AI search methods for sliding picture puzzle game

In this assignment I had to do two AI search methods: the breadth first search and the depth first search. Search algorithms in Artificial Intelligence is very important, these algorithms are the pillar of systematic exploration of alternatives. The goal of search algorithms in Artificial Intelligence is to find the path from the start of the state towards the end goal state.

Implementation

Breadth First Search

The Breadth first search uses levels to explore a problem space. The way that the breadth first search is implemented in this practical assignment is as follows: There is one data structure that is a Queue which is of data type String. A queue is used in this search for its dynamic of FIFO (First In First Out). This queue is used to place the current board state as the first value.

Inside of a while loop, the first value of the queue is dequeued and placed in a variable for testing. The state is tested against the goal state, which is 123456780. If the current state is not equal to the goal state, then the current state’s children will be generated and added to the rightmost end of the queue.

Two methods are used to simulate the actual movement of the board state. The first method (isTransitionValid) is used to see if the blank space can go up, down, left or right without going of the grid. The second method (swap) is used to move the blank space in the position that is most applicable, determined by the previous method (isTransitionValid). The method (swap) then returns the new current state, which is then placed inside the queue. This is a iterative process.

If the current state matches the goal state, then the while loop is terminated and the user is presented with a found message and also how many moves it took to complete the search algorithm and solve the puzzle.

Depth First Search

The Depth first search uses a recursive algorithm which implements the idea of backtracking. The Depth first search is just like the breadth first search but the key difference is that it uses a Stack instead of a Queue.

In this search there is still a while loop to pops the first value of the stack and checks it against the goal state. If it is not correct then the current state’s children will be generated and added to the leftmost end of the stack.

The same two methods are used. One to check if the blank space can move in one of the directions and the second method is used to move the blank space in the position that is most applicable. This new state that is generated by the swap method is then placed inside the stack.

If the new current state matches the goal state, then the loop will be terminated and the user is presented with a found message and how many moves it took to solve the puzzle

Decisions that impacted the design of the game

In the first version of the game, the user had the choice to pic a picture and the shuffle it around and try to solve it. With the new version the game, two extra buttons were placed in the information block, with the caption “Solve by breadth” and “Solve by depth”. The user can now also choose a .csv file with a specific initial board state. This board state will be used to either let the user play the game manually or choose to solve it with the different search algorithms. Extra messages will also be displayed to show information about the two searches.

Experimental Analysis

The Breadth first search examines all the nodes in the search space. Using this search algorithm we are guaranteed to find the optimal solution if there is a solution. A big disadvantage of the breadth first search is that it uses a lot of memory, to hold the search spaces.

The Depth first search examines the entire subtree before examining the sibling nodes. This search is not always applicable to all search spaces. It creates new successors the entire time because of its recursive backtracking.

For this analysis I used the same initial board state for both algorithms. The algorithm with the least amount of moves is the best algorithm.

The results of both searches shows that the Breadth first search is the best algorithm because it gives exponential results for everything. The breadth first search is considered the optimal search algorithm for simple and uncomplicated problems where the solution path is known and short. This is also the best algorithm when the quickest path is necessary.

The depth first search searches the entire space or branch and if the solution is not in the branch, it may not be found. The depth first search is not the most optimal search for a puzzle game because it is much slower because of the wider search space.