Distributed Intelligent Systems
Lab 2 Tutorial

Chiara Ercolani

11.03.2021

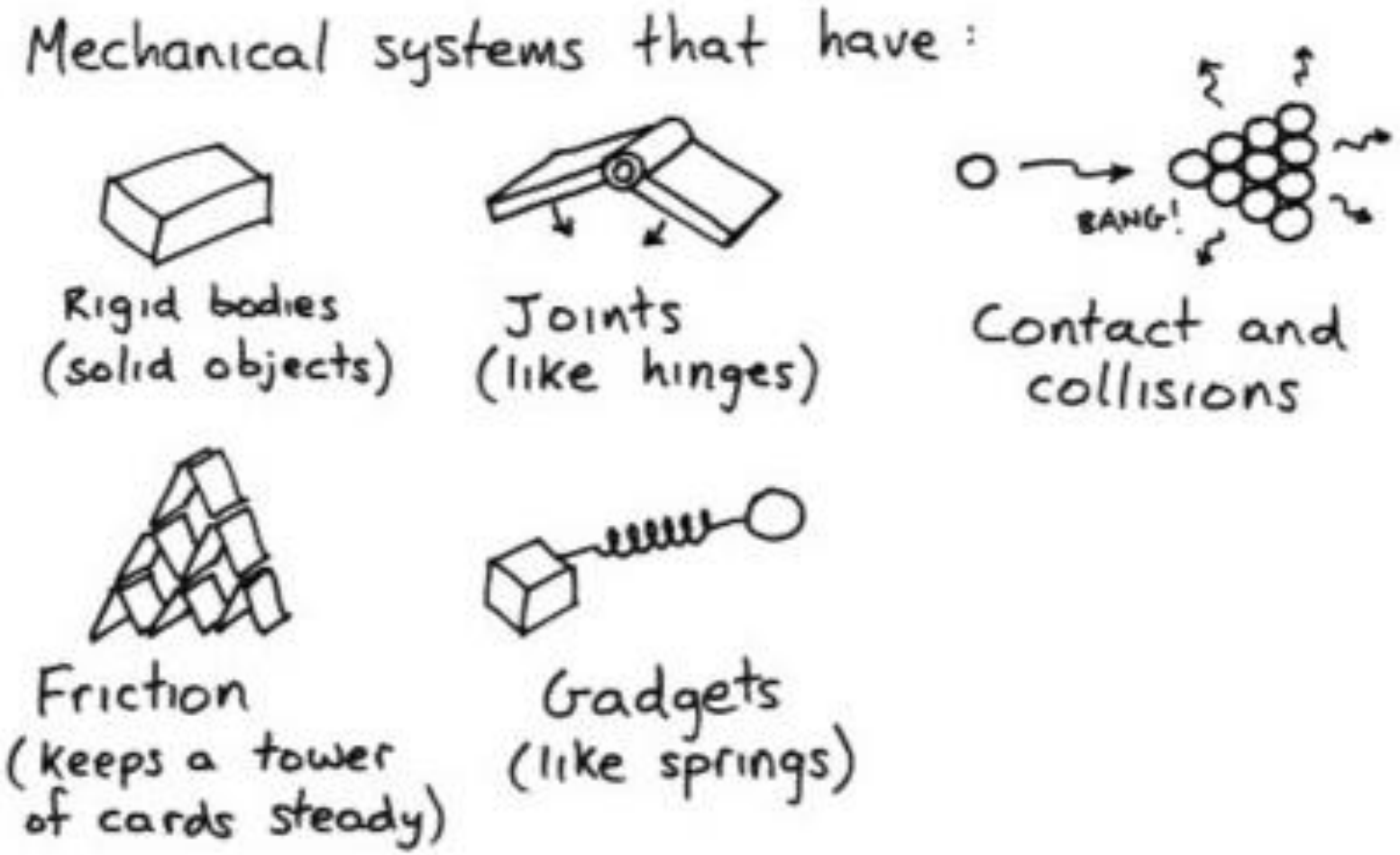
Feedback forms

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)



- 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

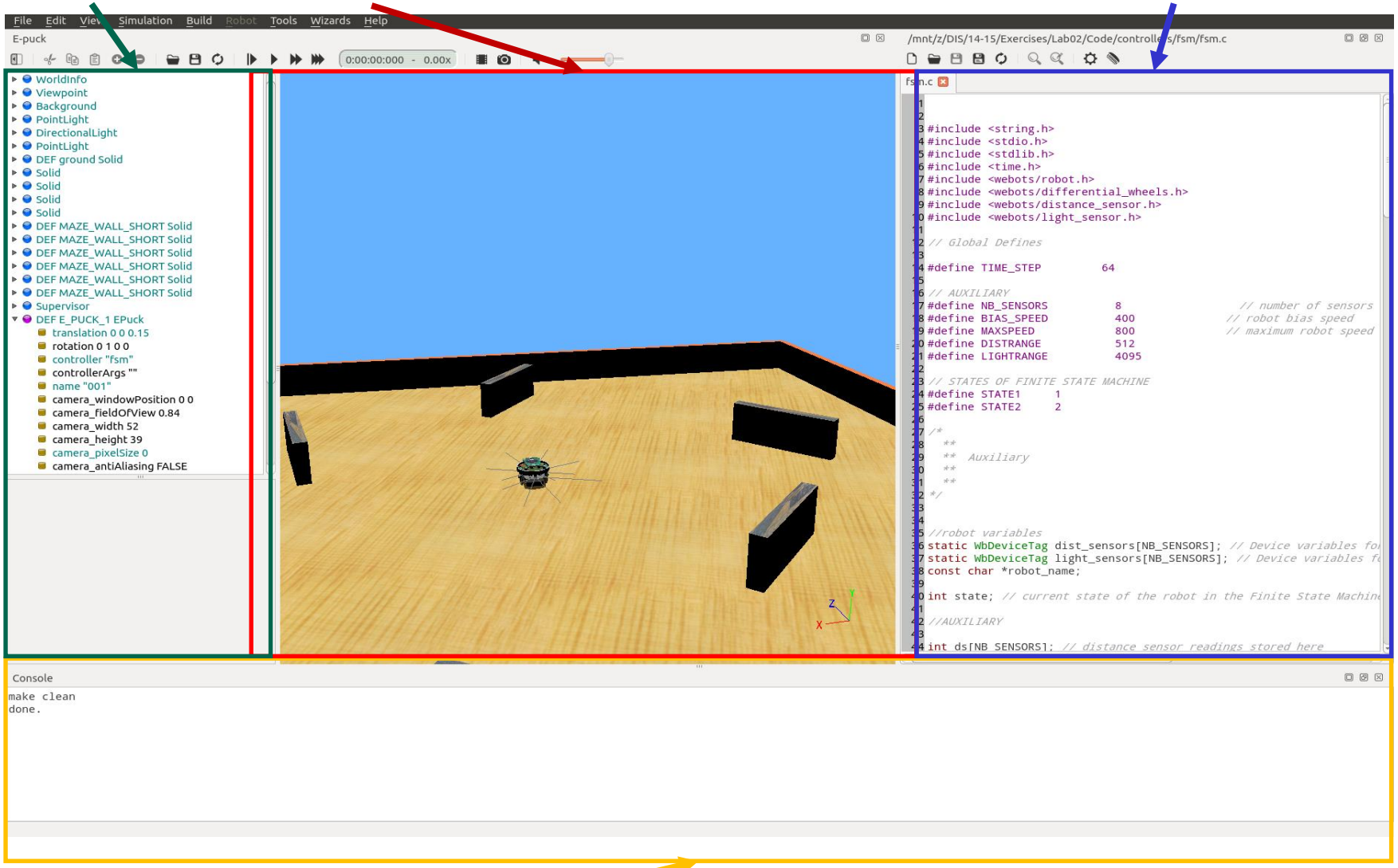
- Available for many OS (Linux Ubuntu 16.04,18.04 and 20.04, Windows 10, MacOS)
- Download Webots installation package from
  - http://www.cyberbotics.com/
  - See installation instructions on Moodle

Reminder: Webots GUI

scene tree

world view

editor

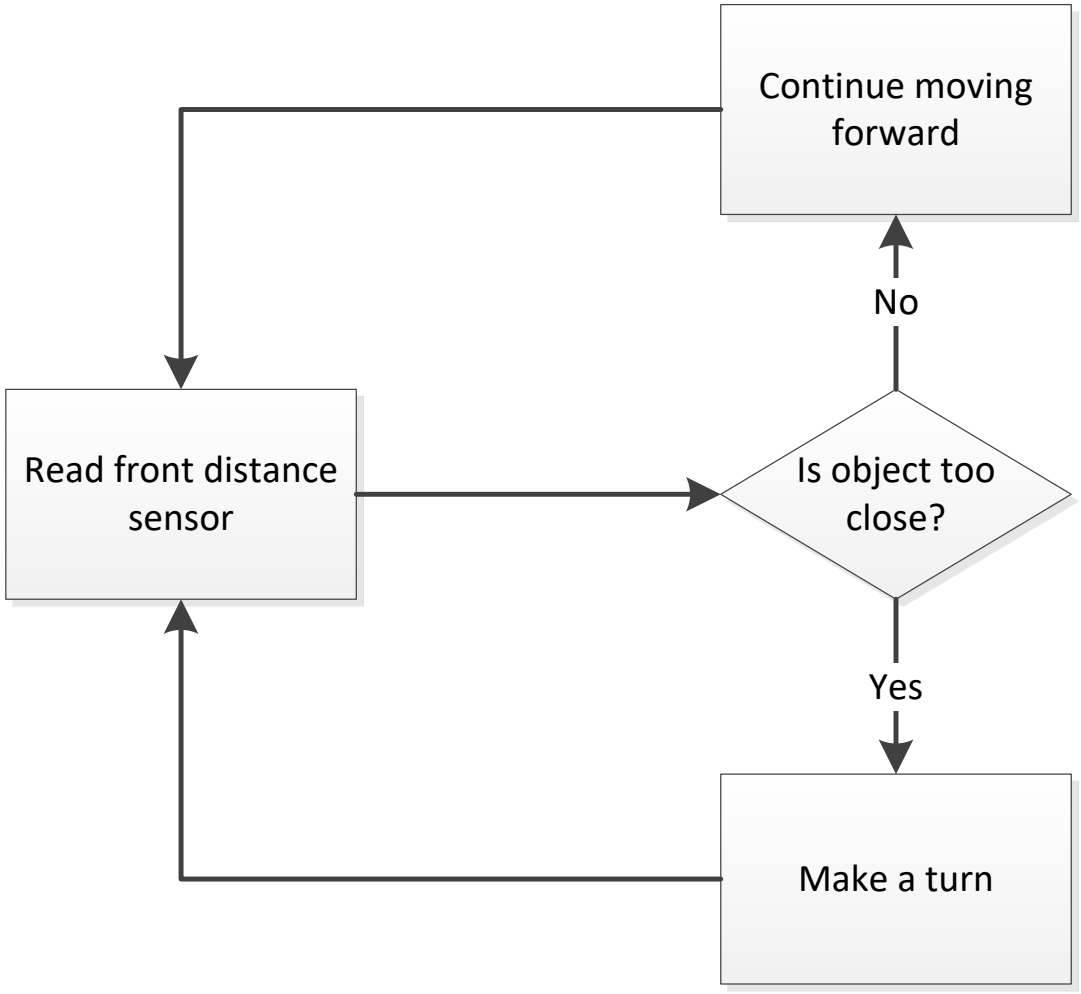


console

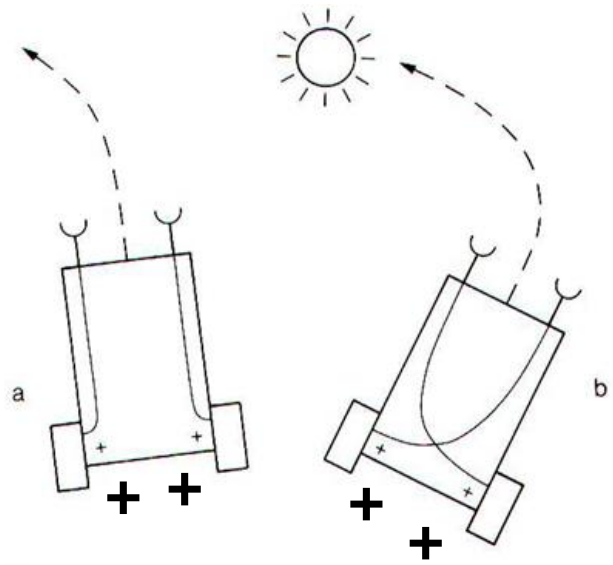
e-puck robot

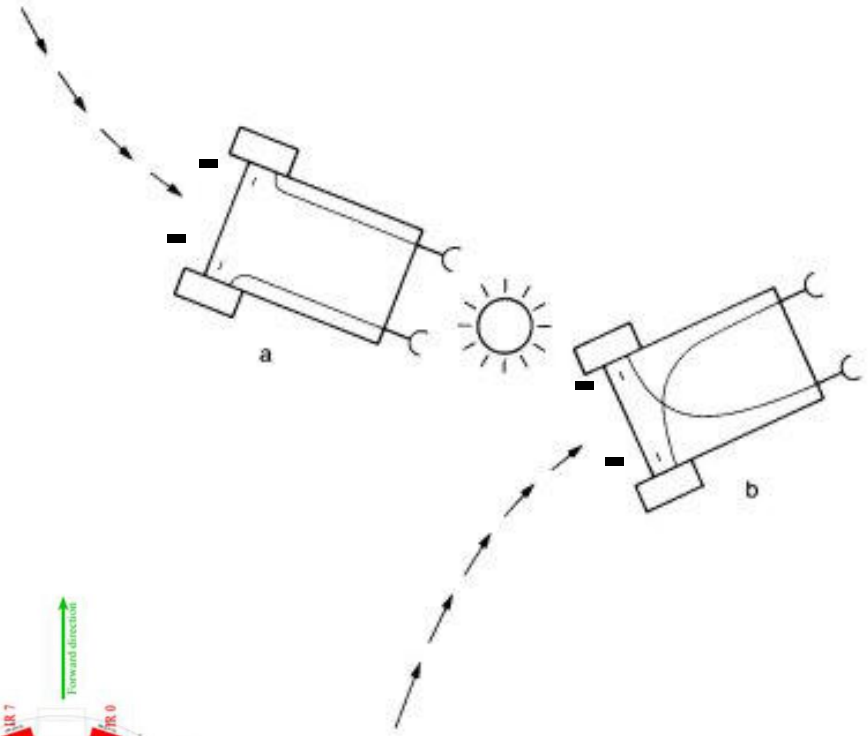
- 8 proximity sensors
- 8 light sensors
- 1 color camera
- 3 microphones
- 1 speaker
- 3 axis accelerometers
- and more ...

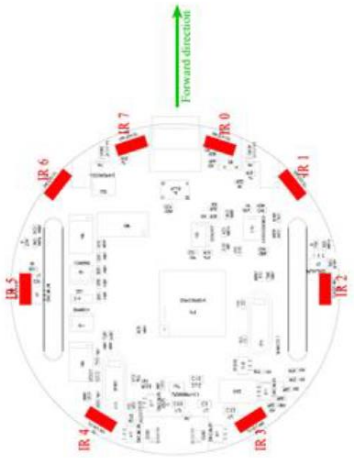
Robot control – Rule based



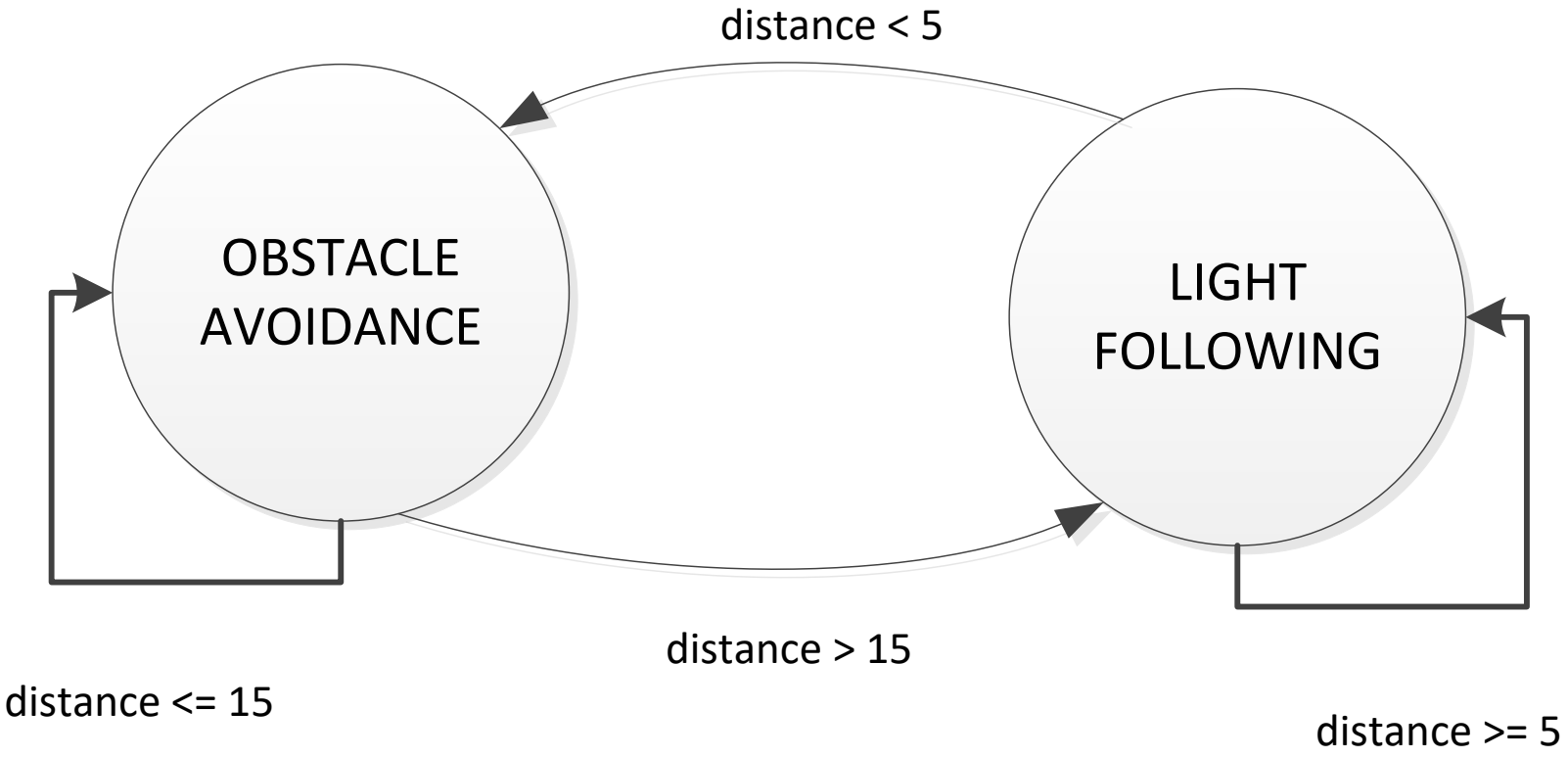
Robot control – Braitenberg







Finite State Machine (FSM)



And now: let's start!