Exercise 4

submission date - 27.1.26 (23:59)

Overview

This exercise is a competitive foraging competition.

Exercise Instructions

Implement a foraging controller that will represent you in the foraging competition.

Competition Structure

The competition will be made up of several **rounds**.

Rounds

1. In each round your controller will face another student's controller in competitive foraging.
2. You will each get a team of K robots controlled by your own implementation.
3. You will share the arena:
   Each team will have **its own base**, but food items **will be shared**.
   meaning, you need to compete over food.
4. Each round will last T time (a finite amount of time)
5. At the end of the round each team will be granted scores according to the number of food items collected.

Competition

1. Each student will face each other student for several rounds (with different random seeds).
2. The part of your grade that is related to success will be based on the **average number of food items you get per game**.

Goal

Get as much food items as you can (to increase your average score), and block the opposing team's attempts to gather food (to decrease their average score).

Presentation

Prepare a presentation that describes:

1. Your team's strategy.
2. The state machine you implemented.
3. What behavior is tied to each state?
4. How does your team react to robots from the same team? and to robots from the other team?
5. Are you satisfied with your team's success?
6. Anything else that might be interesting.
   The presentation should be no longer than 10 minutes!

Presenting

You will need to present your presentation to the class.
The presentations will take place in the last tutorial - `28.1.26`
Register for a presentation time slot & group in lamda

Workspace Structure

The `ex4_ws` Repository - https://github.com/OmriPer/ex4_ws
contains 3 directories:

- **`controllers/`** : Your controller code **(only modify this)**.
  You should implement your controller under `team1` .
  You should use `team2` as the opposite team to test your controller.
  You can implement it by either:
  1. implementing a basic foraging controller (as learned in class)
  2. using your own controller
  3. sharing your controllers and checking against each other.
     Both `Controller1` & `Controller2` inherit from `ForagingController`
- **`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_basePositions` - holds the base position array
- `m_teamColor` - the current controller's team color

Actuators:

- `m_pcWheels` - wheel control
- Do not use the LED actuators as they are used to signal the controllers' team.

Building & Running

```
./compile_ws.sh
argos3 -c ./experiments/round.argos %% with visualization %%
argos3 -z -c ./experiments/round.argos %% no visualization %%
```

Round results will be logged to `foraging_scores.csv` at the end of each round.

Submission

1. You should submit your project in the same structure as `ex4_ws` (zipped).
2. You should submit your presentation by `20.1.26`
3. **Work in groups of 2 only**, register your groups in lamda, only 1 partner should submit.
4. **Do not submit any executable files, or any files that can be regenerated**.

Grade

Your grade will be based on:

1. Your competition score
2. Presentation quality
3. 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 presentation \ code
5. Do not use the LED actuators!