



SAPIENZA  
UNIVERSITÀ DI ROMA

# Robot Programming

## Course Intro

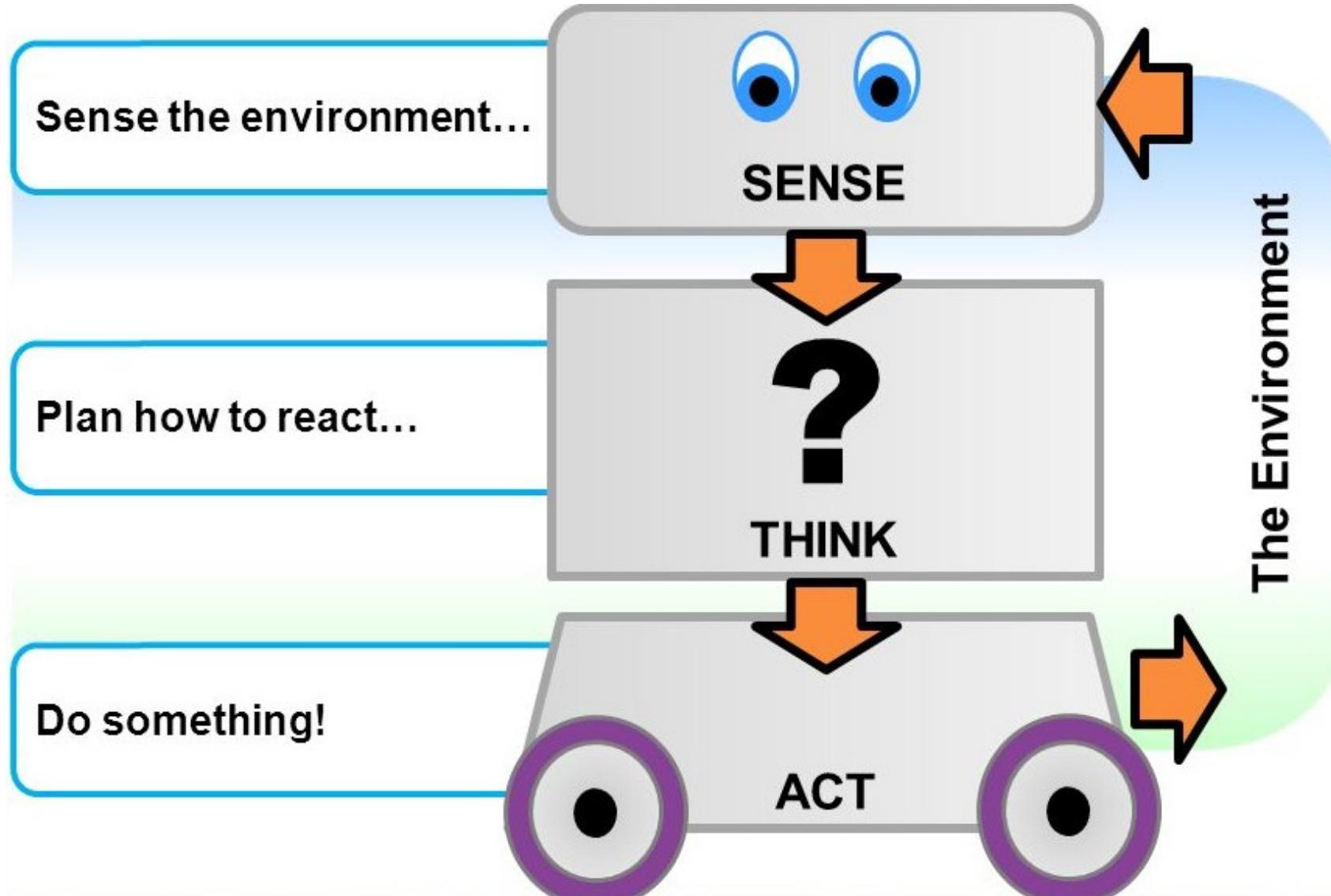
Giorgio Grisetti

Emanuele Giacomini

Simone Ferrari

email: {grisetti,giacomini,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

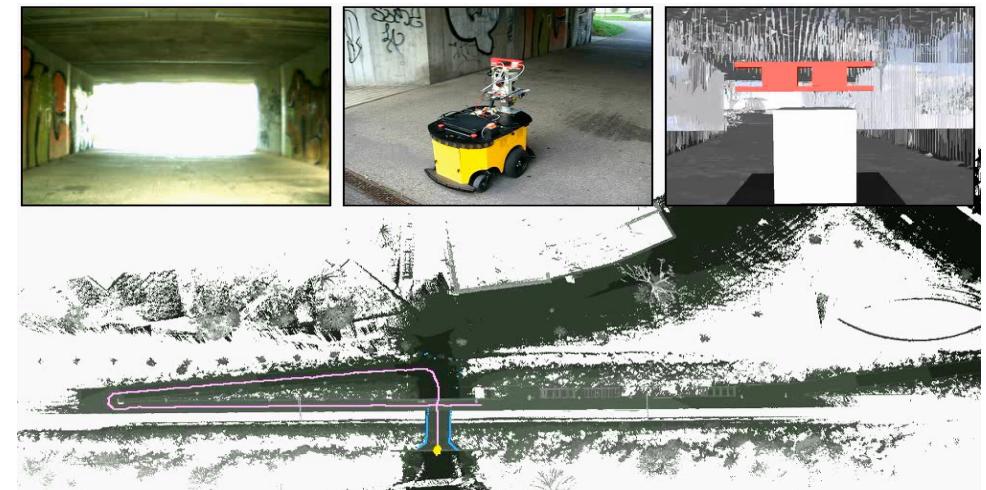


# Navigation

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\\_2022\\_23](https://gitlab.com/grisetti/robotprogramming_2022_23)

We encourage the students to install **natively** either

- Ubuntu 18.04, and ROS melodic
- Ubuntu 20.04 and ROS noetic

Alternatively you can download an ubutu ready to use VirtualBox image and install there the required software

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 [RP2022], so that we process your emails faster.

For clarifications we will reserve 1 h slot at the end of the lesson on Tuesday (18:00-19:00) 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 (binary) will be carried on based on a project assigned at the end of the course

The project will be developed on a public REPO and we will check the history of the commits to track the development

Projects might be carried on in (small) groups as long as the contribution of each group member is clearly assigned

Projects will cover multiple topics of the course

# 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.