Distributed Intelligent Systems
Lab 2 Tutorial

Chiara Ercolani

11.03.2021
Feedback forms

To help us improve, please fill them in. They are anonymous.
Why robotics simulation software?

- Hardware prototyping is time consuming and expensive
- Real commercial robots are expensive
- Ability to quickly change the experimental set-up
- Sometimes easier to measure physical quantities
- Sometimes faster than real-time
  - Numerical optimization methods (GA, PSO, etc.)
Robot prototyping and simulation software

Can model practically any type of robot:
- Wheeled, legged, flying, swimming, etc.

Programming interface to C, C++, Java, Matlab

Accelerated OpenGL graphics

Physics simulation with Open Dynamics Engine (ODE)
Physics-based simulation (ODE)

Mechanical systems that have:

- Rigid bodies (solid objects)
- Joints (like hinges)
- Contact and collisions
- Friction (keeps a tower of cards steady)
- Gadgets (like springs)
Many robot models: Khepera, E-Puck, Aibo, Pioneer 3DX, DARwIn-OP, etc.

- Sensors: distance sensors, light sensors, cameras, touch sensors, GPSs, force sensors
- Actuators: servo-motors, grippers, LEDs, connectors, etc.
- Emitters and receivers (multi-agent systems)
- And more ...
Using Webots @ Home

- Available for many OS (Linux Ubuntu 16.04, 18.04 and 20.04, Windows 10, MacOS)
- Download Webots installation package from
  - http://www.cyberbotics.com/
  - See installation instructions on Moodle
Reminder: Webots GUI

- Scene tree
- World view
- Editor
- Console
e-puck robot

- 8 proximity sensors
- 8 light sensors
- 1 color camera
- 3 microphones
- 1 speaker
- 3 axis accelerometers
- and more ...
Robot control – Rule based

Read front distance sensor

Is object too close?

Continue moving forward

No

Yes

Make a turn

Yes

No
Robot control – Braitenberg
Finite State Machine (FSM)

- **OBSTACLE AVOIDANCE**
  - Distance $< 5$
  - Distance $\geq 5$
  - Distance $> 15$
  - Distance $\leq 15$

- **LIGHT FOLLOWING**
  - Distance $> 15$
  - Distance $\geq 5$
And now: let's start!