Exercise 3

submission date: 30.12.25 (23:59)

Exercise Instructions

Implement **Bug1** and **Bug2** navigation algorithms for the Pi-Puck robot, as described in class.

Goal

Find the **cyan** target and stop.
Display:

- **Blue** LEDs while moving towards the target.
- **Yellow** LEDs while bypassing obstacles.
- **Green** LEDs once the target is reached.

Report

Explain your implementation preferences and report their success in a PDF file

Algorithm description

explain your implementation preference:

1. From which side the robot surrounds the obstacle?
2. Did you use feedback-control? If so, where did you choose to use it (linear/angular/both)? Which parameters did you choose?
3. Add pseudo code of your algorithm.

Success report

were you able to reach your target position? If not, explain why.
Add a screenshot from a bird's-eye perspective of the simulation at the end of the task.
The screenshot must include the simulation clock and the trajectory line.

Workspace Structure

- **`controllers/`** : Your controller code (only modify this).
  You should implement the Bug1 algorithm under the `controller_bug1` class, and the Bug2 algorithm under the `controller_bug2` class.
- **`experiments/`** : `.argos` config files.
  you should try different arenas, as the tests will use more arenas than those provided.
- **`loop_functions/`** : Do not modify.

Available Sensors & Actuators

Sensors:

- `m_pcPositioning` - robot position & orientation
- `m_pcRangefinders` - 8 proximity sensors
- `m_pcCamera` - detects LEDs
- `m_pcSystem` - system time & temperature
- `m_cTargetPosition` - holds the target position vector (not a real argos sensor)

Actuators:

- `m_pcWheels` - wheel control
- `m_pcColoredLEDs` - LED control

Building & Running

```
./compile_ws.sh
argos3 -c ./experiments/bugs.argos
```

Files to submit

1. you should submit your project in the same structure as `ex3_ws` (zipped).
2. Add a README file with your ID.
3. Your report PDF.
4. **Do not submit any executable files, or any files that can be regenerated**.

Grade

Your grade will be based on:

1. Algorithm correctness
2. Report quality
3. Success rate on different arenas
4. Code quality

Rules

1. Code must be tidy and well-commented
2. Include your **ID** at top of each file modified
3. All files must be typed (no handwriting)
4. Cite any source used in your report