**CSC 4301 Spring 2022**

**Text

Description automatically generated**

**Intro. to Artificial Intelligence**

**Assignement 1**

Pathfinding- A\*, DFS, BFS, UCS

**Supervised by: Dr. Tajjeeddine RACHIDI**

**PREPARED By: Asma Dalil**

**Kenza Rchi**

**Introduction** :

In this project , I and Kenza Archi tried to produce pathfindings suing Unity to simulte search agorithms (A\*, DFS, BFS, UCS) and heuristic strategies assuming these all have the same cost in all directions and A\* with different heuristics.

In the following screenshots , there are diiferent paths presented in different colors and the total time and notes visited by each strategy:

* DFS in red
* A\* Manhattan in white
* A\* Euclidian in blue
* UCS in green
* BFS in black

Github project link ; <https://github.com/AsmaDalil/AI--report--pathfinding>

**Summary :**

Euclidean Distance :

* It's the total of the absolute values of the disparities between the goal's x and y coordinates and the present cell's x and y coordinates.

Manhattan Distance :

* It's nothing more than the sum of absolute values of disparities between the goal's x and y coordinates and the present cell's x and y coordinates.

Breadth First Search :

* BFS stands for "Breadth First Search," which is a vertex-based approach for determining the shortest path in a graph. It makes use of a Queue data structure that operates on the first-in, first-out principle. BFS visits and marks one vertex at a moment, then its neighbors are visited and added to the queue.

Uniform-Cost Search

* Dijikstra's approach is a version called Uniform-Cost Search. Instead of placing all vertices in a priority queue, we place only the source, then insert one by one as needed. Every step includes a check to see if the item is already in the priority queue (using visited array). If that's the case, we'll use the decrease key; otherwise, we'll use the insert key.

Depth First Search

* A graph's Depth First Search is analogous to a tree's Depth First Traversal. The sole exception is that, unlike trees, graphs can have cycles (a node may be visited twice). Use a boolean visited array to prevent processing a node multiple times.

**Secreenshots :**

A screenshot of a computer

Description automatically generated with medium confidence

In this schreenshot, we can see the difference in time complexity and space complexity between the different searches.

* The A\* Eucridian took 3 ms and visited 30 nodes from the start position to the end position.
* A\* Manhattan took 0 ms and visited 29 nodes to reach the target from the start position
* UCS search visited 29 nodes in 2 ms to reach the target position from the start position
* BFS search viited 187 nodes in 0 ms to reach the same tartget position

Graphical user interface

Description automatically generated

Graphical user interface

Description automatically generated

Graphical user interface

Description automatically generated

**References :**

**Souce code :** [**https://github.com/SebLague/Pathfinding**](https://github.com/SebLague/Pathfinding)

**Youtube video :** [**https://www.youtube.com/watch?v=-L-WgKMFuhE&list=PLFt\_AvWsXl0cq5Umv3pMC9SPnKjfp9eGW&ab\_channel=SebastianLague**](https://www.youtube.com/watch?v=-L-WgKMFuhE&list=PLFt_AvWsXl0cq5Umv3pMC9SPnKjfp9eGW&ab_channel=SebastianLague)