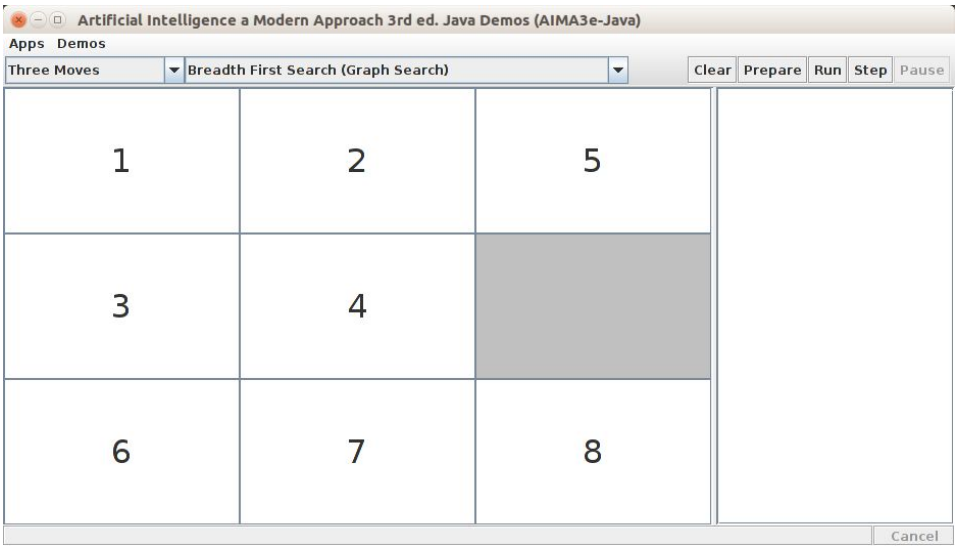
M. Kaan Tasbas
Z23334942

CAP4630 Homework 5 Programming Assignment

1. Programming Preparation Task
    a. 8-Puzzle
        i.



ii. The table below shows certain statistics from the aima-java demonstration of the 8-puzzle game. As the data shows, the Breadth First and AStar searches both find an optimal solution. The main difference between the two is the number of nodes expanded and the maximum size of the queue. These values are much larger for the inferior Breadth First search algorithm. Greedy Best First search is the worst among the three with a path cost of 91, the highest number of nodes expanded, and the biggest queue size.

	Breadth First	Greedy Best First*	AStar*
pathCost	9.0	91.0	9.0
nodesExpanded	288	605	25
maxQueue	206	392	21

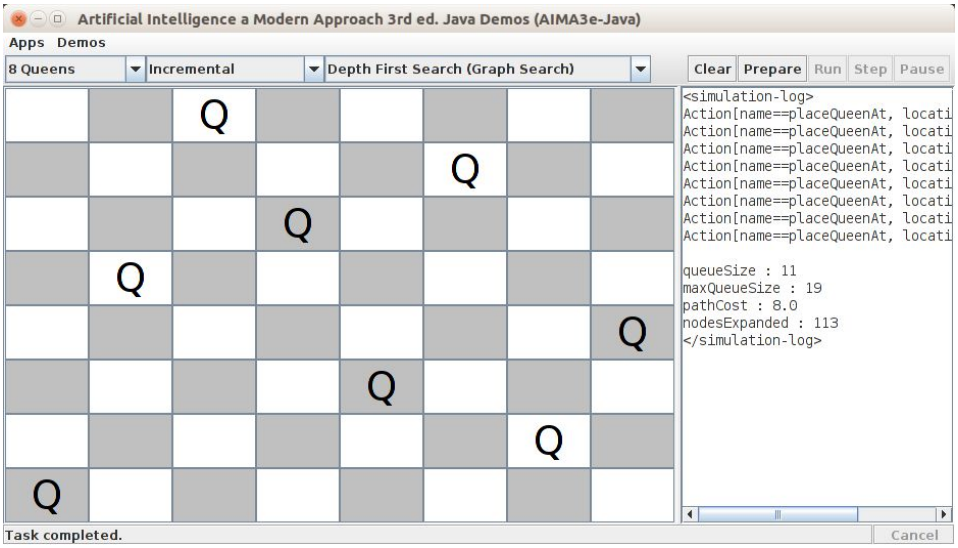
*h(n) = misplaced tile heuristic

iii. Based on the data above, AStar search is the most efficient method as it has the lowest values for pathCost, nodesExpanded, and maxQueue. The opposite is true for Greedy Best First search so this is the least efficient method.

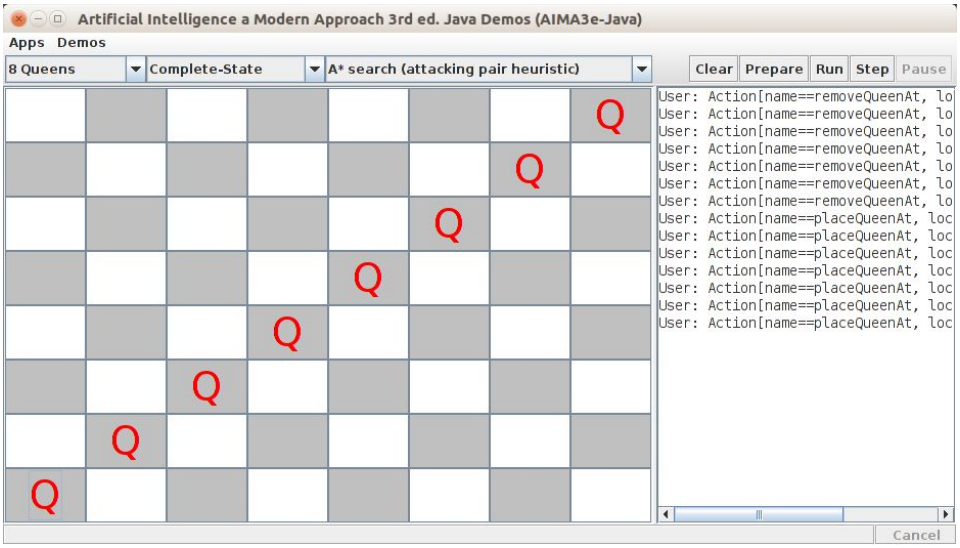
M. Kaan Tasbas
Z23334942

b. N-Queens

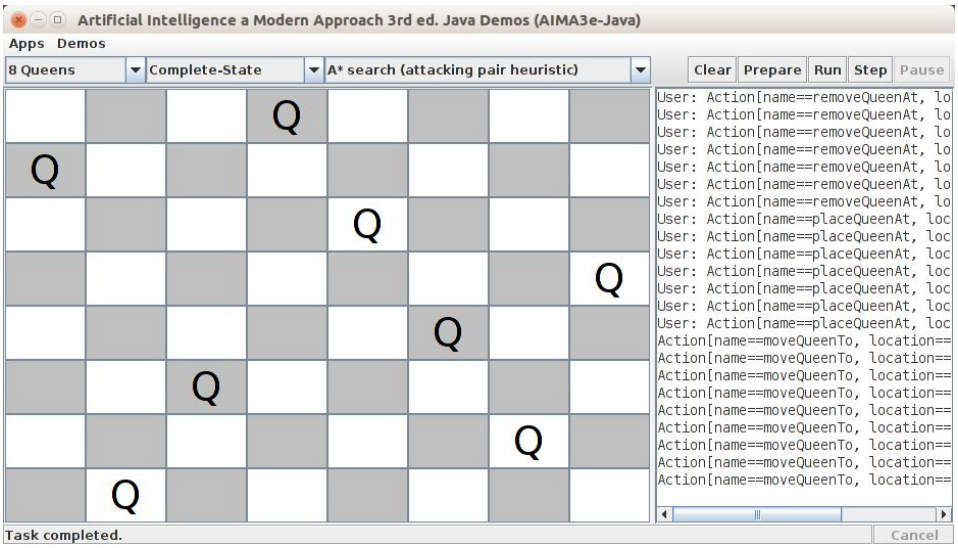
i.



ii.



iii.



f(N) = 22 -> 16 -> 11 -> 9 -> 5 -> 3 -> 2 -> 2 -> 0

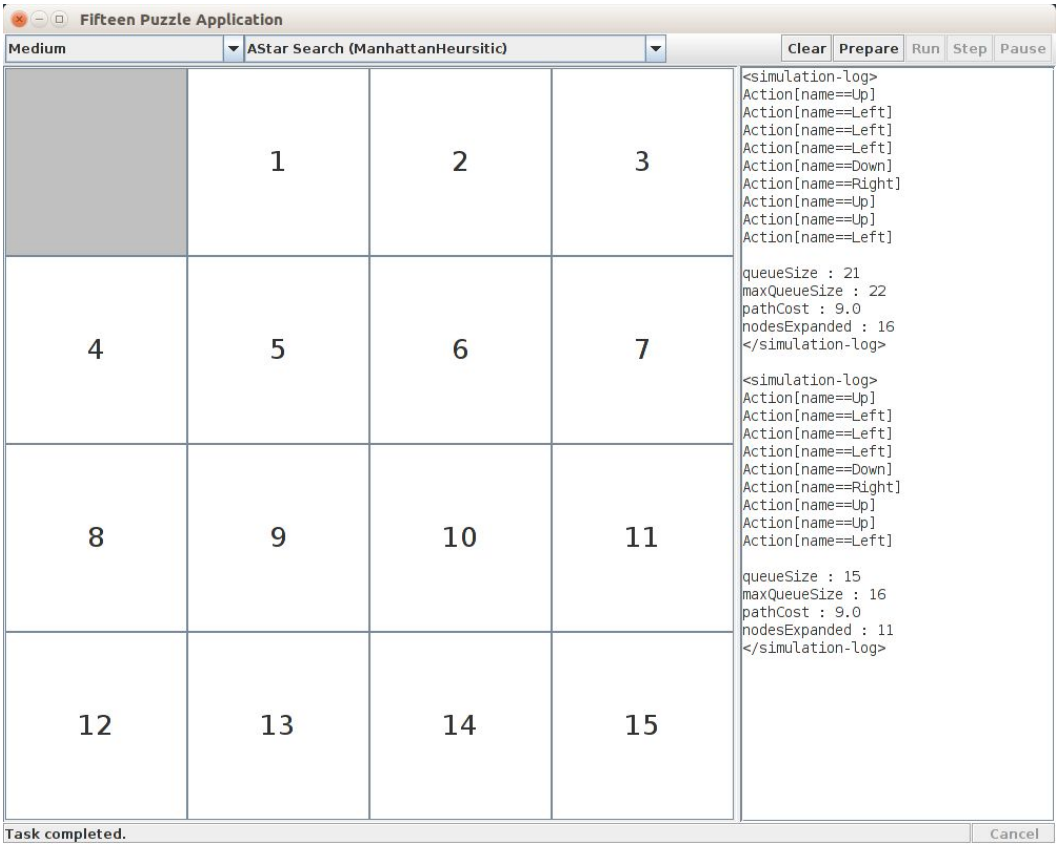
iv. The Hill Climbing search is unable to find a solution from the given initial state. The algorithm only makes a move if it “improves” on the current state and therefore gets stuck at a local minimum as it attempts to minimize the number of attacking pairs. Introducing randomness to the initial state would improve the search so that it may eventually find a solution from a random set of initial states.

2. Programming Tasks
    a. The misplaced number of tiles heuristic, implemented as MisplacedTileHeuristic, counts the number of tiles in the incorrect position when estimating the path cost. The manhattan distance heuristic, implemented as ManhattanHeuristic adds up the x and y distance the tile is from its correct position. Both of these heuristics are consistent, and therefore admissible. We can also say that the manhattan heuristic is more accurate than the misplaced tile heauristic because for any node N ManhattanHeuristic(N) >= MisplacedTileHeuristic(N).
    The ManhattanHeuristicFunction class uses a for loop to iterate over all tiles and a switch statement to calculate the distance of the selected tile. A running total is kept and returned.
    The MisplacedTileHeuristicFunction class uses a set of if statements to check whether each board position has the correct tile. Each incorrect tile adds 1, and the empty tile is properly ignored.

    b. aima.core.environment.fifteenpuzzle.GoalReachable uses the permutation inversion technique to determine whether the goal state is reachable from the current (initial) state. This technique adds up the number of inversions on each board, calculated (inversions % 2), and compares the values. If the two boards have similar inversion % 2 values, the states are reachable and so the function returns true.

M. Kaan Tasbas
Z23334942

c. The following GUI screenshot shows the values for the misplaced tile heuristic (first simulation log) and the manhattan distance heuristic (second simulation log):

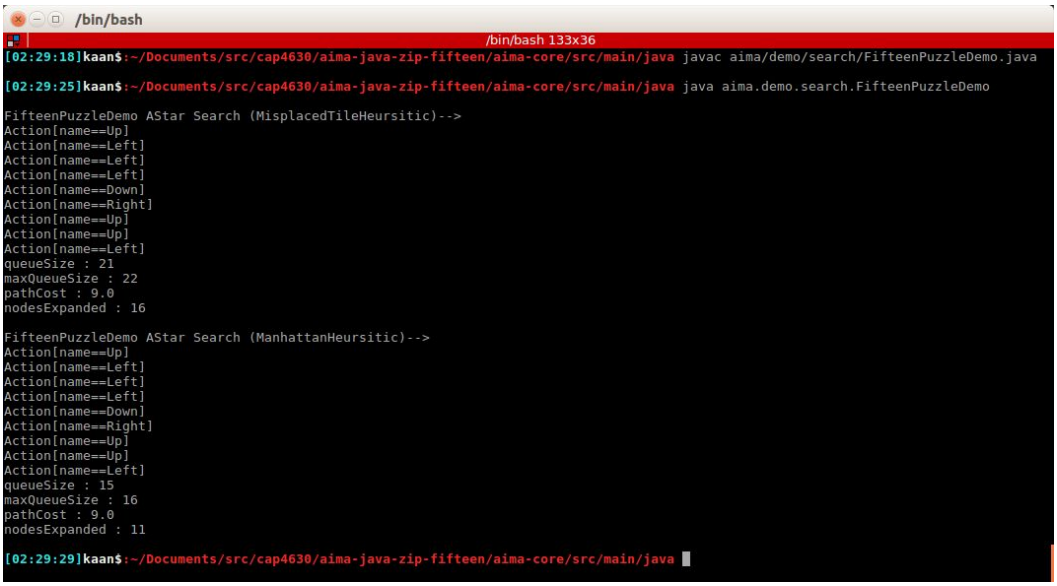


The statistics are summarized in the following table:

	MisplacedTile	Manhattan
pathCost	9.0	9.0
nodesExpanded	16	11
maxQueue	22	16

As the table shows, the manhattan distance heuristic has superior performance. Both are able to find the optimal path to the goal state, but the misplaced tile heuristic expands more nodes and has a larger queue. As the problem complexity grows, the manhattan heuristic finds a solution more efficiently

d. The following is the output when run from a terminal:



The initial state is { 1, 5, 2, 3, 9, 6, 7, 11, 4, 8, 10, 0, 12, 13, 14, 15 }.
The goal state is { 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15 }.
The movement of tiles is shown in the terminal output above.

e. The GUI for the 15-puzzle has two drop down menus: one for selection of the board (Three Moves, Medium, Extreme, and Random) and another to select the search method (AStar Search (MisplacedTileHeuristic), and AStar Search (ManhattanHeuristic). Sample images of the GUI being used are below:

M. Kaan Tasbas
Z23334942