NORGES TEKNOLOGISK-NATURVITENSKAPELIGE UNIVERSITET

TDT4136 - Assignment 3

Sondre Løvhaug
Mikael Mathisen

October 16, 2017

TASK 1 - A*-ALGORITHM IN SIMPLE BOARD FILES

1. *Well-commented source code for a program that finds shortest paths for boards with obstacles and that visualizes the results. If you did not write the A* implementation your self, specify where you got the code from.*

   We wrote the code our self, but got some help with implementation by searching the internet. The source code is attached.

2. *Visualizations of the shortest path for the four boards given above.*

   We chose to use text-visualization, where "X" marks the shortest path. See the appendix with the name "Output1-2.txt" for the visualization.

TASK 2 - A*-ALGORITHM IN COMPLEX BOARD FILES

1. *Well-commented source code for a program that finds shortest paths for boards with different cell costs (as specified in Table 1) and that visualizes the results.*

   The source code is attached.

2. *Visualizations of the shortest path for the four boards given above.*

   See the appendix with the name "Output1-2.txt" for the visualization.

TASK 3 - A*, DJIKSTRA AND BFS

1. *Well-commented source code for a program that finds paths using A*, BFS and DijkstraâĂŹs algorithm and that produces visualizations with open and closed nodes, such as the one above.*

   The source code is attached. This is visualized by marking the nodes with (x) if they are open and [x] if they are closed. See the appendix with the name "Output31.txt" for visualization

2. *For at least one game board from Part 1 and at least one from Part 2, show your visualizations using A*, BFS and Dijkstra. This should give at least 6 visualizations in total.*

   See the appendix with the name "Output32.txt" for visualization

3. *For each game board processed above, a brief analysis of (a) any differences in the path found by A*, BFS and Dijkstra, and (b) any interesting differences in the number of open and closed between for the different algorithms.*

   - Board 1-1
     Interesting about board 1-1 and 1-4 is that we can clearly see the advantage of the A Star algorithms estimate of distance. Both the Djikstra and BFS has to close almost every node on the board to be sure of the shortest route while A* can simply know that huge amounts of the board does no need to be checked.

   - Board 2-4
     Board 2-4 is interesting because Djikstra and A* perform quite similarly. This only goes to show that when they both deal with weighted nodes they both handle the situation good. This could be a computing advantage for the Djikstra since it has one calculate function less than A* and only looks at the actual cost while A* estimates distance in addition. On systems where memory is crucial this could be relevant, but not as much on small programs that we run on everyday computers. In comparison the BFS does not care about weights at all and just brute forces its way from A to B, while not considering many of the other possibilities.