

AI Lab 1

Karl-Johan Djervbrant

November 20, 2019

Task 1

1.1 a) To implement a random agent

The task were to implement a random agent, what I did were to use the `World.execute()` function, with random speed on both wheels and a random execution time. Every 10th simulation second (if the the code does not perform `World.execute()`) it tries to pickup an energy crystal. The performance is not great for this agent.

1.2 b) To implement a fixed agent

This agent follows a pre-programmed course and tries to pickup energy crystals every simulation second. The performance is better than the random agent, but it can only pickup crystals along it's route.

1.3 c) To implement a simple reflex agent

The reflex agent works by reading the energy sensor to get the direction to the closest energy crystals. It then aligns with the closest crystal (`trimDirection()`) and when in range, it picks it up. This performs ok, but if an energy crystal is behind a wall it will just continue driving in to the wall.

1.4 d) To implement an agent with a memory

This agent is much smarter than the previous ones. It follows the basic principles of the reflex agent in which it turns to align with the closest energy crystal and drive to it. But if the robot runs into a wall, it will follow the wall until the distance to energy $d < 0.5$ or $d_{current} + 0.3 < d_{old}$. A wall is defined by:

1. The front sensors tells there's an object in front of it
2. The object doesn't disappear if the robot tries to collect the crystal

If the robot doesn't collect an energy crystal for a set amount of cycles, it will reverse in a random direction and then repeat the whole process again.

1.5 What I learned in Task 1

In this task I learned that simple/trivial rules or definitions for us as humans can be really hard to explain in code, such as how to define what a wall is.

Task 2

The whole task is object oriented where every agent inherits from the parent `PokerPlayer`. To change which agent is playing you simply set either `player1` or `player2` to be the agent. E.g. `player1 = RandomPlayer()` in `Lab1.Task2.Main.py`.

2.1 a) Random agent

This bids a random amount each time, it performs ok with a cash/win ration approximately between \$120-\$160 per round against another random agent.

2.2 b) Fixed agent

This agent is set to bid \$25 each time and it's performance cash/win ration against another Fixed-agent is always \$150 per round, since both bet the same amount. It's a bit more reliable than the Random agent, but it's still 50/50 chance if it beats the other agent.

2.3 c) Build you own environment

As said earlier, I developed the program with OOP in mind where each player class inherits from the parent player. There's also a class for cards, the deck, a hand and the table.

2.4 d) Fixed vs Random agent

The fixed-agent bids \$25 each time and it's performance cash/win ration against a random is approximately between \$140-\$160 per round. And it's a step up from the Random vs Random agents, but it doesn't always win more than the random agent.

2.5 e) Reflex agent

This agent chooses what to bid depending on the value of its hand. If it has a good hand it bets more and bet less if the hand is bad. This performs best out of the other agents, because the player2 will win less if the chance to win for player1 is low since player1 will bet less. If player1 has a good hand, it will bet more and therefore the pot will increase. The cash/win ration is approximately between \$140-\$160 for player1 (reflex) and \$110-\$140 for player2 (random).

2.6 What I learned in Task 2

In this task i didn't learn anything specific with AI, but I got learnt how to inherit classes and how to structure a OOP project from scratch.