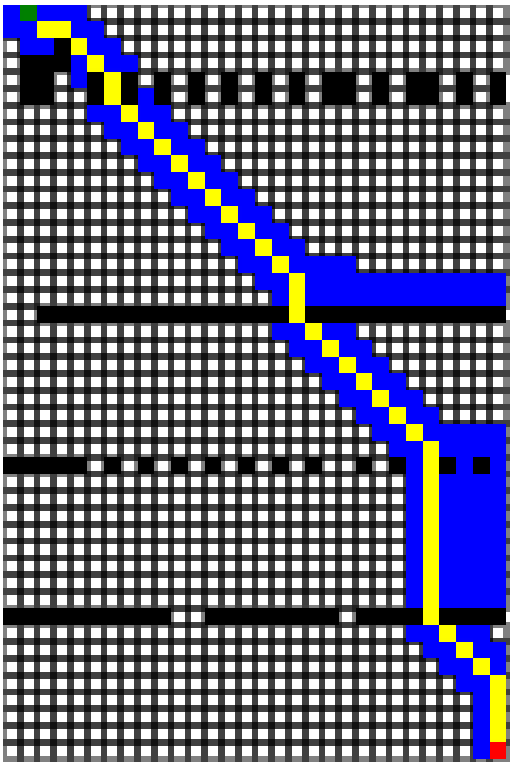
Andriy Yavorskyy

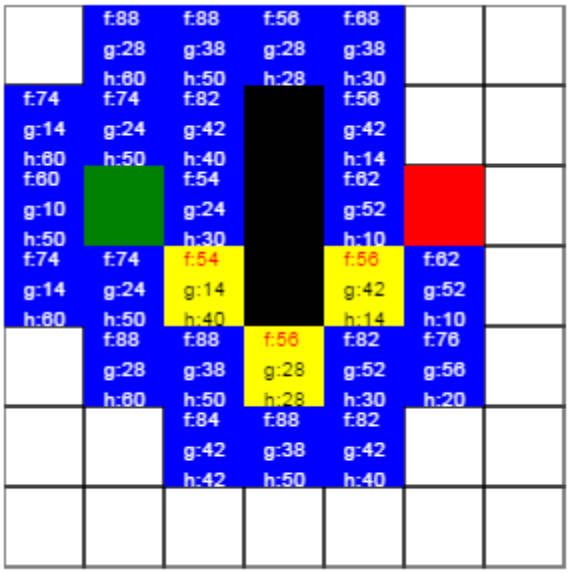
I implemented the a* search algorithm. The blue tiles are the neighbors while searching. The yellow tiles at the end is the final path. Black tiles are obstacles.

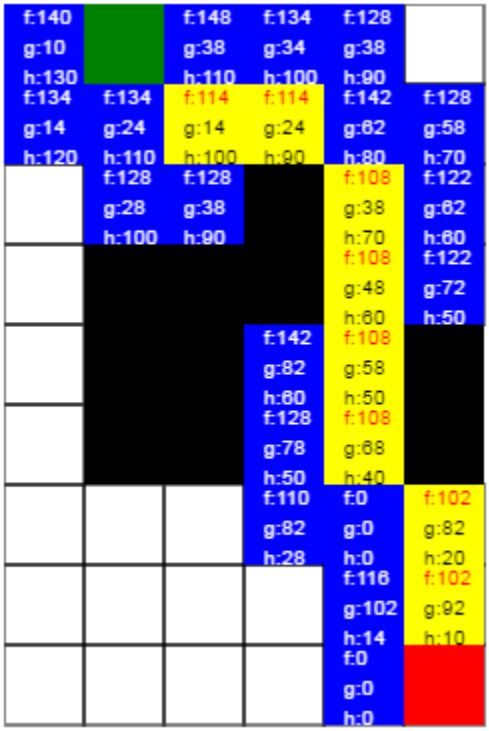
First all the neighbor tiles are shown in blue all the paths are drawn. Then the yellow correct path is shown. Once the grid length is greater than 40, the tile size gets much smaller (5px).

When you click faster/slower it changes the speed for the next run not the current one. I used Manhattan distance to get the heuristics. setInterval method was used for the timing of the animation. Below are some screenshots of the final search.

The algorithm is very fast with larger data sets it will take longer make sure to bump up animation speed.




f:88 g:28 h:60
f:88 g:38 h:50
f:56 g:28 h:28
f:68 g:38 h:30
f:74 g:14 h:60
f:74 g:24 h:50
f:82 g:42 h:40
f:56 g:42 h:14
f:60 g:10 h:50
f:54 g:24 h:30
f:62 g:52 h:10
f:74 g:14 h:60
f:74 g:24 h:50
f:54 g:14 h:40
f:56 g:42 h:14
f:62 g:52 h:10
f:88 g:28 h:60
f:88 g:38 h:50
f:56 g:28 h:28
f:82 g:52 h:30
f:76 g:56 h:20
f:84 g:42 h:42
f:88 g:38 h:50
f:82 g:42 h:40


f:140 g:10 h:130
f:148 g:38 h:110
f:134 g:34 h:100
f:128 g:38 h:90
f:134 g:14 h:120
f:134 g:24 h:110
f:114 g:14 h:100
f:114 g:24 h:90
f:142 g:62 h:80
f:128 g:58 h:70
f:128 g:28 h:100
f:128 g:38 h:90
f:108 g:38 h:70
f:122 g:62 h:60
f:108 g:48 h:60
f:122 g:72 h:50
f:142 g:82 h:60
f:128 g:78 h:50
f:108 g:58 h:50
f:108 g:68 h:40
f:110 g:82 h:28
f:0 g:0 h:0
f:102 g:82 h:20
f:116 g:102 h:14
f:102 g:92 h:10
f:0 g:0 h:0