Lab 1 - Vacuum Cleaning World
T-622-ARTI - Artificial Intelligence

Eva Ósk Gunnasdóttir (evag18@ru.is)
kt: 050899 2849
Hólmfríður Magnea Hákonardóttir (holmfridurh17@ru.is)
kt: 040396-2179

18. january 2021

1 Introduction

In the early 20th century, most women in the United States did not work outside the home. Now more then hundred years later only 5% of women work inside the home. So how is left to clean the house?

We introduce the **ROOMBA !**, brand new **smart** vacuum cleaner.

In this papper we are going to talk abut the agent we created for our **ROOMBA !**. Discust it's stratergy to clean the house.

Task 1: *(12 points)* Characterise the environment (is it static or dynamic, deterministic or stochastic, . . .) according to all 6 properties mentioned on slide 13 (Agents) or section 2.3.2 in the book.

Solution:

1. Partially observable: Because we only see things we touch, but we can't see how long the wall is or where the corner.
2. Stochastic: Because we don't know where the next step takes the bot, will he hit the wall or find more dirt?
3. Episodic: Since the environment has no influence on the last, the board always just resets the same.
4. Static: The environment dose not change between moves.
5. Discrete: The environment has finite many states. Since the robot can only move one unit forward, turn 90° or suck/not suck up the dirt.
6. Single agent: There is only the vacuum robot

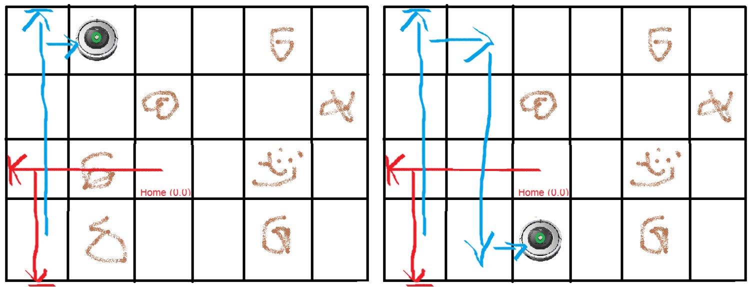
Task 2: *(10 points)* Develop a strategy for the agent such that it fulfills the goal and describe this strategy in a few sentences.

Solution: The Robot starts at its home $(0, 0)$, the center of the universe, and facing forward. His **first task** is to turn on, *very important*. His **second task** is to find the lower left corner. His **third task** is to clean the entire room using the *zik-zak* technic. His **final task** is then to return home, and go to sleep.
Here below is a detailed description of each task, along with explanatory pictures.

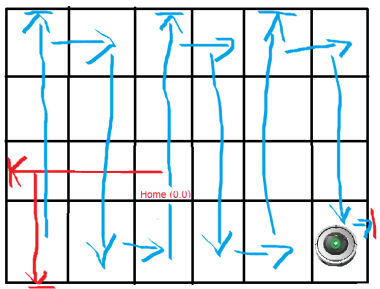
Step 1. turn on.

Step 2. Find the lower left corner.
Turns to the left, moves til he *bumps* in to a wall, then turns left again and moves til he *bumps* then he has found the lower left corner. then he takes two more turns two the left.

Step 3. Clean the entire room. Now he starts to clean every dirt he meets. He moves forward til he bumps to a wall. Turns right and takes one step forward. Then he turns right again and moves til he sees the wall again, turns to the right, moves one step and then turns to the right again.

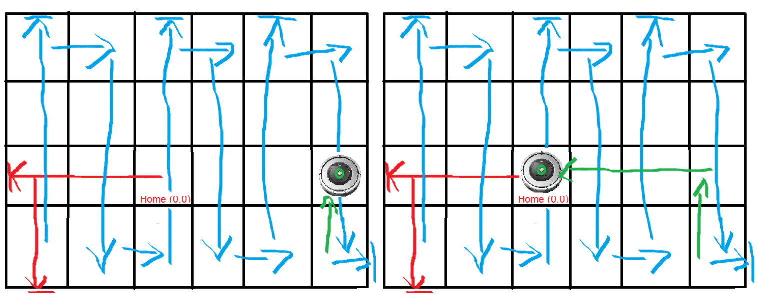


Then he repeats step 3. until he hits a corner again.

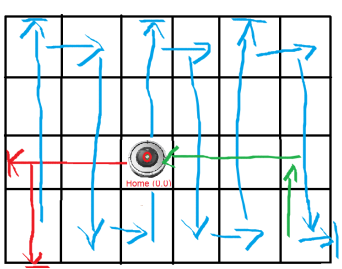


Step 4. go home.

First check if we are above or below our home. If we are below then turn right if not then turn to the left then move til your aligned with your home. Then turn towards your home and move til you are home.



Then turn of.



Task 3: *(60 points)* Implement the missing parts of the vacuum cleaner Java program (see below) such that it encodes your agent function (strategy)

Solution:
You can find the code on GitHub: https://github.com/EfaOsk/VacuumCleaningWorld.
The original agent: lab1/src/MyAgent.java
Optimized agent: lab1/src/OptAgent.java

Task 4: *(10 points)* Test your program with all three provided environments. Test with environments with random in their name can give slightly different results each time, so they should be repeated several times. Record the number of steps it takes to finish each environment.

Solution:

We tested both our agents, both the naive- and the optimized agent in all three environments (see the test below). We see that the Optimized agent is slower then the naive agent, but gives better results (Finnish in less steps and gets higher score).

Vacuum cleaner:

Naive Agent	score	steps	time
test 1	43	63	237 ms
test 2	43	63	190 ms
test 3	43	63	62 ms

Optimize Agent	score	steps	time
test 1	51	55	879 ms
test 2	51	55	791 ms
test 3	51	55	805 ms

Vacuum cleaner random:

Naive Agent	score	steps	time
test 1	43	63	86 ms
test 2	5	77	55 ms
test 3	23	71	27 ms

Optimize Agent	score	steps	time
test 1	49	57	519 ms
test 2	52	54	539 ms
test 3	52	54	562 ms

Vacuum cleaner random big:

Naive Agent	score	steps	time
test 1	31	180	125 ms
test 2	31	180	272 ms
test 3	31	180	27 ms

Optimize Agent	score	steps	time
test 1	61	150	1373 ms
test 2	66	145	1297 ms
test 3	58	153	1390 ms

Task 5: *(5 bonus points)* Optimize the agent to reduce the number of steps it takes to clean an environment. Think about which actions are undesirable in which situation and how to avoid them.

Solution: One easy fix for us to optimize the agent is to limit the bump, we can to that by storing a variable of how long the room is and store just turn before hitting the wall. Another way would be to make the robot always clean in a circle, it might have as many turns as the zic-zac method but it also can be a better solution for the "bad cases"that is where the room is really long and narrow and we are zic-zacing on the shorter side, that would lead to a lot of turns and therefor more moves, also by going in circles we should end somewhere in the middle which should and are therefor ending up closer to home so we could be cutting the return home step in half. The optimized agent can be found in the OptAgent class, after altering the class it now runs in less moves, the test-agent now runs in 54 moves and the smaller random agent is averaging in 60 moves. So it has better over-all performance.

Task 6: *(8 points)* Is your agent rational? Justify your answer.

Solution:

Naive Agent: No, this agent is not rational since it follows a naive algorithm. There are better algorithms out there, and there for the agents performance can be better.

Optimize Agent: I believe that this agent is rational. He makes the best possible decision based on the limited information it has. But if you know where you start and how big the room is, this task, can be finished in fever steps.