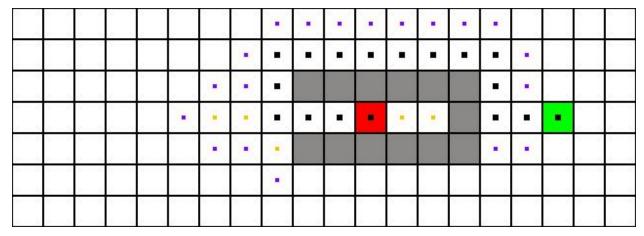
Assignment 3

TDT4136 - Introduction to artificial intelligence Kristian Skog & Kenneh Westli

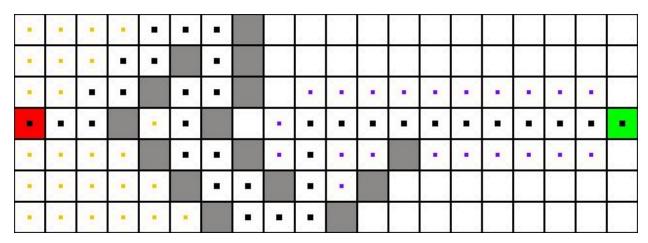
Problem A: Pathfinding in 2D Games

Subproblem A.1: Grids with Obstacles

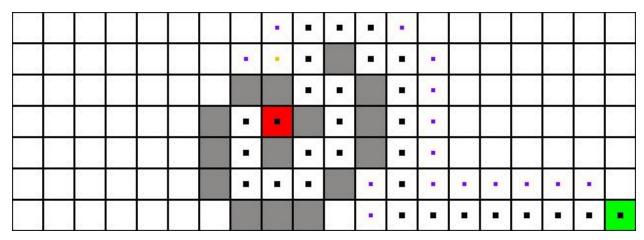
The figures below show the path calculated by the A* algorithm. The Open nodes are represented by purple dots while Closed nodes are represented by orange dots.



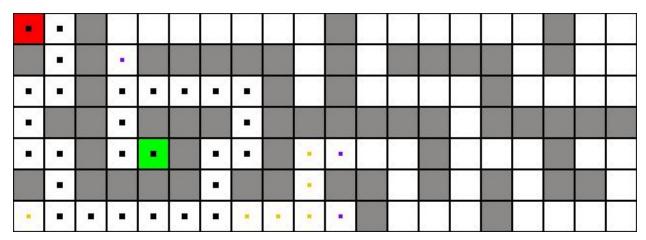
Board 1.1



Board 1.2



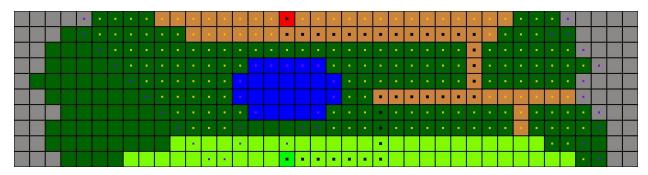
Board 1.3



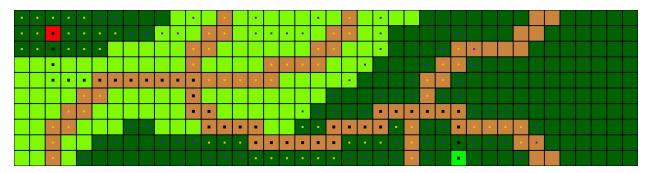
Board 1.4

Subproblem A.2: Grids with Different Cell Costs

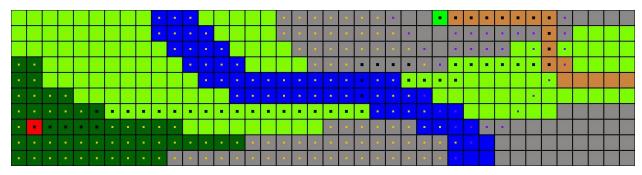
Here, the implementation is modified to being able to parse weighted boards and does also accurately handle cost calculations of the weighted nodes.



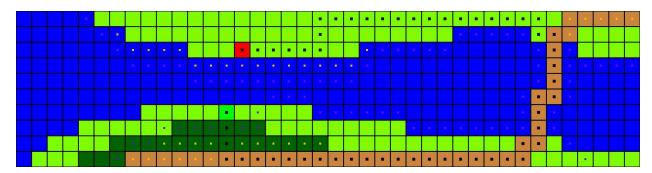
Board 2.1



Board 2.2



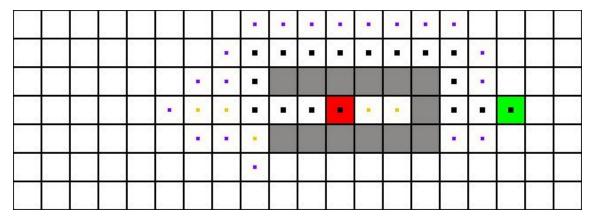
Board 2.3



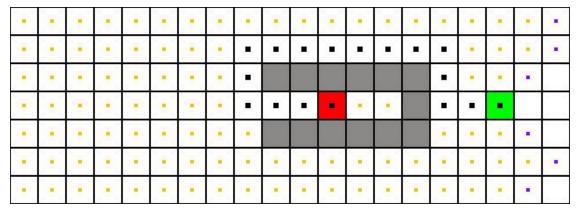
Board 2.4

Subproblem A.3: Comparison with BFS and Dijkstra's Algorithm

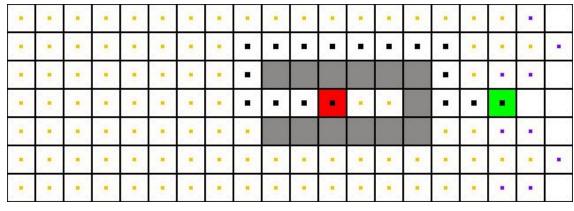
Board 1.1 All solutions find the optimal solution, A* checks far less nodes in order to find it.



Board 1.1 A*



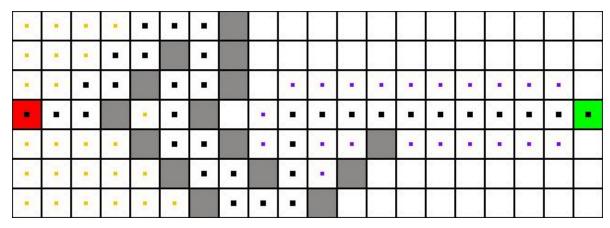
Board 1.1 BFS



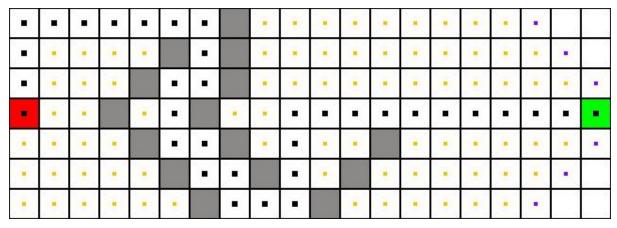
Board 1.1 Dijkstra

Board 1.2

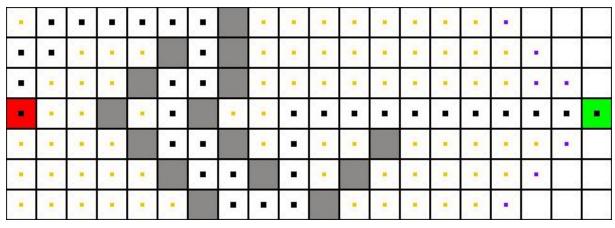
Also here all algorithms finds the shortest path, but again A* uses less nodes and is more efficient.



Board 1.2 A*

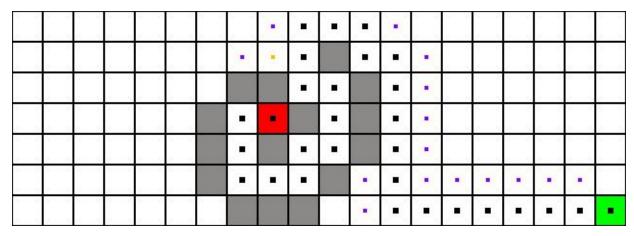


Board 1.2 BFS

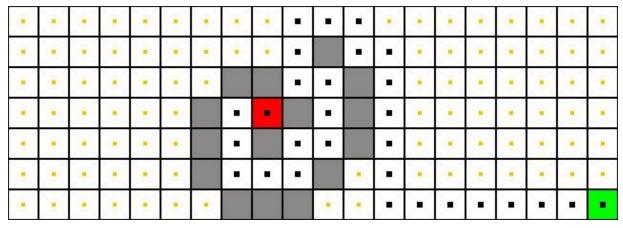


Board 1.2 Dijkstra

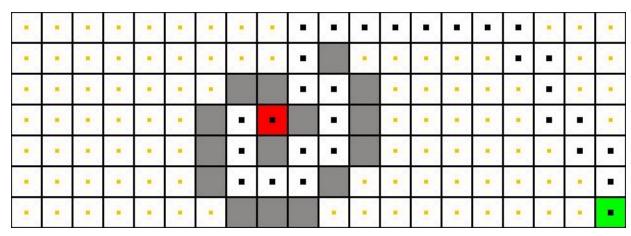
Board 1.3 Here A* is again far more efficient.



Board 1.3 A*

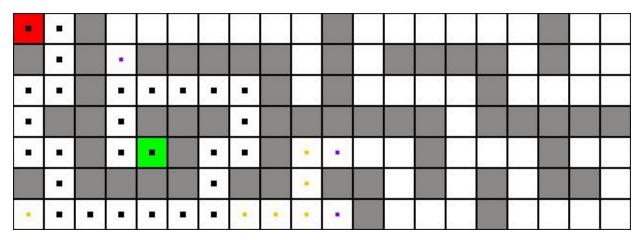


Board 1.3 BFS

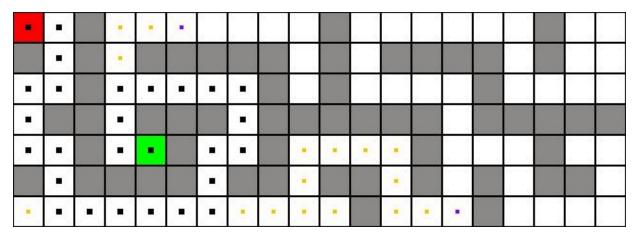


Board 1.3 Dijkstra

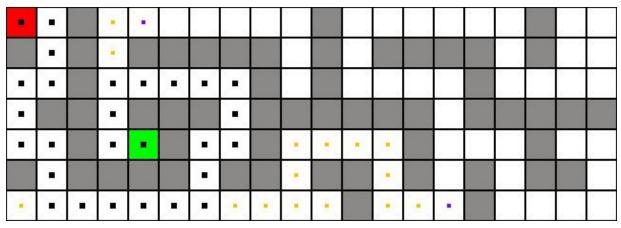
Board 1.4
A* marginally most efficient.



Board 1.4 A*

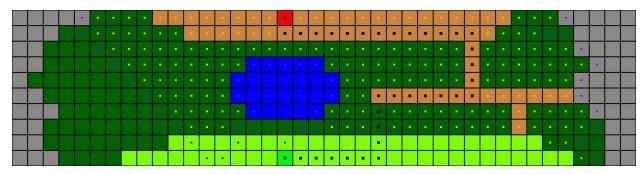


Board 1.4 BFS

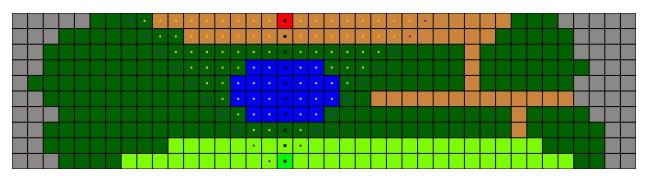


Board 1.4 Dijkstra

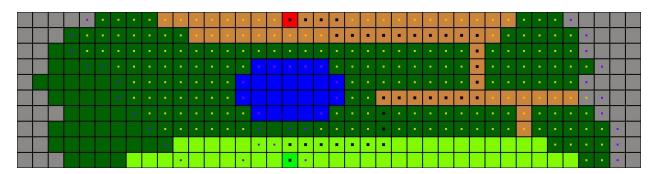
A* and Dijkstra both find optimal solution, A* slighly more efficient. BFS checks few nodes and finds a very unoptimal solution.



Board 2.1 A*

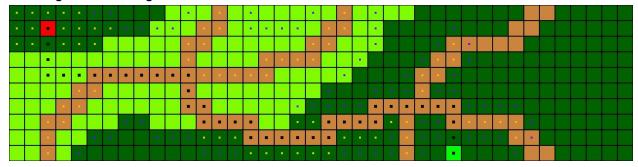


Board 2.1 BFS

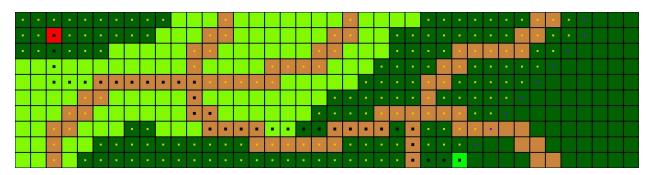


Board 2.1 Dijkstra

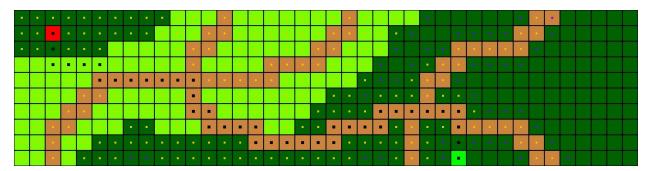
Here both A* and Dijkstra does well, while BFS uses more nodes, but A* avoids processing low cell weight nodes weights.



Board 2.2 A*

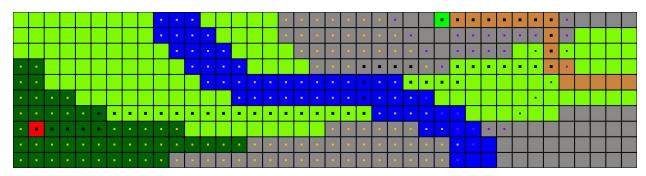


Board 2.2 BFS

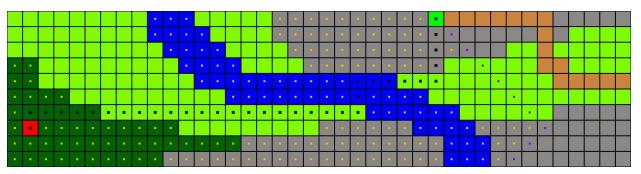


Board 2.2 Dijkstra

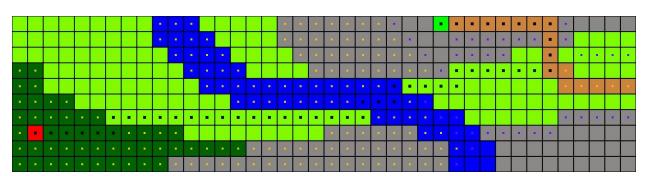
A* and dijkstra finds equally optimal paths, A* checks slighly less nodes. BFS discovers a very unoptimal path, seemingly ignoring cost.



Board 2.3 A*

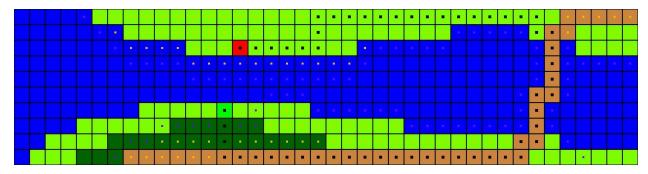


Board 2.3 BFS

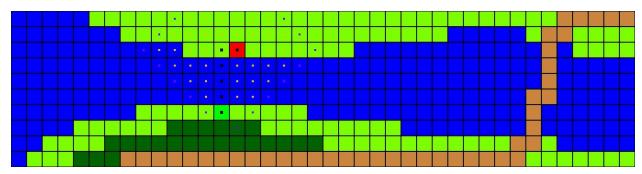


Board 2.3 Dijkstra

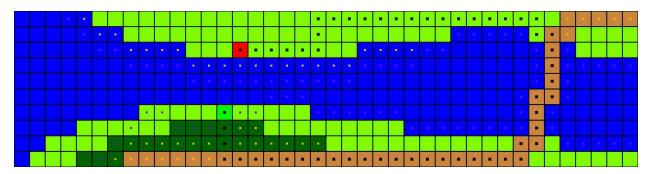
Here A* and Dijkstra finds the same path, although A* processes fewer nodes. BFS discovers a very unoptimal path, seemingly ignoring cost.



Board 2.4 A*



Board 2.4 BFS



Board 2.4 Dijkstra