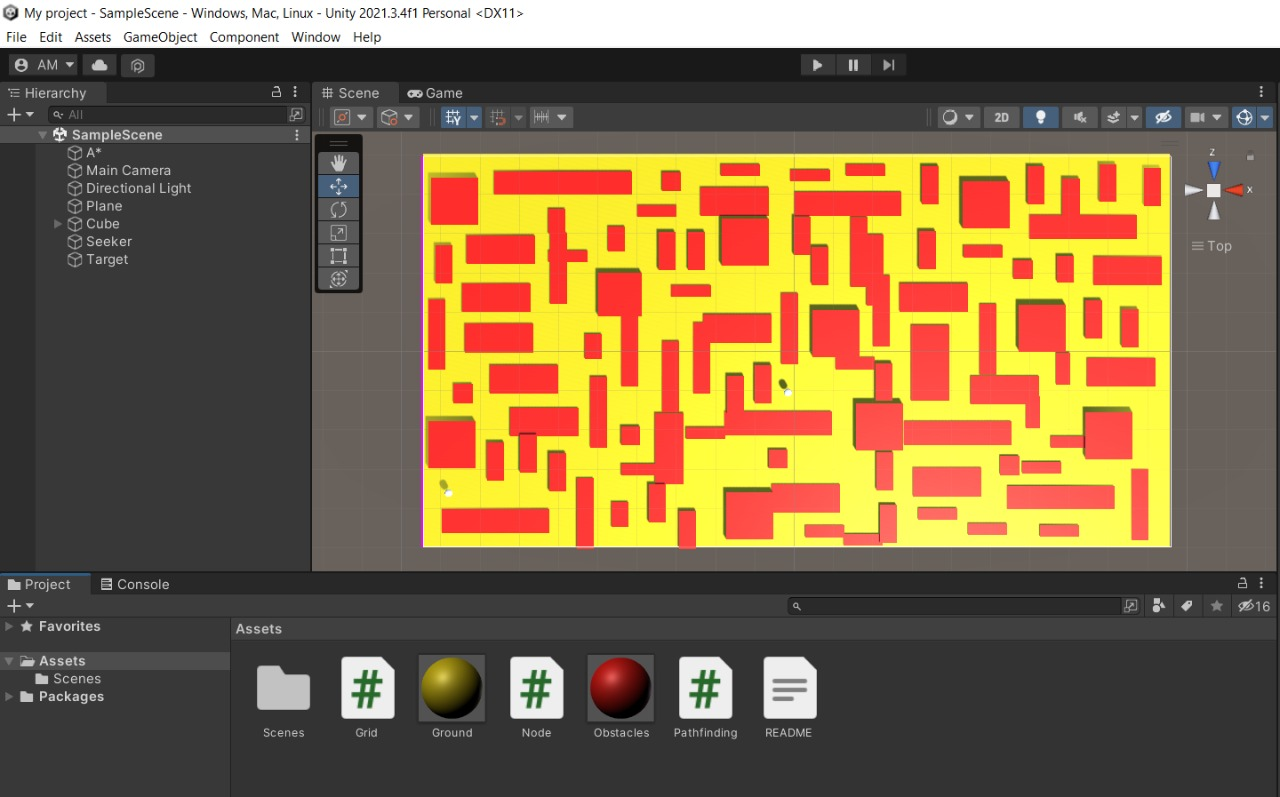
Me and my teammate Hamza Ammad have improved the code of @SebLague algorithm for finding the shortest path between a target and a start node in the Unity game engine. We have upgraded and added new algorithms to the code such as:

* 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. Implementing the search algorithms

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 ones of these paths are the black and white one which belong to the uniform cost search and the A star search 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.

A screenshot of a video game

Description automatically generated**case1:**

A screenshot of a video game

Description automatically generated

**Case 2:**

