

# Fundamentals of Artificial Intelligence

## NAO Planning Competition 2019



ALMA MATER STUDIORUM  
UNIVERSITÀ DI BOLOGNA

**Michela Milano**  
**Allegra De Filippo**

*Dipartimento di Informatica – Scienza e Ingegneria  
Università di Bologna*

[michela.milano@unibo.it](mailto:michela.milano@unibo.it)  
[allegra.defilippo@unibo.it](mailto:allegra.defilippo@unibo.it)

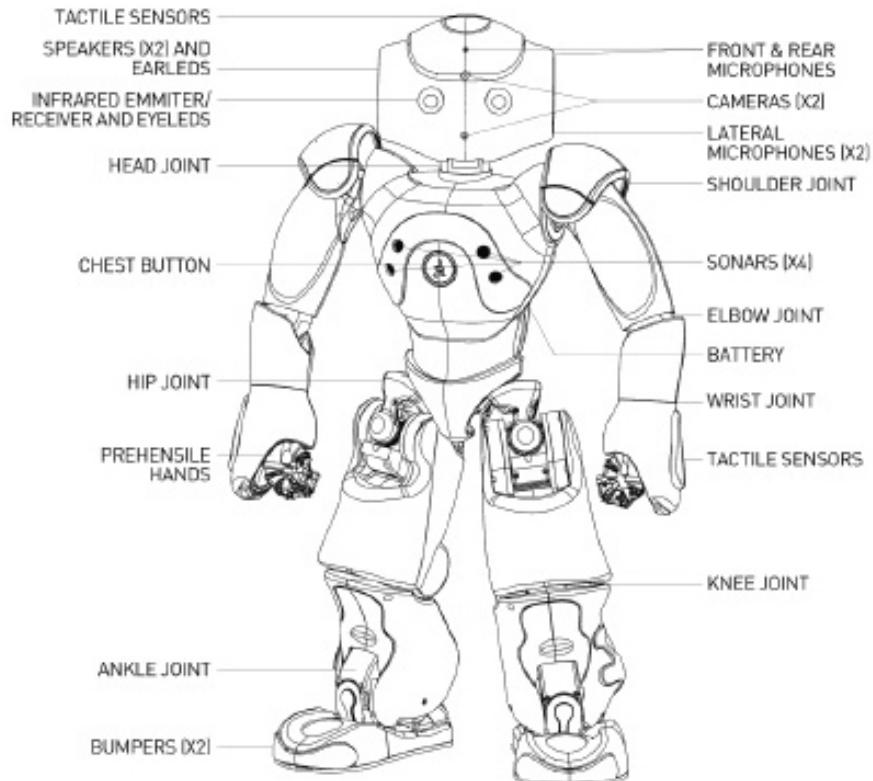
# NAO Planning 2019 – objective of the competition

---

- To stimulate the comprehension and the discussion regarding the basic algorithm for planning, in the context of AI discipline
- To test your skill on a fun and intuitive case study: the humanoid NAO robot
- To WIN the competition!



# NAO Robot – some info



# NAO Robot – some info

## MOVE

- 25 degrees of freedom
- Motors controlled by software
- Complex movement capabilities

## SENSE

- 2 HD camera
- 4 microphones
- 2 bumpers
- 2 sonars

## INTERACT

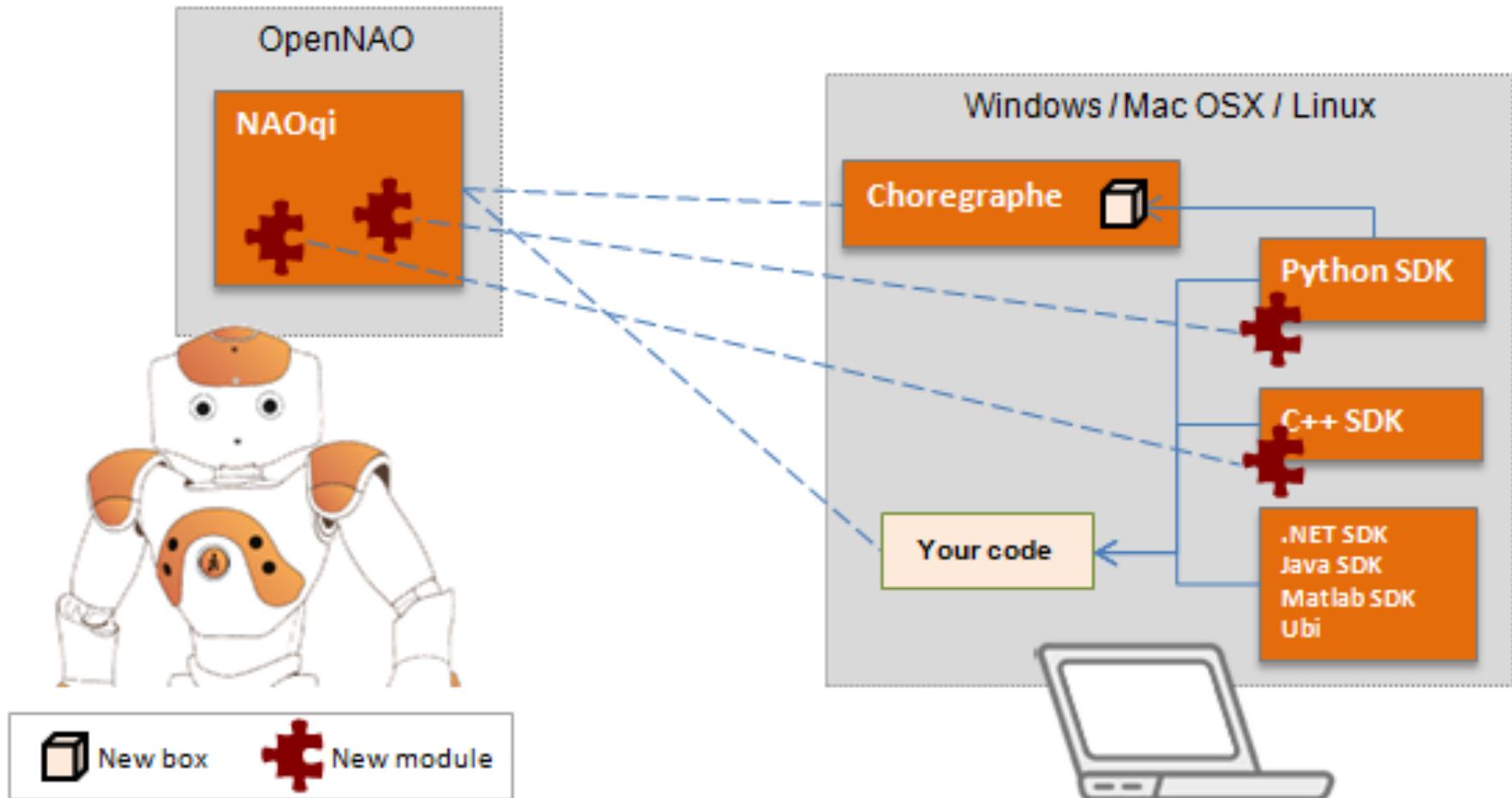
- 2 speakers
- multiple LEDs
- tactile sensors
- prensile hands
- infrared sensors
- WiFi connection

## THINK

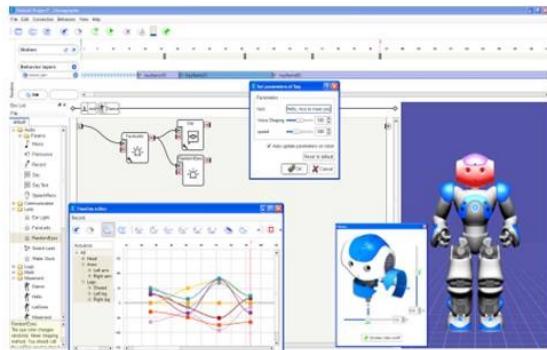
- Intel Atom 1,6 GHz CPU
- 1 Gb RAM
- 8 Gb Flash Memory
- Software suite



# NAO Robot – some info

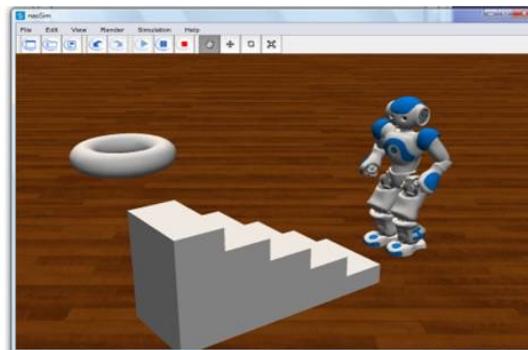


# NAO Robot – Software Suite



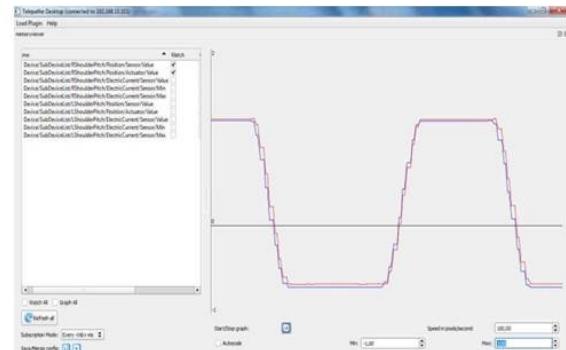
## C Choregraphe

- ✓ Graphical Development of Behaviors
- ✓ Ergonomic and user-friendly Interface



## S NAOsim

- ✓ Physical Simulation Engine
- ✓ Behaviors Simulation and validation



## M Monitor

- ✓ Ergonomic Interface to monitor actuators and sensors data



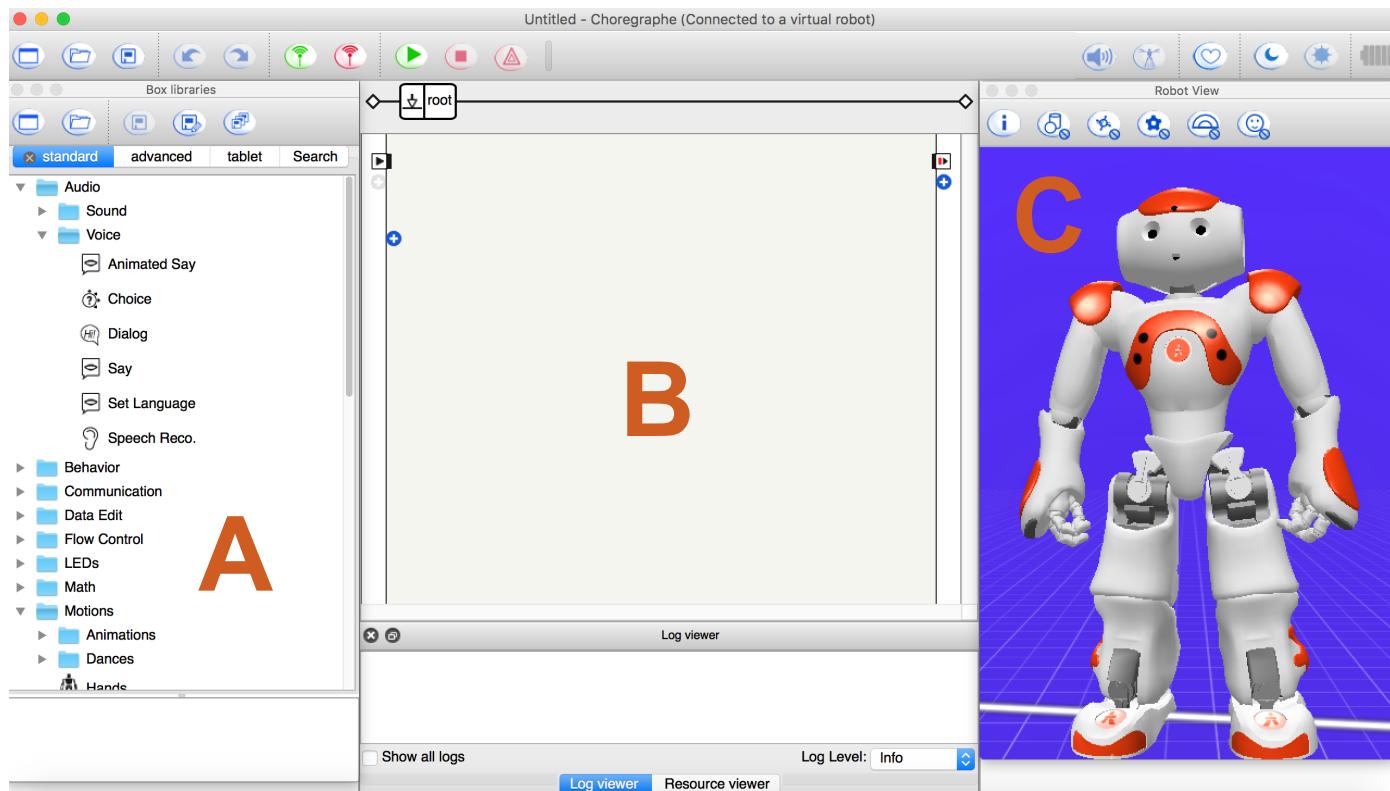
## SDK

- ✓ Compilation and debugging tools
- ✓ MatLab, Java, Python, C++, .NET, MS Robotics Studio

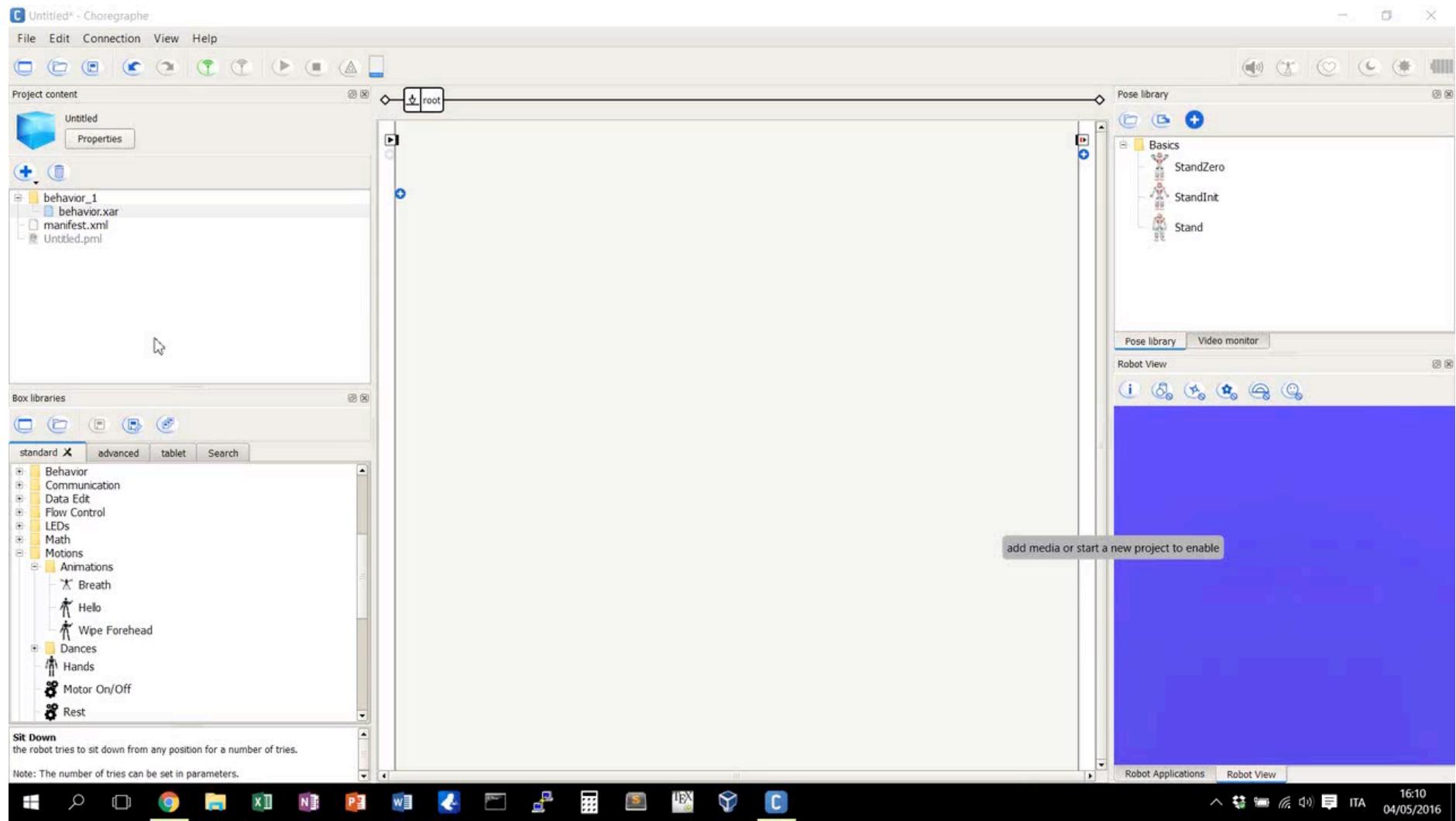
# NAO Robot – Choregraphe

- A Box libraries panel
- B Flow diagram panel
- C 3D Robot View for simulation

[here](#) to  
download  
Choregraphe



# NAO Robot – Choregraphe



# Performing NAO – AI and creativity

---

- Performing Robots: automatic generation of theatrical dance movements in robots

The objectives of this project:

- to devise techniques for **automatic and creative** generation of complex movements in robots, such as choreographies
- strong interdisciplinarity, involving robot learning, human-robot interaction, cognitive studies of movement, creativity



# NAO Planning 2019 – competition rules

---

- Teams divided into groups of **2 or 3 students**
- Each group must **plan a choreography** (sequence of positions) given a problem description
- Each group must **choose a music** suitable for the choreography (by respecting the total time limit of 3 minutes) and **test it on the virtual NAO** (using Choregraphe)
- A **day of voting** will then take place (during the last lectures) in which **the winning choreography will be decided**, considered the most satisfying from the **artistic point of view**

# NAO Planning 2019 – problem description

## Modeling the problem

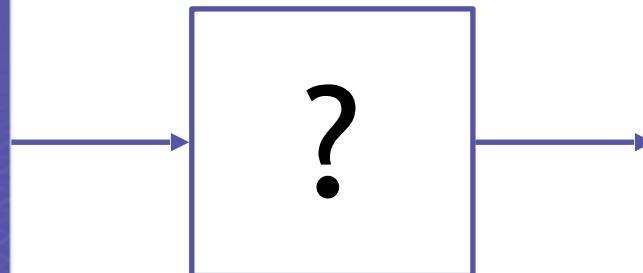
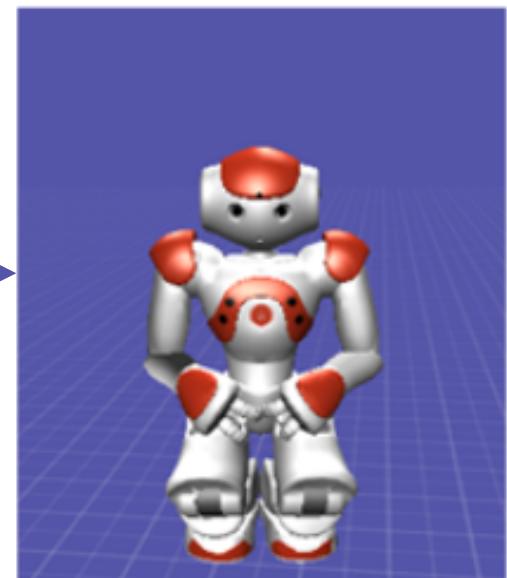
**Initial state**

**StandInit**

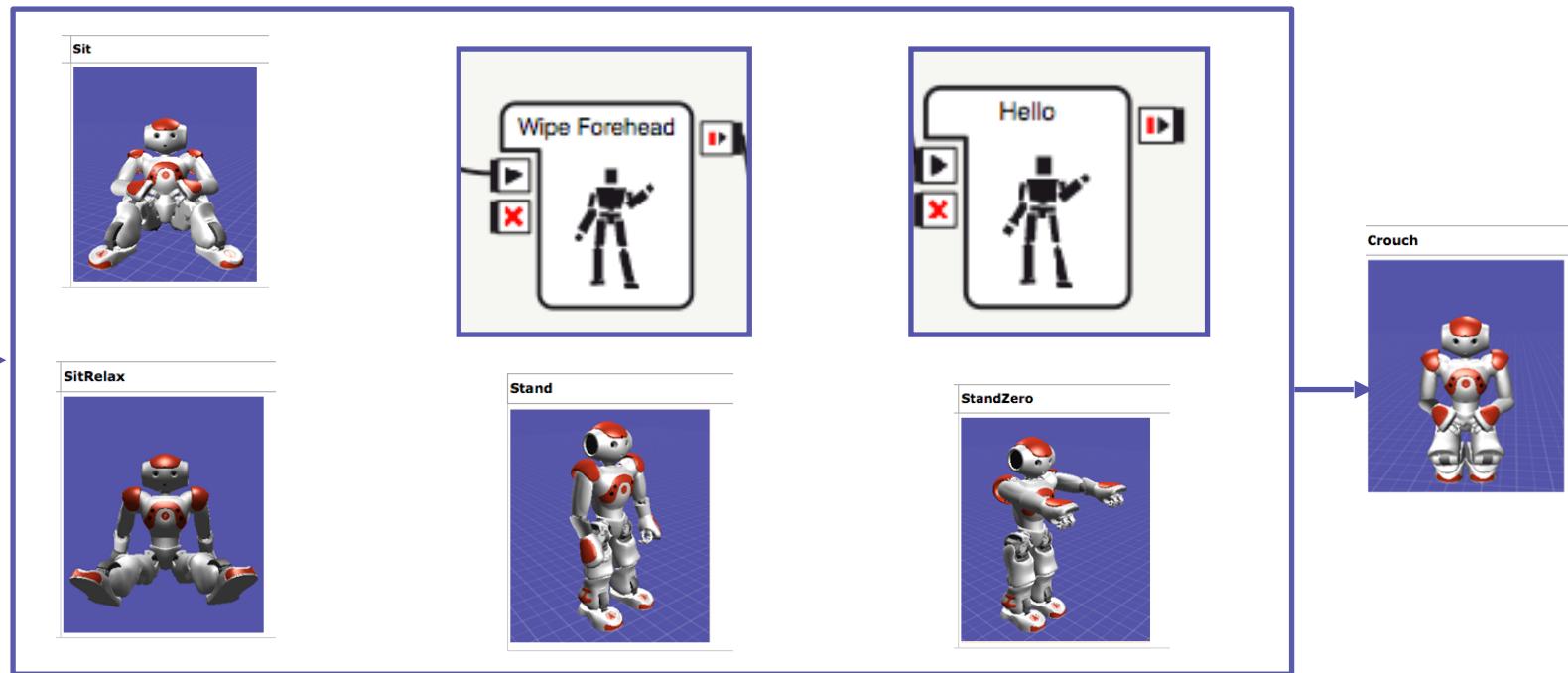


**Goal**

**Crouch**



# NAO Planning 2019 – problem description



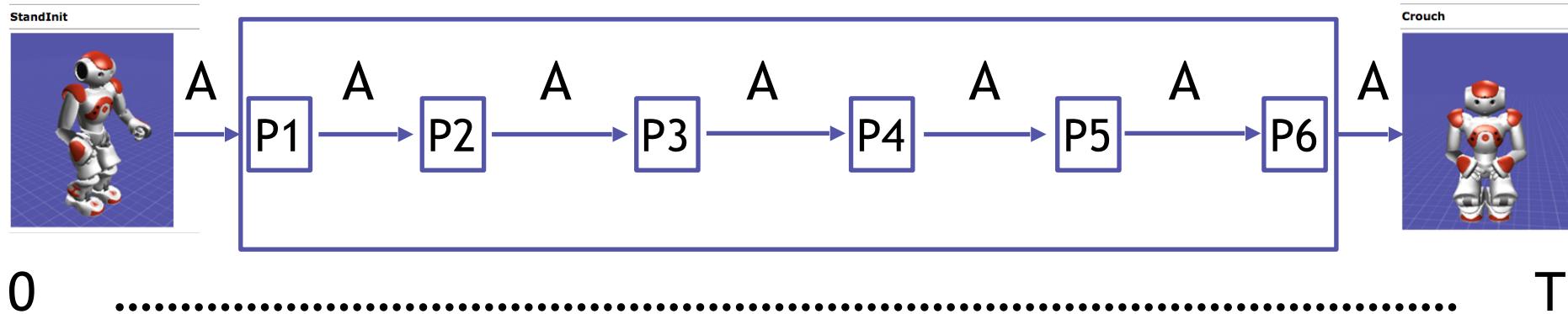
Mandatory positions

# NAO Planning 2019 – competition rules

---

- To move from a mandatory position to another, you can use positions from the available **set of intermediate positions** (see next slide)
- **Constraints** to be satisfied:
  - possible incompatibilities between two consecutive positions (use simulator in choreographe to understand if and what they are)
  - time constraints
  - constraints on the number of intermediate positions to be used in the whole choreography
- Generate an **algorithm A** able to plan the sequence of positions satisfying the given constraints (using a heuristic, or a planner for each sub-sequence of intermediate positions, ... we leave you free on the implementation choice)
- Hint: use Python

# NAO Planning 2019 – problem description



- **P1...P6** = mandatory positions
- **A** = algorithm to generate the transition between 2 mandatory positions by using the given pool of positions
- **T** = total time of choreography (3 minutes)
- **A** must use **at least 5** of the positions in the set and/or in the following .crg files

# NAO Planning 2019 – set of intermediate positions

---

- **rotation\_handgun\_object:** Nao makes a movement with the arm holding (possibly) an object
- **right\_arm:** right arm rotation
- **double\_movement:** rotation of both upper limbs
- **arms\_opening:** opening and rotation of both upper limbs
- **union\_arms:** movement of union of the arms
- **move\_forward/backward:** 3 steps forward/backward
- **diagonal\_left/diagonal\_right:** 1 left/right diagonal step
- **rotation\_foot\_Lleg/foot\_Rleg:** movement with one foot

# NAO Planning 2019 – crg files (for other intermediate positions)

---

.crg files to import directly on Choregraphe [1]:

- **sing\_with\_me**: NAO plays guitar
- **arm\_dance**: NAO dances by moving arms
- **birthday\_dance**: NAO dances birthday dance
- **sprinkler**: NAO dances sprinkler dance
- **workout**: NAO trains

[1] Copyright © 2014 University of Notre Dame (F.U.N. Lab)

# NAO Planning 2019 – position description

---

- All positions are coded in Python language and executable on a simulated robot (see [here](#) for further details of simulated NAO and [here](#) to download the robot positions)
- Choose a suitable music of 3 minutes of duration for your choreography.

**N.B.** Playing music cannot be tested on a simulated robot. Your algorithm A needs to solve the problem for simulated tests.

# NAO Planning 2019 – organization

---

- The **demonstration and voting day** will take place around the middle of December

## Important Dates:

- sending projects by **December 1** included (11.59 pm Italian time)
- presentation of projects, discussion, votes and winner (mandatory presentation of the whole team): during the last lessons of Module 1
- **Registration** by **October 28** included with an email to Allegra and Michela (in cc) specifying the name of your team and the email of all the members
- It will be possible to withdraw from the competition in any moment (by email!)

# NAO Planning 2019 – deliver

---

- **What:**
  - The folder containing the files of the entire project
  - A readme.txt file to specify
    - names and emails of the team participants
    - any necessary libraries that must be present
    - other useful information to test your project on simulated NAO
- **How:**
  - Mail containing links to repositories (Github, Dropbox, Drive etc.)
  - DO NOT ATTACH ZIP OR SIMILAR IN AN EMAIL

# NAO Planning 2019 – Demo

The image shows a dual-pane software interface for robotics development. On the left, a code editor displays a Python script named `chiamate.py`. The script performs several tasks, including printing piano keys, reading a file of moves, executing those moves via a robot's IP address, and handling command-line arguments for robot and port. On the right, the `Choregraphe` application is running, showing a 3D simulation of a NAO robot standing on a grid. The interface includes toolbars, a project browser, a pose library containing a `StandZero` pose, and a robot view window.

```
03_script planning + chiamate.py - C:\Users\michi\OneDrive\... - □ ×
File Edit Format Run Options Window Help
print "PIANO."
for line in piano:
    if line.startswith("****"):
        mosse.append(line[3:-1])
    print "\t"+line[3:-1]

#eseguo il piano
print "ESEGUEO messa in posizione iniziale"
#os.system("path_posizione_iniziale "+robotIP+str(port))
CREATE_NO_WINDOW = 0x08000000 #valore del flag creation
p = Popen(['python',path_posizione_iniziale,robotIP,str(port)], stdout = p.stdout.read())#per vedere l'output della chia
print output

for mossa in mosse :
    print "ESEGUEO "+mossa
    path_mossa = path_mosse+file_mosse.get(mossa)
    #print path_mossa
    #os.system(path_mossa)
    p = Popen(['python',path_mossa,robotIP,str(port)], stdout = p.stdout.read())#per vedere l'output della chia
    print output

if __name__ == "__main__":
    robotIP = "127.0.0.1" #"192.168.1.11"
    port = 52915 # Insert NAO port

    if len(sys.argv) <= 1:
        print "(robotIP default: 127.0.0.1)"
    elif len(sys.argv) <= 2:
        robotIP = sys.argv[1]
    else:
        port = int(sys.argv[2])
        robotIP = sys.argv[1]

    main(robotIP, port)

Ln: 82 Col: 16
```

# NAO Planning 2019 – final info...

---

- Some rules can be changed in every moment, with a notification on the course website (it is just the first challenge with NAO and planning).

## AI and creativity research:

**how can we train a neural network to try to recognize the artistic beauty of a choreography?**



- At the end of the course, all the choreographies and the related scores received during the day of presentations and voting will create a dataset for a neural network (which we will develop)

# NAO Planning 2019 – final info...

---

- Andrea and Federico will explain the rules of the other competition we propose: the Tablut challenge
- Participation is not mandatory
- You can participate in only one of the two competitions
- Bonus of 2 points on the final grade for those taking part in one of the two competitions

