AI Lab 2

Karl-Johan Djervbrant

December 5, 2019

Task 1 Path Planning

1.1 a) Unified Search Algorithms

In this task we were supposed to implement three different types of search algorithms.

1. Random Agent
2. Breadth First Agent
3. Depth First Agent

The random agent works by simply adding a random weight to the node between 1 and 10, then sort the list of all nodes by the weight and remove the first node in the list.
Breadth First and Depth first are very similar in how they are implemented, it's simply a difference in how one removes a node from the list. In Depth first, the last element in the list is removed first whereas in Breadth first you remove the first element.

1.1.1 Performance

On a set of 100 runs this is the mean for each algorithm.

1.2 a) Informed Search Algorithms

1.3 b) USA vs ISA on map with obstacle

1.4 b) A* with heuristic

1.5 What I learned in Task 1

Task 2 Poker Bidding

In this task we had to implement three different kinds of search algorithms to beat an opponent player in a game of poker. The implemented algorithms were a Random-, Breadth First- and a Greedy Search algorithm. The key used when sorting the list of states for the Greedy Search algorithm were the `nn_current_hand` which is a measure of how many hands the dealer have handed out in this state. I use this key since we sought to find the state were the agent win more than 100 coins from the opponent in the least amount of hands.

2.1 The implemented agents

2.1.1 Performance

100 Games	Random Search	Breadth First Search	Greedy Search
stack	306.50	400.30	402.10
nNodes	9102.80	135871.28	4209.28
depth	13.24	10.36	10.64
nHands	7.22	2.30	2.10
opponentStack	293.50	199.70	197.90

After playing 100 games of poker we can see that the Greedy Search algorithm performs the best out of the three algorithms, with fewest number of expanded nodes. It also wins over the opponent for most of the times with only two hands. The search depth on the other hand is on average a bit larger compared to the BFS algorithm. Random search performs the worst out of all three, not only does it expand many nodes, it also doesn't win very much.

2.2 What I learned in Task 2

In this task I increased my understanding of reading and understanding code someone else had written, and also the importance of sorting your data on the correct key to increase the performance of a search algorithm.