Project 3 Report

A* Path Planning

Objective:

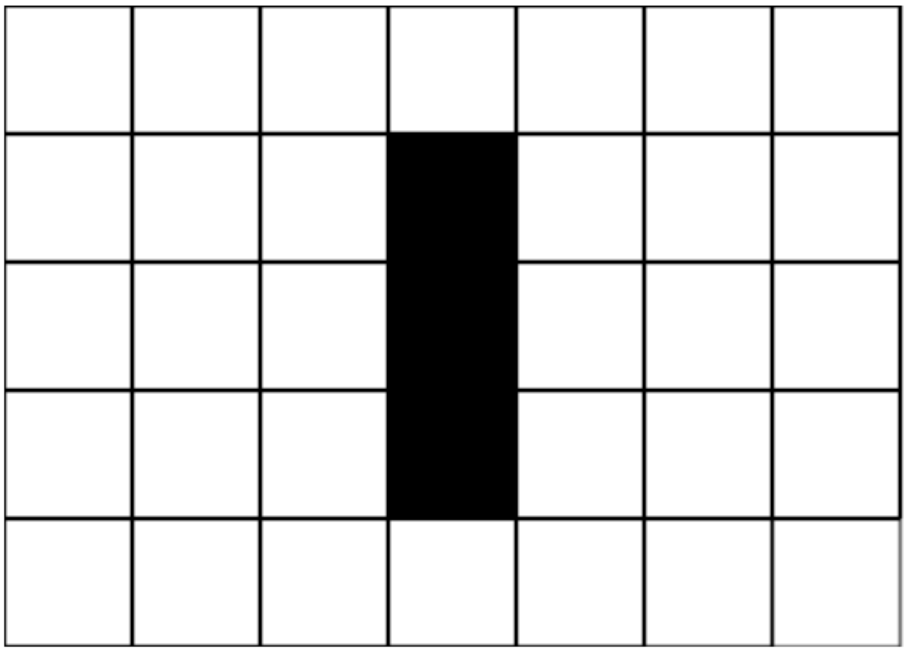
The main objective of this project is to find a path between one point to another in a grid. You select a map which is in the form of a text file containing e for empty space and o for Obstacles. You then choose a starting source node and destination node and then click start to find the path automatically or to find the path step by step.

Working:

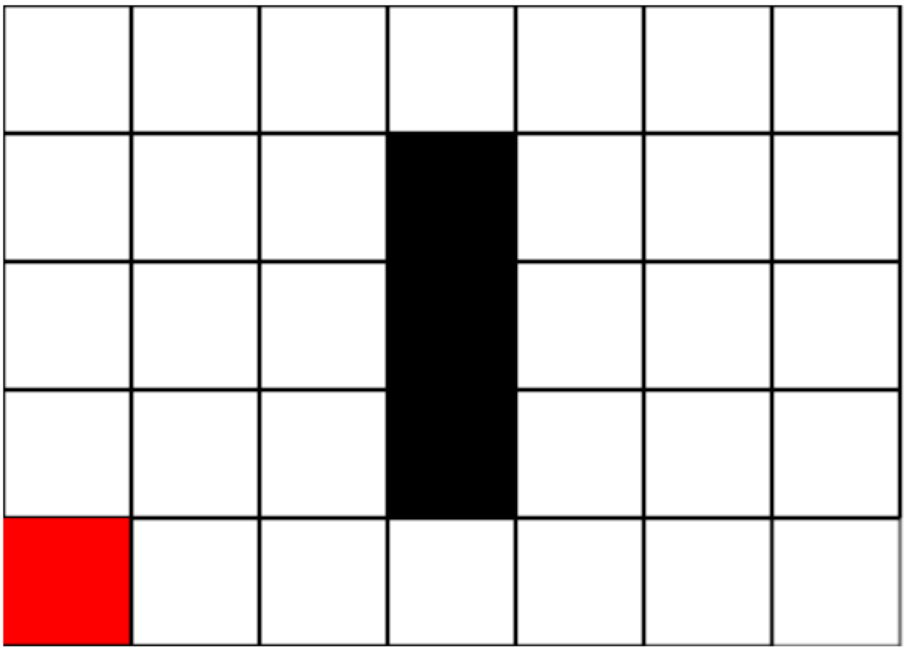
The user writes a map for drawing the grid which contains a sequence of e and o representing empty space and obstacles respectively, the program reads the file and creates a grid which white spaces where e is printed and black spaces where o is printed. For example if the user writes a map as following:

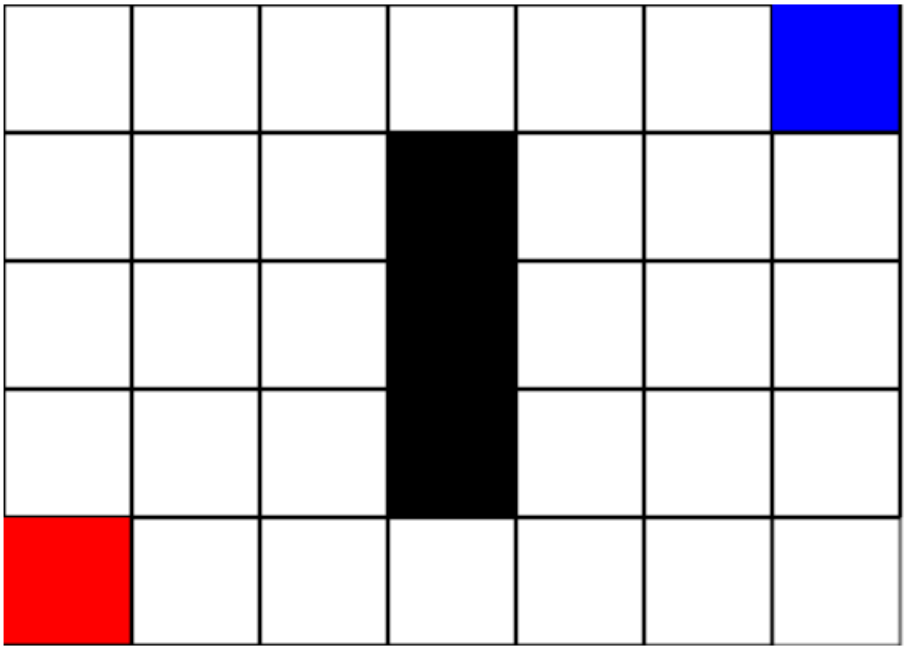
eeeeeee
eeeoeee
eeeoeee
eeeoeee
eeeeeee
eeeeeee

The Grid is drawn as:



Once the grid is drawn the program waits for a user input to select a white cell to be considered as the source. When the user clicks on the cell it is colored as red to highlight source. The program again waits for another user input which represents destination. Once the user select a destination it is in blue color. The grid will now look as follows:





Once the user has selected the grid the algorithm finds the neighboring cells of the source cell and highlights them in green. As soon as it find the neighboring cell it detects the distance of those cell as $g(x)$, which is 10 if the neighboring cell is in the same y axis or x axis as the source

and 14 of the cell is diagonal to the source, it also finds the distance of the neighboring cells to the destination node, this distance is h(x) which is based on the following heuristics:

$$h(x) = |i_{destination} - i_{source}| * 10 + |j_{destination} - j_{soucre}| * 10$$

This is simply the straight distance from the source to destination without considering the obstacles in between.

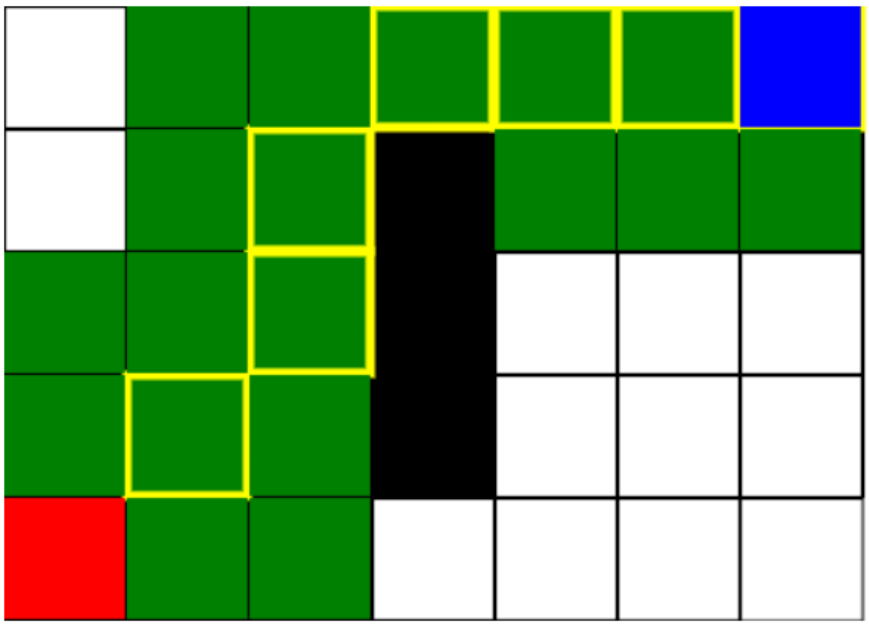
Finally the factor used to compute which node is to be selected by the A* star algorithm to be chosen next in the path is $f(x)$ which is simply the addition of $h(x)$ and $g(x)$.

$$f(x) = g(x) + h(x)$$

The neighboring cell with the minimum $f(x)$ is selected as the new node to be applied A* on.

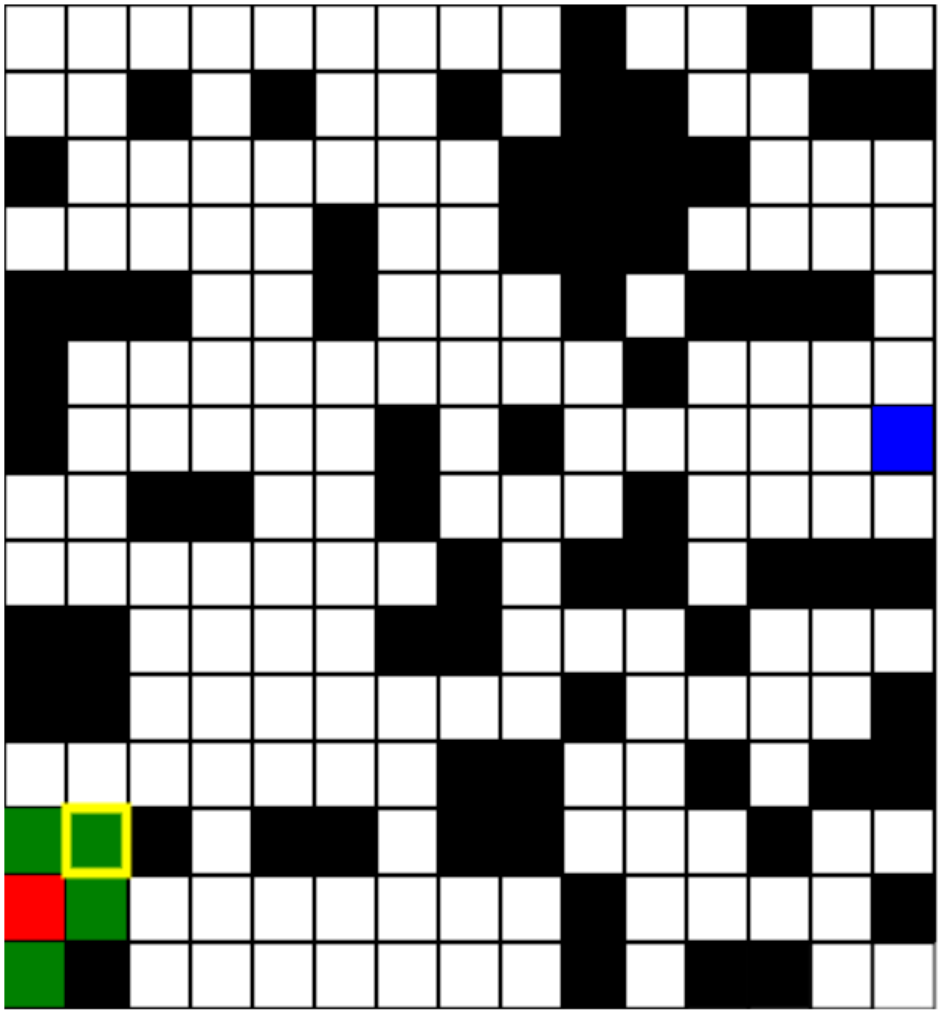
Screenshots:

1) 7x5 grid

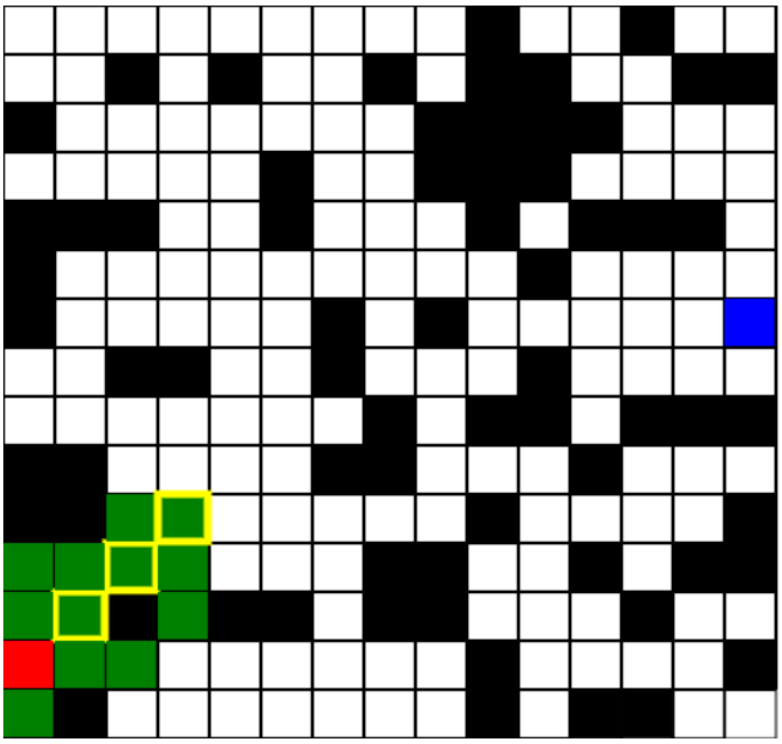


Previous F(x): 82
Previous G(x): 72
Previous H(x): 10
Final F(x): 82
Final G(x): 82
Final H(x): 0

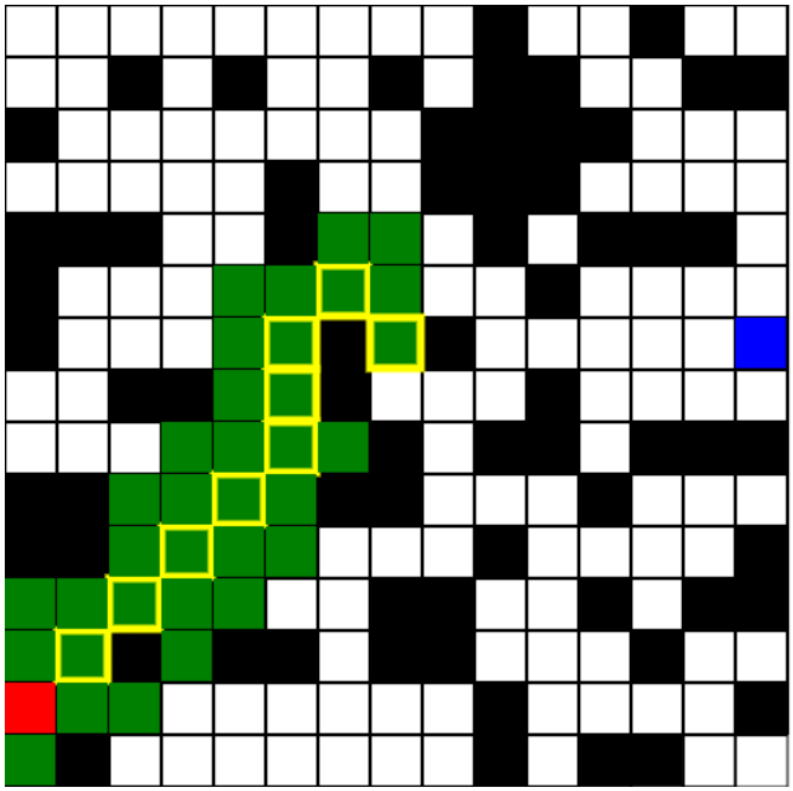
2) 15x15 grid – More than 200 cells – Step by Step screenshot



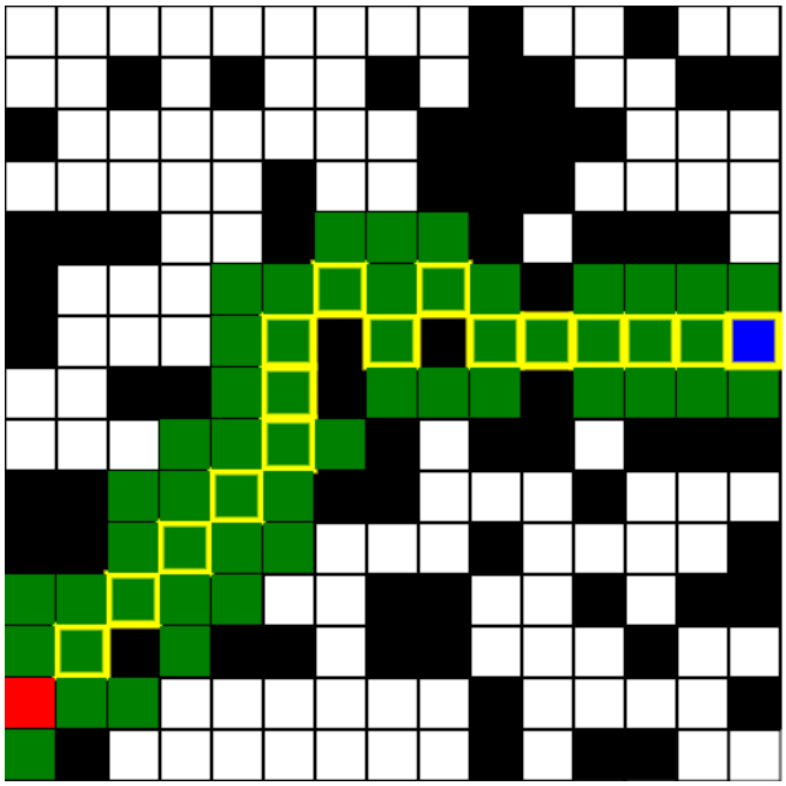
Final F(x): 0
Final G(x): 0
Final H(x): 0



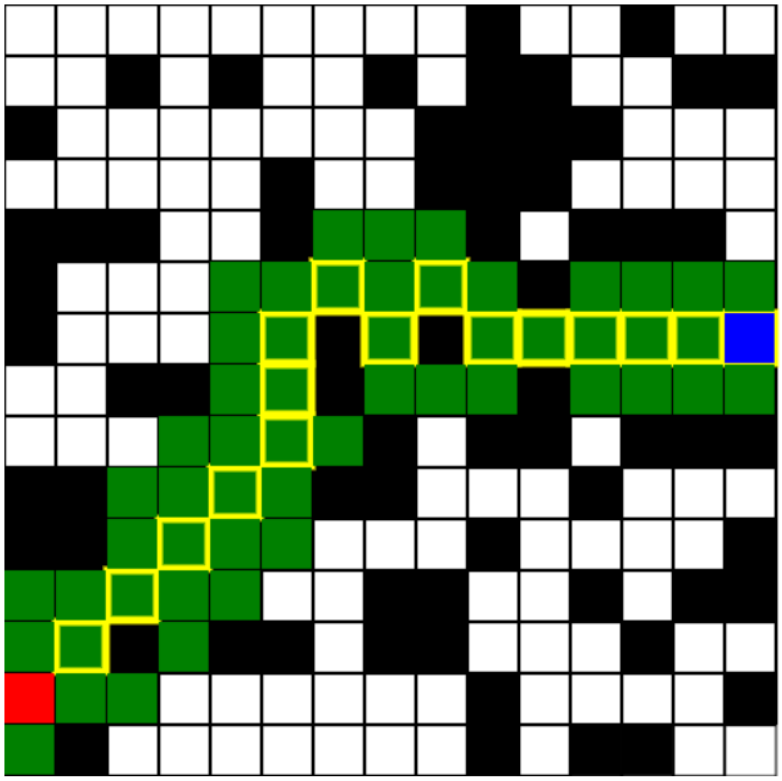
Previous F(x): 204
Previous G(x): 14
Previous H(x): 190
Final F(x): 198
Final G(x): 28
Final H(x): 170



Previous F(x): 180
Previous G(x): 90
Previous H(x): 90
Final F(x): 194
Final G(x): 104
Final H(x): 90

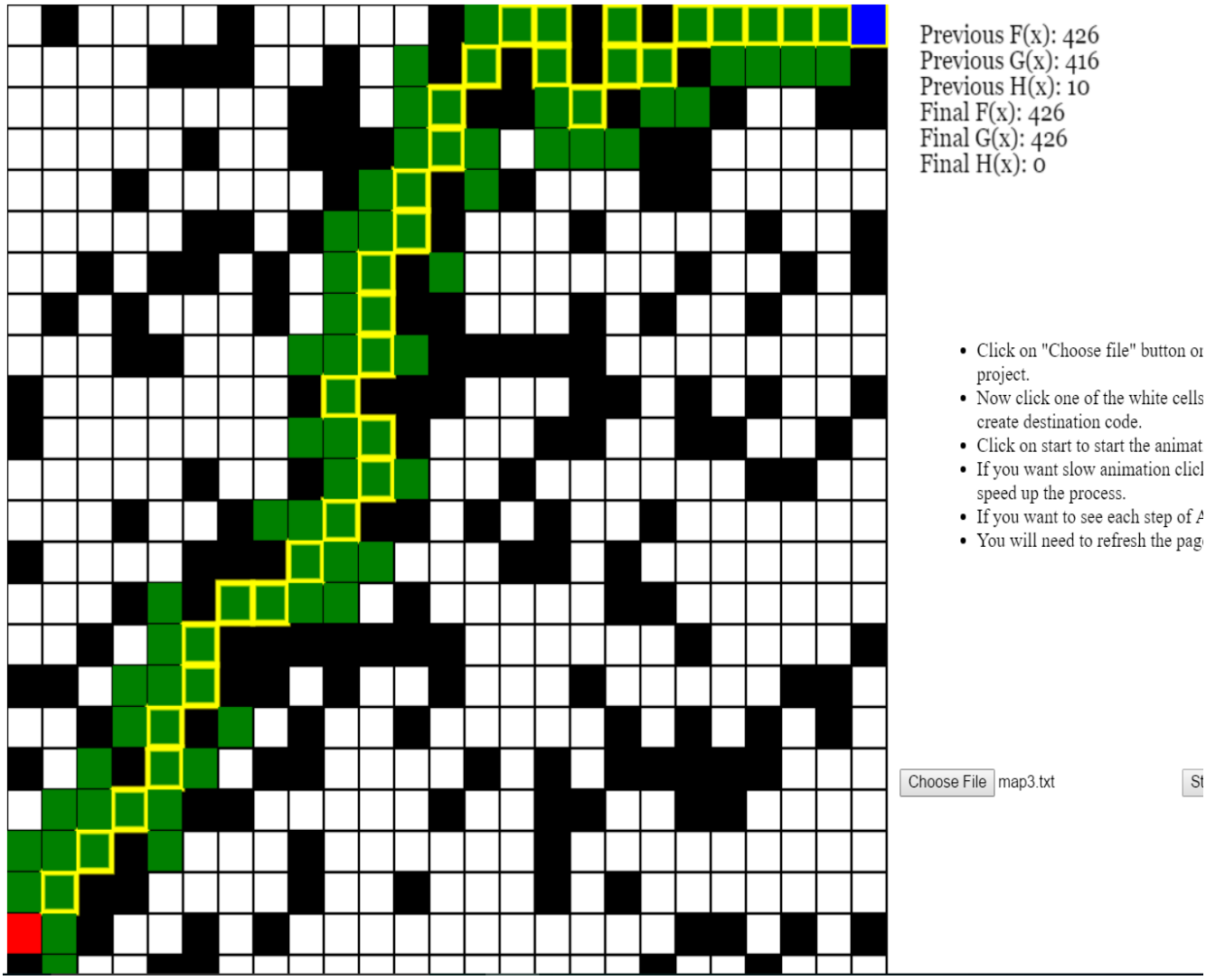


Previous F(x): 196
Previous G(x): 176
Previous H(x): 20
Final F(x): 196
Final G(x): 186
Final H(x): 10



Previous F(x): 196
Previous G(x): 186
Previous H(x): 10
Final F(x): 196
Final G(x): 196
Final H(x): 0

3) 25x25 grid – More than 500 cells



Results:

The algorithms works for grid of more than 500 cells. The algorithm finds path almost all the time but there were some cases where the algorithm didn't find the path or chose a longer path.