Research on Nao robots.
Arina Yefimova
Instructor Lisa Miller(KCC)
Spring 2025

Let's start with some basic information about NAO:

Key Facts About NAO & Its Manufacturer:

- Original Developer: Aldebaran Robotics (a French company founded in 2005).
- Acquisition: In 2015, Aldebaran was acquired by SoftBank Robotics, which is headquartered in Japan.
- Current Manufacturer: SoftBank Robotics Europe (formerly Aldebaran).
- Official Website: SoftBank Robotics
- Primary Use Cases: Education, research, healthcare, human-robot interaction, and AI development.

Study NAO Robot View — Aldebaran 2.1.4.13 documentation
New NAO NAO AI Edition
Aldebaran - Nao6

Software downloads

Choregraphe download Aldebaran - NAO6 Software downloads
Make sure you have Microsoft visual C++ Download Visual C++ Redistributable Packages for Visual Studio 2013 from Official Microsoft Download Center

Python Python 2.7.0 Release | Python.org
Python GUI

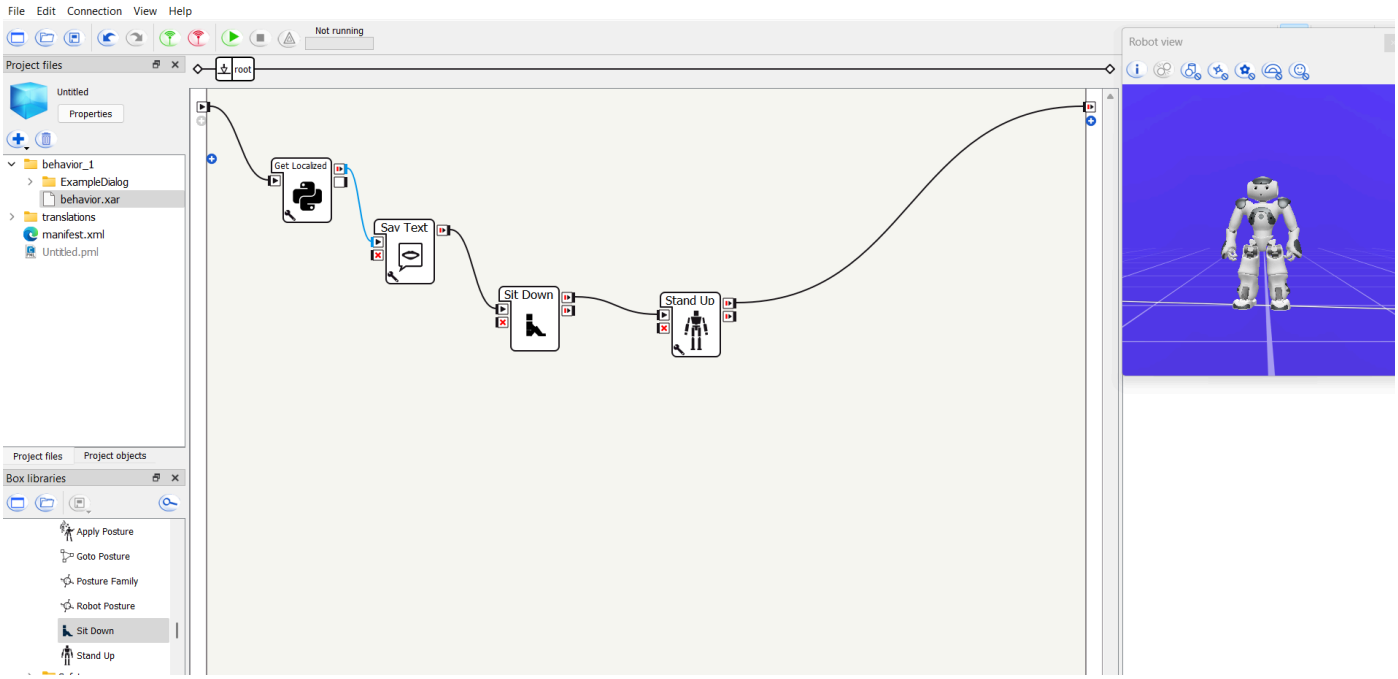
Study Choregraphe An Introduction To Robotics With Nao
Study Python Using the API - Making NAO speak — Aldebaran 2.8.7.4 documentation

Choregraphe:
Change the robot: edit -> preferences ->virtual robot

My research is focusing on Human-Robot Interaction (NAO responds to human speech or gestures).
Goal: Develop a system where NAO can interact with humans by recognizing speech commands and responding dynamically with voice and gestures.
Preliminary Prototype:
A system where NAO can listen to a human saying a command (e.g., 'Hello', 'Sit down', 'Stand up', 'Walk forward', 'Turn left', 'Turn right') and respond with appropriate speech and movement.

Progress on the project.

Jane helped me to connect Choregraphe software with Nao(connect your laptop to the same wifi as a robot) and I made NAO customize talk and stand up, sit down using sets of blocks in Choregraphe.



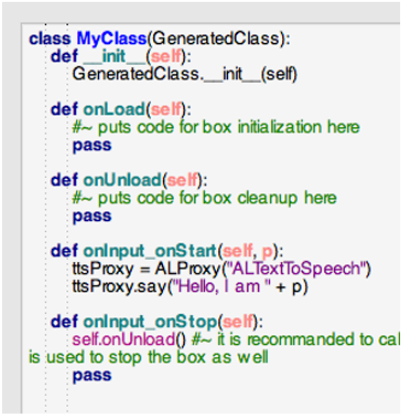
Coding in Python for NAO

Saying Hello

```
class MyClass(GeneratedClass):
    def __init__(self):
        GeneratedClass.__init__(self)

    def onLoad(self):
        #~ puts code for box initialization here
        pass

    def onUnload(self):
        #~ puts code for box cleanup here
        pass

    def onInput_onStart(self):
        ttsProxy = ALProxy("ALTextToSpeech")
        ttsProxy.say("Hello world!")

    def onInput_onStop(self):
        self.onUnload() #~ it is recommanded to call onUnload of this box in a onStop method, as the code
written in onUnload is used to stop the box as well
        pass
```

- The **ALProxy class** is part of the **NAOqi API**, which allows Python scripts to interact with different modules on the NAO robot.
- `"ALTextToSpeech"` is a built-in NAOqi service that converts text into speech.

For example, if the Speech Recognition box heard "nao", then the expression `"Hello, I am " + p` would evaluate to "Hello, I am nao".



```
class MyClass(GeneratedClass):
    def __init__(self):
        GeneratedClass.__init__(self)

    def onLoad(self):
        #~ puts code for box initialization here
        pass

    def onUnload(self):
        #~ puts code for box cleanup here
        pass

    def onInput_onStart(self, p):
        ttsProxy = ALProxy("ALTextToSpeech")
        ttsProxy.say("Hello, I am " + p)

    def onInput_onStop(self):
        self.onUnload() #~ it is recommanded to cal
is used to stop the box as well
        pass
```

```
def onInput_onStart(self, p):
    ttsProxy = ALProxy("ALTextToSpeech")

    if p == "nao":
        ttsProxy.say("Hello, I am a Nao humanoid robot.")
    elif p == "r 2 d 2":
        ttsProxy.say("Hello, I am R 2 D 2.")
    elif p == "c 3 p o":
        ttsProxy.say("Hello, I am C 3 P O, human cyborg relations.")
```

Conversation with Nao:

Person: Hello

Nao: Hello I am Nao. I am a humanoid robot designed to interact with humans. What is your name?

Person: p(name)

Nao: Hello + "p". How are you?

```
class MyClass(GeneratedClass):
    def __init__(self):
        GeneratedClass.__init__(self)

    def onLoad(self):
        #put initialization code here
        pass

    def onUnload(self):
        #put clean-up code here
        pass

    def onInput_onStart(self,p):
        ttsProxy = ALProxy("ALTextToSpeech")
        ttsProxy.say("Hello" + P + "How are you?")
        #self.onStopped() #activate the output of the box
        pass

    def onInput_onStop(self):
        self.onUnload() #it is recommended to reuse the clean-up
as the box is stopped
        self.onStopped() #activate the output of the box
```

Person: I am good/I am sad

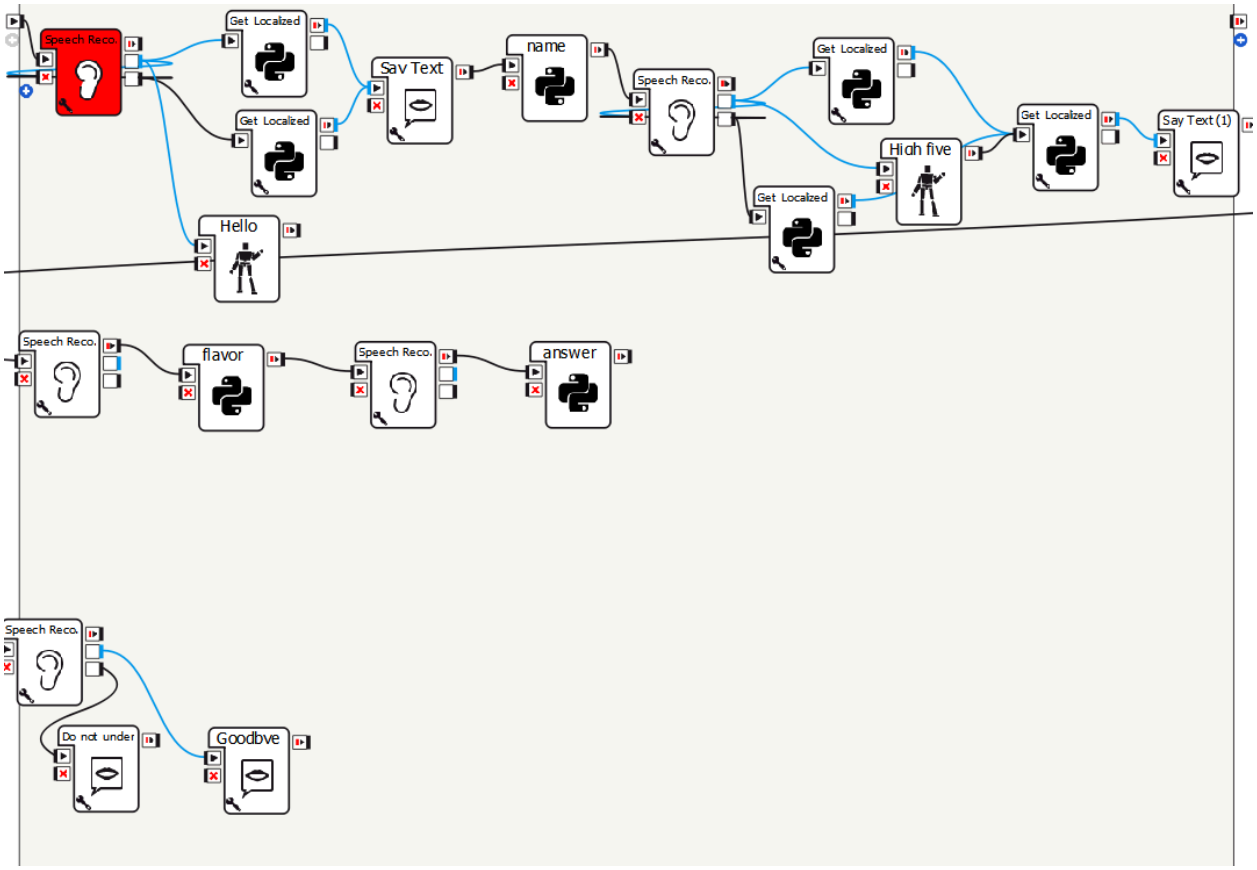
Nao: That's good to hear(gives high five)/ I hope your day gets better. I have a question for you. Do you like ice cream or cake?

Person: Ice cream/ cake

Nao: Do you like chocolate or fruit ice cream/cake?

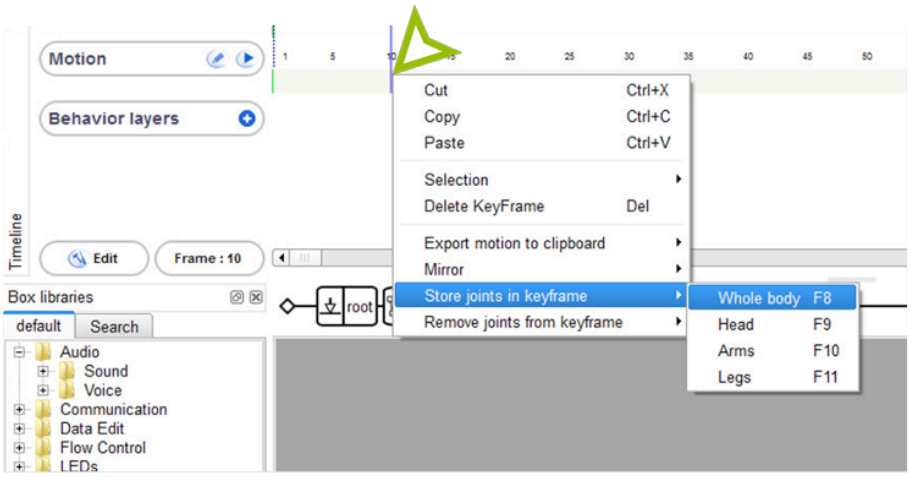
Person: Chocolate/fruit

Nao: I like chocolate/fruit cake/ice cream too, especially with whipped cream.



Plan in choregraphe but need to test it on a real robot.

Also set up the movement for high five when he replies to how are you and make him raise a hand (will be doing it when I connect to real robot using keyframe function in choregraphe).



Then add commands to ask the robot to sit down or walk around.