Technical University of Moldova

Operational Research

Laboratory Work 4

Finding shortest path between two points in a maze using A* algorithm

Author:

Luca Victor

Supervisor:

Eugen Torica

1 A* algorithm

A* is a widely used pathfinding algorithm. It is used in a lot of fields. For example it might be used to find shortest path from point A to point B. A* algorithm find the shortest path among all possible paths. At each iteration of the main loop algorithm picks the path with the smallest total cost. Total cost is composed from: G which is distance from the source node + H which is distance to the destination node. The algorithm is using to lists: open, closed. In open list there are nodes which haven't been reached, meaning that they don't have shortest path yet. In the closed list we have all the nodes which already have there shortest path. When we add node to closed list we check all possible nodes around and determine there total cost. The algorithm is stopped when H cost of the current cell which is added to closed list is equal to zero.

2 Which algorithm is faster

If we would compare number of iteration needed to compute shortest path in a maze using Dijkstra's or A* algorithm then we would say that A* algorithm is much more faster. In the worst case(ex: $r = f(\theta)$ nr of iteration needed is equal.

3 My implementation

My github repo is attached to laboratory work. As input I use txt file with the maze where: '-' is wall, '+' is walking path, 'A' is source and 'B' is destination node. In the end program creates new output file with solved maze where 'p' letters shows the shortest path.

4 Conclusion

It took me a lot of time to implement the algorithm by myself using just pseudocode and the tutorial from youtube. Still my program have some bugs and I hope to figure out why is it crashing in some cases. If there would be a program written in openCV for example which transforms image to needed format in .txt file it would be possible to solve any kind of maze regardless of the type.