Practical session: Lemmings

Introduction

This project is designed to further explore the kinds of stigmergetic & collective behaviors that lie at the foundation of autonomous robotics and dynamic systems approaches to modeling intelligence. The Lemming is designed to model ant brood sorting behaviors. The behavior of the Lemmings is more complex than the Didabots because they can sort elements of their environment relative to their appearances, here, by the color of the blocks (e.g red and blue). In an ant nest larvae are sorted annularly with larger larvae sorted in concentric bands by increasing size around a center. The Lemmings clear blue blocks to the wall, leaving red blocks in the center of the room.

Assignment

Embodiment of the Lemming

You can find an illustration of how a robot lemming can be constructed in Dawson et al., sections 8.9 through 8.29 (see Figure 1).
Note, that because you are simulating a lemming in a 2D environment appose to building a 3D robot, you'll have to be creative in programming the simulation.

Most importantly, the robotic lemming detects blocks using a lower ultrasonic (distance) sensor, walls using an upper ultrasonic sensor, block color using a color sensor and has a gripper to carry the blocks. The simulation environment you are using is 2D (sensors cannot be placed at different heights). Also, no color sensor implemented yet and the robot cannot carry a block. In order to code Lemming-like behavior you will need to modify your simulator to circumvent these problems. This means you need to implement a realistic color sensor (including noise), give blocks different colors, determine a way to distinguish between walls and blocks (see the notes section at the end for hints on how this can be done), and modify the robot to be able to carry blocks.

You can change the shape of the robot into a more suitable form by editing the SVG file `robotbody.svg`. One good program to edit SVGs is Inkscape, available at https://inkscape.org/. Note that depending on the shape, the sensor placement may get a bit tricky. It is OK if a sensor is not quite directly attached on the body boundary. Use your own judgment for what you think is reasonable.

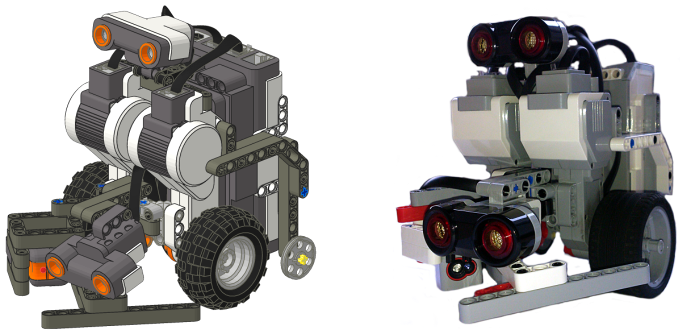

Figure 1, The Lemming using Lego NXT or EV3.

Programming the Lemming

The Lemming avoids obstacles (walls and other Lemmings), detects blocks and discriminates different blocks with a color sensor.
You have to implement the following rules:

- By default, the Lemming should wander around in a slight right curve.
- If it senses a block its subsequent behavior depends on whether a block is in the gripper and the block color.
  - If it carries no block it should drive straight towards it, and thus get the block into the gripper.
  - If it carries a blue block, it keeps wandering and ignores the detected block.
  - If it carries a red block it *turns left* to leave the block.
- If it senses a wall its behavior should depend on whether a block is in the gripper and the block color.
  - If it carries no block it should turn either left or right.
  - If it carries a blue block it should *turn left* to leave the block.
  - If it carries a red block it should *turn right* to keep the block.

Assignment

Once you have built and programmed the Lemming, you are able to observe its behavior. To start out behavioral explorations, first create a checkerboard pattern of 7x7 red and blue blocks and place this pattern in the middle of the arena. First observe the behavior of your single Lemming. Figure 2 shows the starting arena and end result obtained from physical robots.

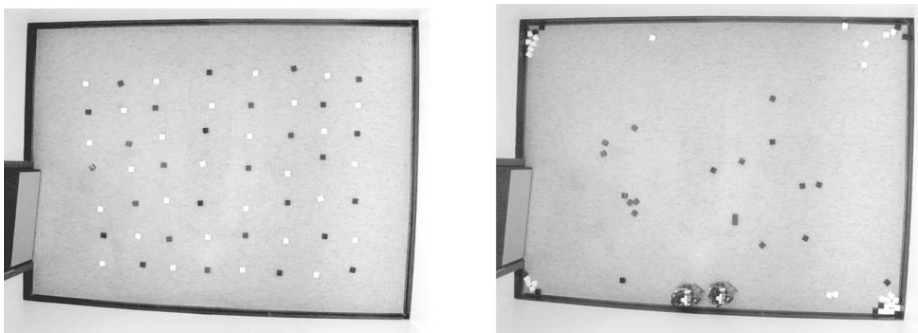

Figure 2, The experimental setup (taken from Dawson et al.).

The goal of this project is to attempt to replicate Dawson et al.'s results and also extend the experiments to a new variable. Dawson et al, provide figures for how quickly on average 1, 2, & 3 Lemmings are able to sort the Lego blocks in their environment. The figures show that as the number of increases the improvement in sorting speed is non-linear (this also means the time to sort decreases non-linearly with robot number). You need to:

1. Determine how quickly the simulated lemming(s) sort the blocks in their environment according to their color. Consider the experiment finished if all the blocks have been sorted (i.e. when all red blocks are in the center of the arena, and all blue blocks are at the wall or in the corners of the arena). To do this you will need to determine how many steps of the simulation have elapsed before the blocks are sorted.

2. Test how this sorting speed is affected by some variable of the simulation. For example, you might test how changing the noise magnitude of the color sensor affects sorting speed. Manipulate this variable in three levels and determine for each level and robot number the sorting speed.

Data Collection

For each simulation allow a maximum of about 50,000 steps to be completed. In case you think this number is too much or too little, you can adjust this (and justify your choice in the report). For each simulation determine the number of steps needed for the blocks to be sorted, this can be computed in the simulator, you also need to decide what to do if the blocks are not all sorted. Run at least 10 simulations per condition. Graph the distribution of sorting speeds and provide means per condition. No need for significance testing. Screenshot an example of the end of the simulation for each experiment condition (but not every run). Note you can run multiple simulations simultaneously with multiple browser windows (next to each other; not only in a tab).

Report

- Write a scientific paper reporting the results of your replication (Experiment 1), and of your investigation of your chosen factor (Experiment 2).
- Include a description and reasoning of the design of your vehicles and sensors, and modifications you made to the simulator (with enough detail for your peers to replicate).
- Report the replication results: Did you exactly reproduce the sorting speeds they found? If not, any hypothesis why not?
- Report the factor you chose to examine further for its role in sorting speed, and why you chose it (i.e., why you think it's important: what's your theory?).
- For experiment 1 and 2 make similar graphs to Dawson et al. Discuss similarities and differences between their results and yours.

The overall goal is to figure out to the best of your ability (and time) why sorting behavior happens. The report should convince us that you thought about this, came up with a good idea to test, tested it thoroughly, and understand the implications of your results.

Notes:

1. You need to be able differentiate between walls and blocks. There are multiple ways to do this, as a start here are two possible ways. Firstly, you could modify the distance sensor to differentially detect walls and blocks (i.e one distance sensor which detects walls, one which detects blocks) this simulates two different sensors at different elevations. Secondly, you could modify the sensor placement to allow sensors to be placed in front of each other, you could then use a combination of multiple color and distance sensors to achieve the required behavior.
2. Only use (realistic) information that is provided by (noisy) sensors. If you create a sensor, make sure it is somewhat physically realistic, and randomly will give noisy information.
   One way to simplify the sensor placement on/at the robot's body might be by ignoring the robot's own body shape.

Deliverables

With your report please include the following in a single zip file named `Lemming report (Group XY).zip`, where `XY` stands for your group number.

- Screenshots of one exemplary simulation result for experiment 1 and experiment 2 (you do not need to screenshot each simulation run).
- All code files, clearly label in your code what needs to be changed in order to switch between experiment 1 and 2 and the different conditions.

References

1. Dawson, M. R. W, Dupuis, B., & Wilson, M., Lemmings, From Bricks to Brains.