



SAPIENZA
UNIVERSITÀ DI ROMA

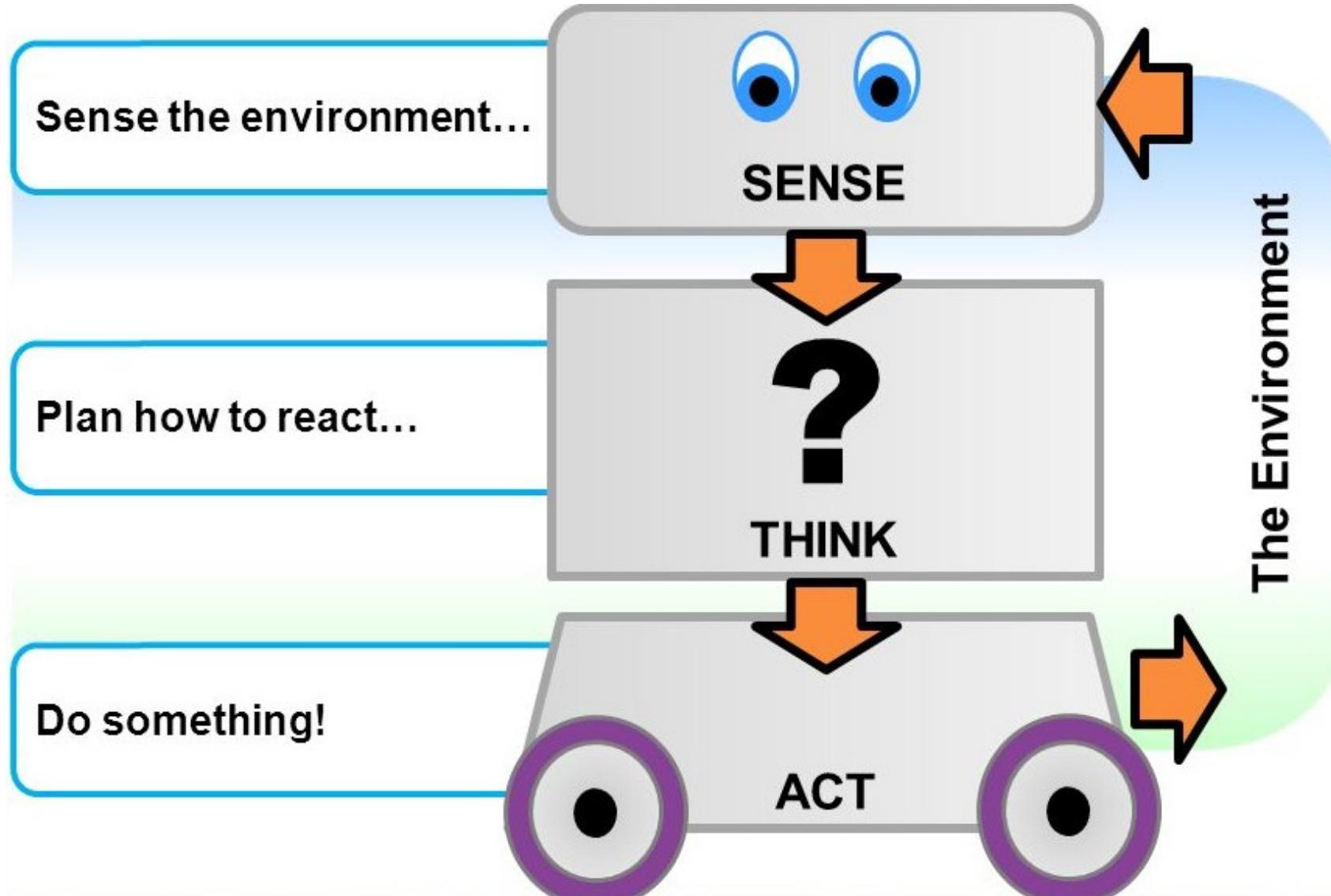
Robot Programming

Course Intro

Giorgio Grisetti
Emanuele Giacomini
Simone Ferrari

email: {grisetti, giacomin, ferrari}@diag.uniroma1.it

Sense-Plan-Act



Topics

Build Systems

- Make
- CMake

C++

- Basics
- Classes and OOP
- Meta Programming

ROS

- Concepts
- Robots and Sensors
- Robot Setup
- Navigation

C++

Compiled to machine code

Supports

- Imperative programming
- Object Oriented
- Generic Programming

A mixture of them

Used in doing the engines you
are used to work with

```
template <typename ContainerType>
int loadPoints(
    ContainerType_& dest,
    std::istream& is) {
    using VectorType =
        typename ContainerType_::value_type;
    int dim=VectorType::RowsAtCompileTime;
    while (is.good()) {
        VectorType v;
        for (int i=0; i<dim; ++i)
            is >> v(i);
        dest.push_back(v);
    }
    return dest.size();
}
```

Robots and Sensors

Sensors: devices that measure some environment quantity and report it in digital form to a program.



Actuators: devices that alter the environment by applying a perturbation to some quantity based on a program input.



ROS

Middleware for robotic/agent systems

Provides an infrastructure to design an application with multiple processing nodes

Provides basic utilities and core functionalities for designing robot systems

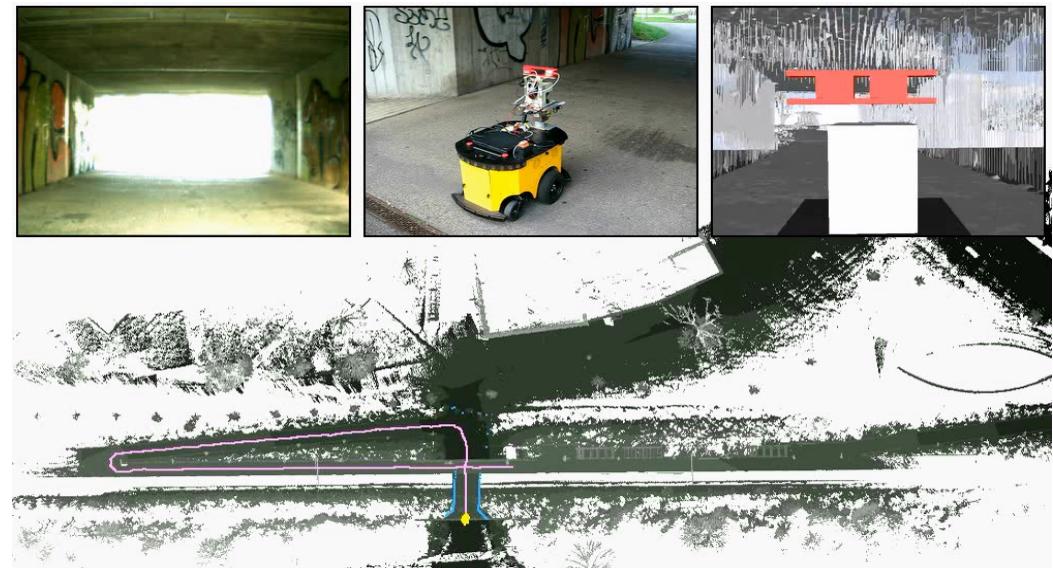


Navigation

Ability to move from A to B, in a MAP.

Requires

- Constructing a map
- Localizing in a map
- Planning a feasible path
- Executing the path



Where do I get the material? On the net

All info about the course is available at this webpage

Students **must** enroll in the form.

All material is available at this git repo

https://gitlab.com/grisetti/robotprogramming_2024_25

We encourage the students to install **natively** Ubuntu 24.04 and ROS2 Rolling.

Additional software will be made available on the repo. During the course

Contacts

Write an email to the teachers (**all of them**), with subject starting with [RP2024], so that we process your emails faster.

For clarifications we will reserve 1 h slot at the end of the lesson on Thursday (11:00-12:00) so that we can address your issues in presence (in the lesson room).

Should this not be sufficient you can reserve an appointment in presence or remotely with the teachers by email.

Evaluation

The evaluation is binary: pass/no_pass

It consists of

- A questionnaire to be done in class
- **Either** a final project **OR** 2 out of three homeworks

The project/homeworks will be developed on a **git repository**.

The evaluation is done by

- Checking the correctness of the result
- Inspecting the git history of the development
- Asking questions about specific parts of the developed code

Cases of plagiarism, copy and improper use of AI tools will be resolved by invalidating the project/homeworks and assigning a new one.

Structure of the Lessons

Besides the initial phase, we will proceed by alternating

- a “lecture” part on Tuesday, and at the end of a lesson we will propose a set of exercises.
- an “exercise” part on Thursday that assumes you did the exercises and presents the solution.

Exercises require a computer with Linux.

Doing the exercises allows for an easy development of the project, which might consist of the combination of some of them, plus some glue code.