Title

Authors

2019-11-10

Abstract

Abstract

Contents

1 Laboration report 1

This report discuss the following parts of the AI laboration: Part 1 that is to create a robot that can move and navigate though in an environment and a part 2 that consist of creating a poker game with at least two players. The report includes the following: Part 1 (a-d + bonus), Part 2 (a-e + bonus).

1.1 Part 1 (robot)

The different agents is doing as it is told according to the (a-c) assignments. However the reflex with memory works a little different. In order to confirm that the robot doesnt run into the walls so the critera for bonus part will be uphold additional sensors were introduced. It was discovered during the lab that it is hard to promise that it wont hit a wall unless we have full control over the area around the robot. The greater view the more information and the decision can be based on it to give a better path for the robot. Since the robot it not very smart it doesnt actually know anything about its surrounding except what is in front of it. It can take a lot of time to finish the map and the possibility to stay on one place on the map have a high probability. The current way is to first try to pick the ball if that doesnt work it will start to go random in order to find alternative ways to target, the time it will go random increases the longer the time it takes for it to find a ball. Alternative solutions for would be to store what the robot have done and try to create an internal map by storing data from the sensors. This would make the robot know were and what it is doing and from there navigate though the map. Alternative use an algorithm that make the robot cover the full map, this could however mean that it would ignore were the balls are until it is close enough.

1.2 Part 2 (poker)

The poker game is built upon the different agents. As seen in the project (all parts is included in this, and bonus). The project is structured with the following classes: Agent, Board, Deck, Hand and Simulation and additional agents that inherit the Agent class. The main program is called by creating an object from Simulation. Since all is object we can do multiple agents on one board. By passing the agent object as an index we can store each opponent for the agent and it can make decision based on correlation. For example: The agent, reflexAgentMem2 knows about their opponent, its hand and can from there make a decision. It will take a few rounds before it knows who the opponent it (minimal amount of data to do correlation) and it will check them all. It will always try to make the highest bid in order to get the most in the win pot. According to image we can see that both the agents perform almost equally and the graph is normally distributed if their mean bet is 25 and do 1000 games, and each game contains 50 rounds. Reminder that this check for part d only checks agent 0 (can be modified in main and is therefor assumed that this is the agent we are looking at.

2 Conclusion

Additional improvements could be made to the first part, including algorithm for better search or more sensors to give a better view of the surrounding.

The second part is flexibledynamic and proves that it handles multiple agents of different kindes and is easy to implement new onces without breaking the game. A few part is less dynamic like the performance check of an agent (only handle first if not changed in the Simulation code).