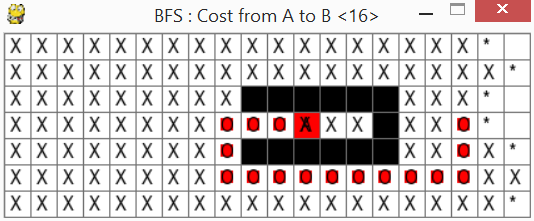
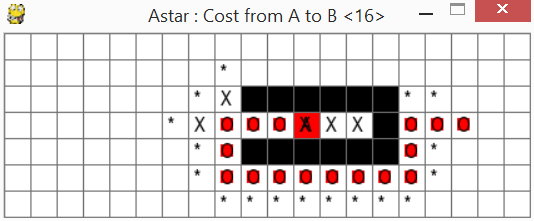
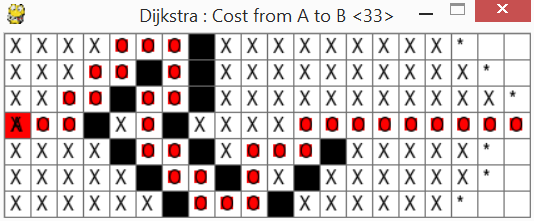
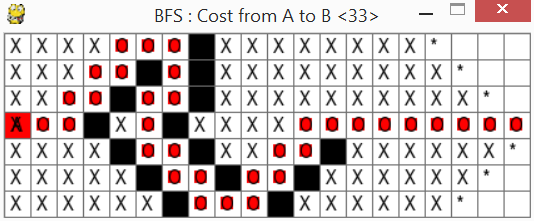
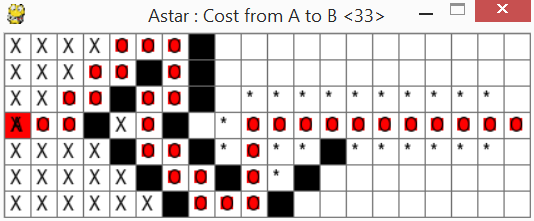
Level 1.1



For this level all the three algorithms have found different paths, but they all have the same cost so no path can be said to be more efficient than another. However the A\* algorithm returns far fewer open and closed nodes compared to BFS and Dijkstra's who open and close nearly all the nodes. Dijkstra has a slightly better performance than BFS that evaluates three extra nodes.

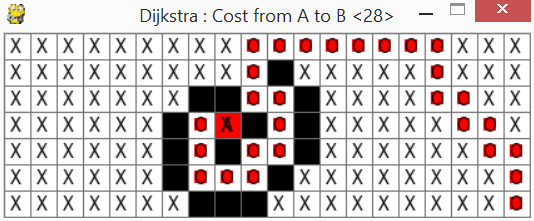
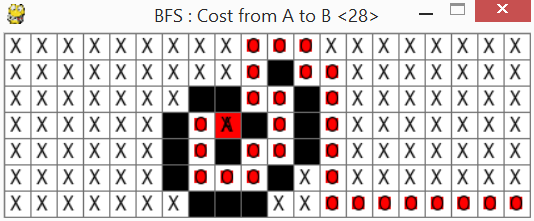
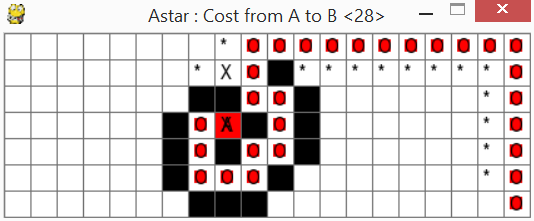
Level1.2



Again the algorithms find slightly different paths with the exactly same cost.

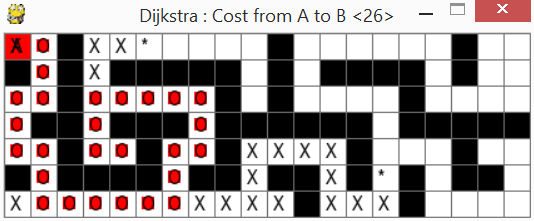
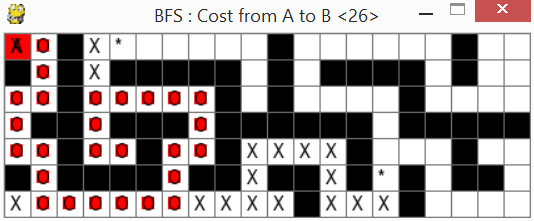
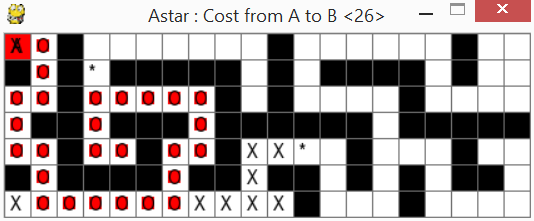
A\* opens and closes far less nodes than the other two algorithms with BFS beating Dijkstra's ever so slightly by 2 nodes.

Level1.3



The algorithms produce different paths for the same cost. A\* outperforms BFS and Dijkstra's by only having 14 nodes open and one closed while BFS and Dijkstra's close all the nodes.

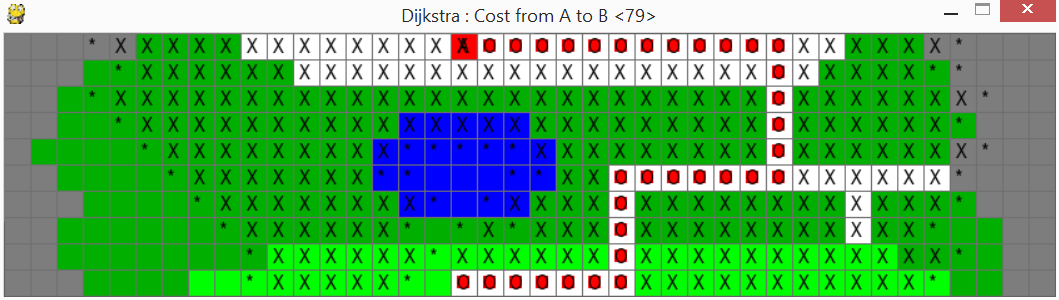
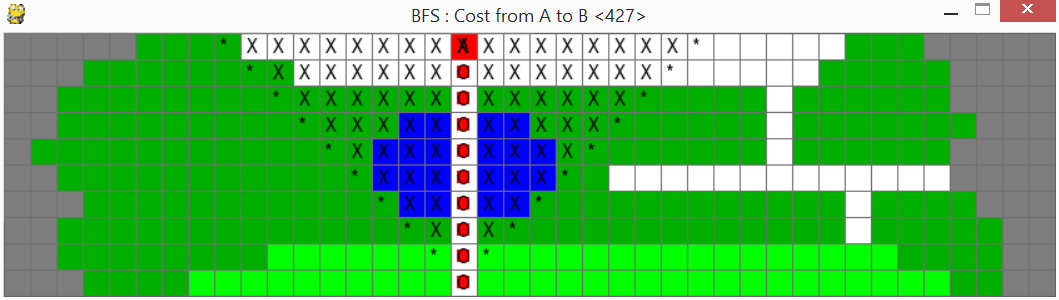
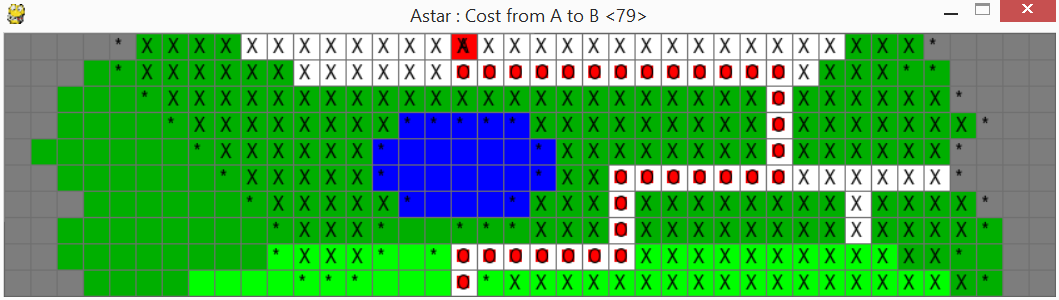
Level1.4



The algorithms have all found exactly the same path. The A\* algorithm opens and closes less nodes.

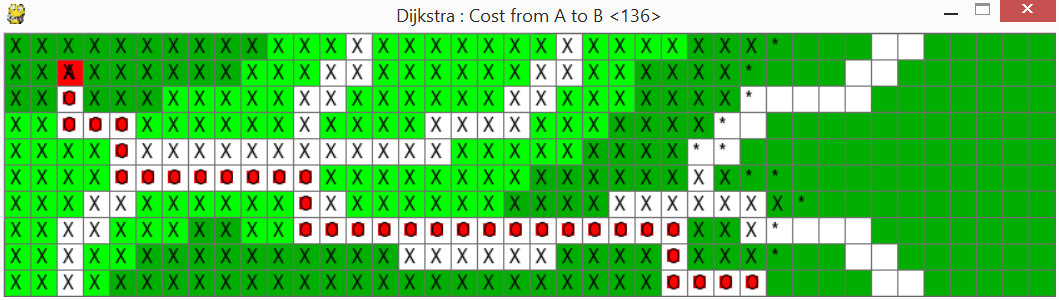
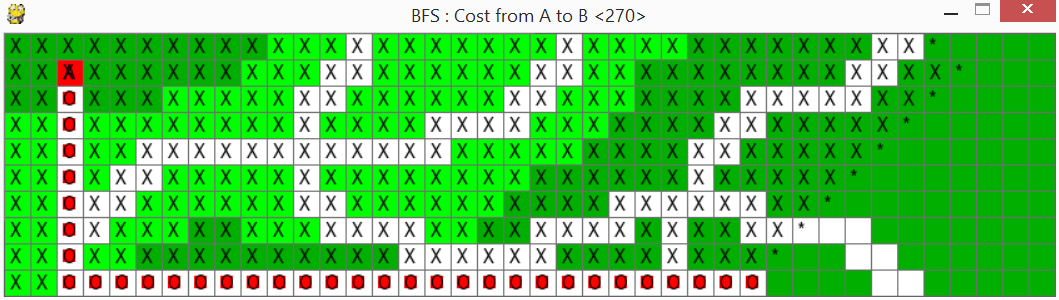
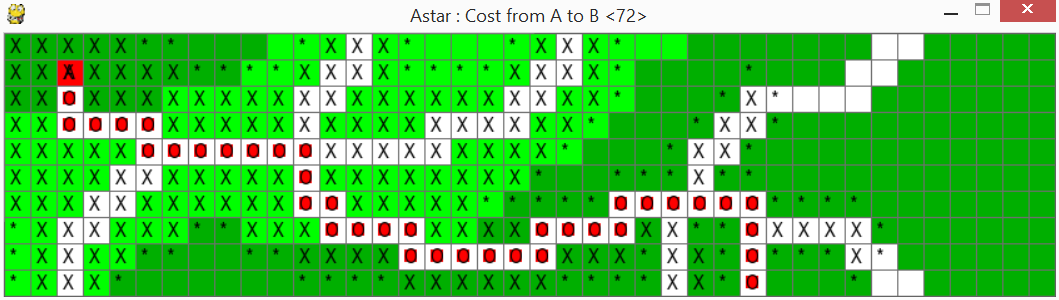
The difference in amount of nodes open and closed is not so large for this level.

Level2.1



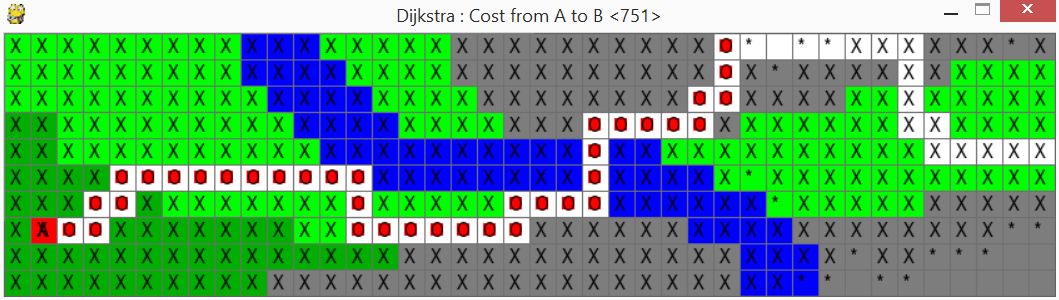
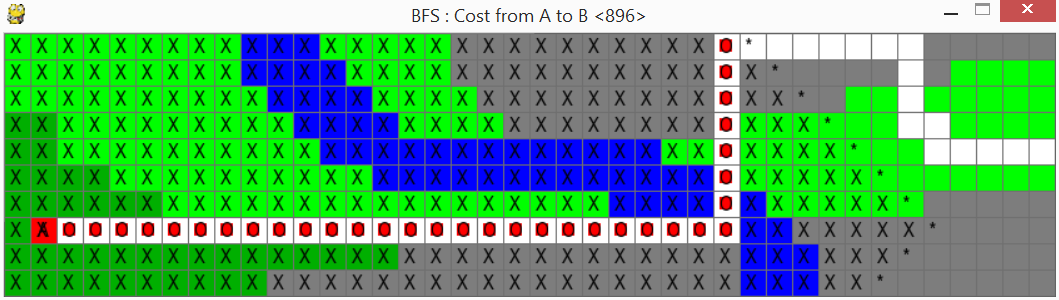
When we add cost differences to the nodes we can see that A\* and Dijkstra outperform BFS and find the same cost for the path. A\*star does slightly better by opening and closing fewer nodes than Dijkstra. BFS opens and closes the least nodes but has a cost of 427 compared to 79 that A\* and Dijkstra's found.

Level2.2



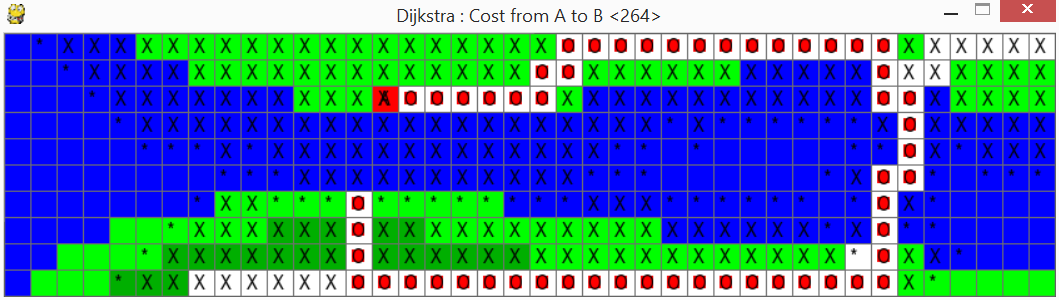
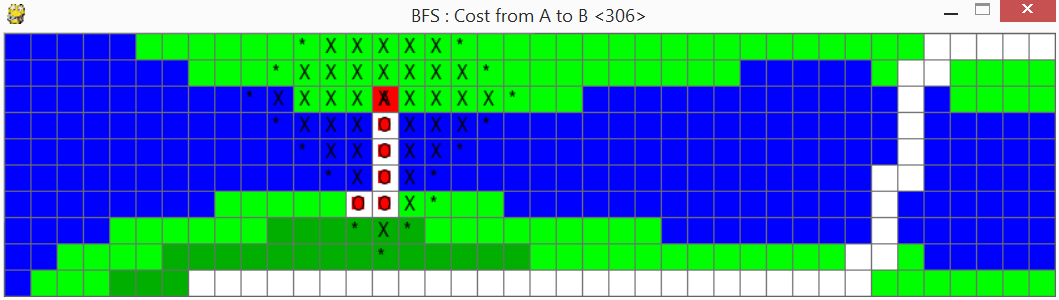
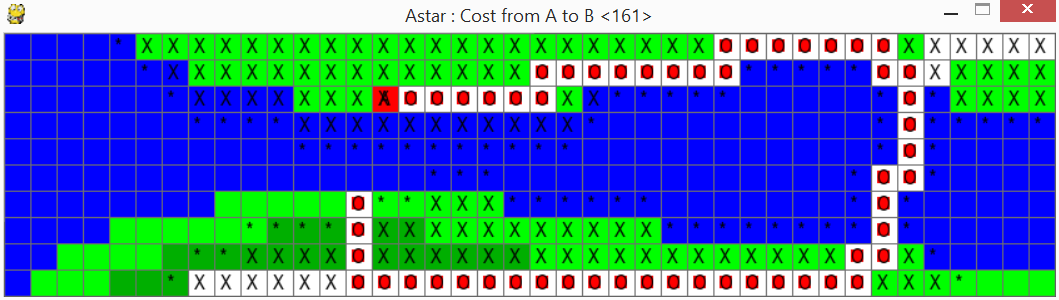
A\*star is the clear winner with the cheapest path of 72, Dijkstra's is second with 136, and BFS last with a massive 270. Dijkstra's is very close to the cheapest path, but as it only focuses on the next cheapest move it manages to miss the road next to its path that it would cost a lot less to follow. BFS and Dijkstra's open and close more nodes than A\*.

Level2.3



For this level the algorithms have made three different paths with different costs. A\* has the cheapest path with a cost of 596, Dijkstra has 751, and BFS has 896. Dijkstra's open and closes the most nodes with A\* and BFS being close but at a 300 cost difference.

Level2.4



A\* has the cheapest cost with 161, Dijkstra's is 264, and BFS path cost is 306. BFS opens and closes the least amount of nodes but has the highest cost. I notice here that Dijsktra's makes a very odd choice that makes me fear that there might be an unknown feature in my code.