

Faculty of Computer Science

Fall Semester 2020

CS316

Artificial Intelligence Project Fall 2020

**Done by:**

|  |  |
| --- | --- |
| **Name** | **ID** |
| Ahmed Mohamed Abdelfattah Issa | 191947 |
| Ali Mohamed Hobeldein | 192231 |
| Soliman Harby Gourgui salama | 191523 |

**Submitted to:**

Dr. Ehab Emam

|  |  |
| --- | --- |
| |  | | --- | |  | |

**Maze solving agent:**

In this project, we developed an efficient AI maze solving agent to find the shortest path from source to destination by using Path-Finding algorithms such as BFS, Greedy-Search and A\*

**Main Program:**

The program contains 5 class :

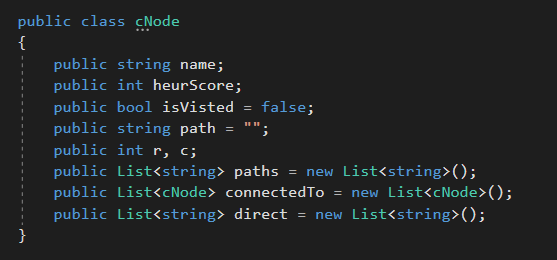
Main Form , cNode

Greedy Search – ( Ahmed Issa )

A Star – ( Ali Hobeldein )

Breadth First Search – ( Soliman Harby )

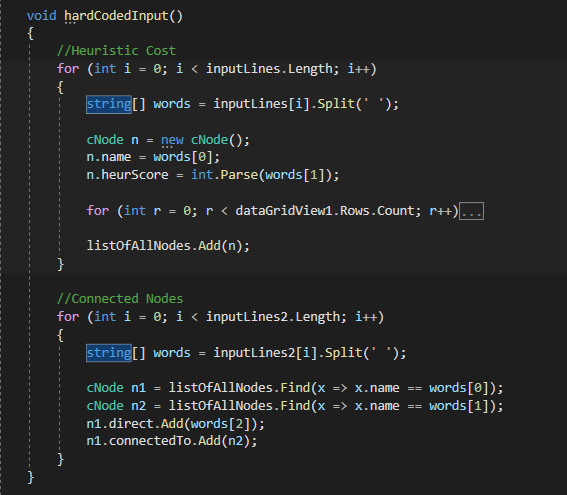
Class cNode which holds nodes data.



We have one hardcoded maze as an input.

hardCodedInput() is a function which is used to build a complete tree

nodes in ListOfAllNodes and connectedTo to add an edge to graph.



**Greedy Search:**

Greedy best-first search algorithm It is the combination of depth-first search and breadth-first search algorithms. It uses the heuristic function and search.

The function SearchGraph() traverse the tree using recursion

Step 1 : Pass the root to the function.

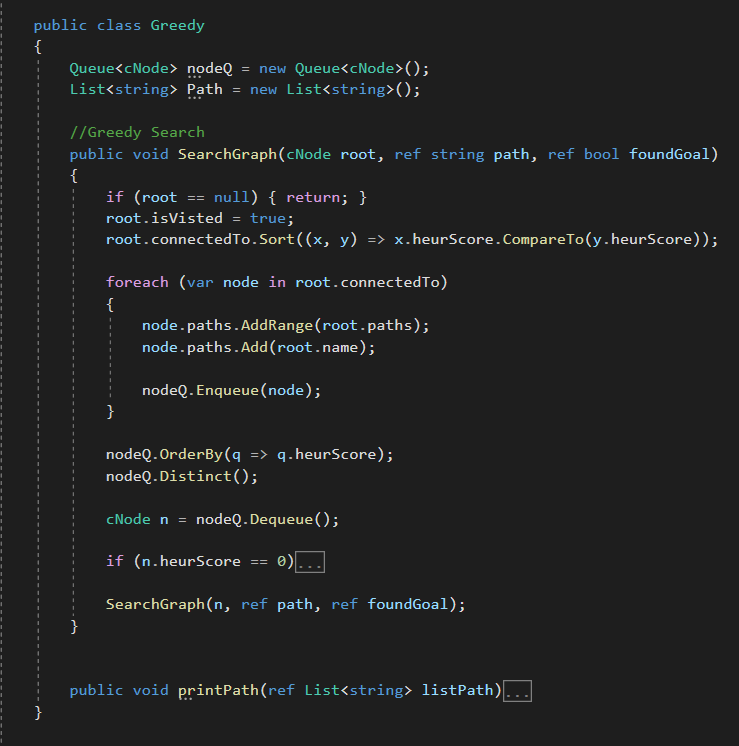
Step 2 : Check if the node equal null then terminate the search.

Step 3 : generate the successors of node from connectTo list and get nodes path.

Step 4 : Enqueue the nodes in ascending order and remove the duplicate with the greatest value.

Step 5 : Dequeue node and check if the node is the goal then terminate the search and return goal node path.

Step 6 : else take last node and return to step 2.



**Breadth First Search:**

We used data-grid-view to implement breadth first algorithm

Step1: searching for the goal node using first in first out by adding Childs and push them in a list.

Step2: after finding the goal by using function dataentery() which returns the path to the goal node.



After getting the goal node in a whole visited list then trying to get the direct path to the node from the start, and print it.

4- sending the path to the draw()function to draw it in the data-grid view and animate it by function animate().

