



LUND  
UNIVERSITY

# Project NAO

By Fredrik Folkesson & Johan Nyholm



# Plans and Goals

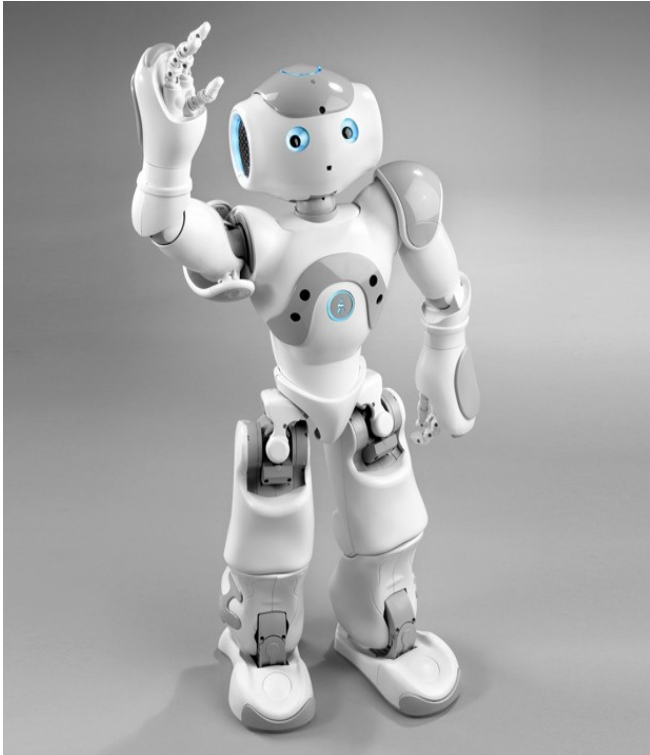
---

- **Goal:** To let the NAO robot walk in a corridor and greet familiar faces.
- **Additional:** Learn how to use the NAO robot python API, its restrictions and potential for serving as a platform for implementing algorithms.



# Project Initialization

---



- Coreographer, Python and C++.
- Read up on the NAO Python API.
- Develop out first NAO Python scripts.
  - Say "Hello"
  - Let the robot walk forward



# The NAO Python API

---

- Connect to robot using IP and port.
- Create proxies for handling different NAO modules such as: movement, posture, speech, camera etc.
  - `motionProxy.moveToward(X,Y,Theta)`
- Subscribe to sensor data which later can be polled.
- Listen to pre defined events.



# Face Detection

---

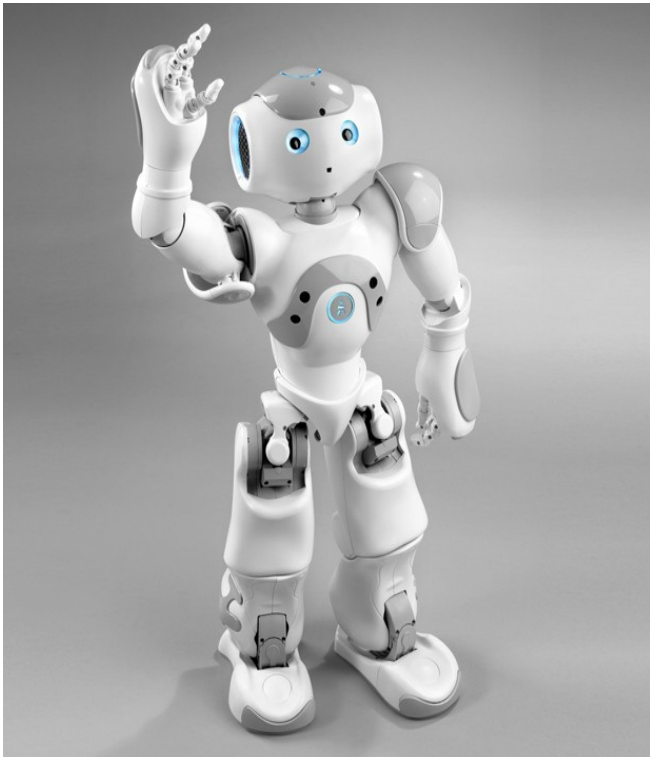


- Started researching how to do face detection and recognition.
- NAO already had finished face detection and recognition modules, callable through the Python API.



# Corridor Walking

---



- Problem: NAO can't walk "straight".
- Let NAO find and follow a wall using sensors.
- Decided to use the NAO robots sonar to keep a constant distance to the wall.



# Difficulties with NAO Sonar

---



- Used NAO Sonar to measure distances to a wall.
- Tried to use sonar readings to calculate how to navigate.
- Sonars on the left and right side often gave inconsistent and erroneous measurements.
- Difficulties with Sonar API, and inconsistency in documentation.
- Stereo vision?



# Implemented navigation tools

---



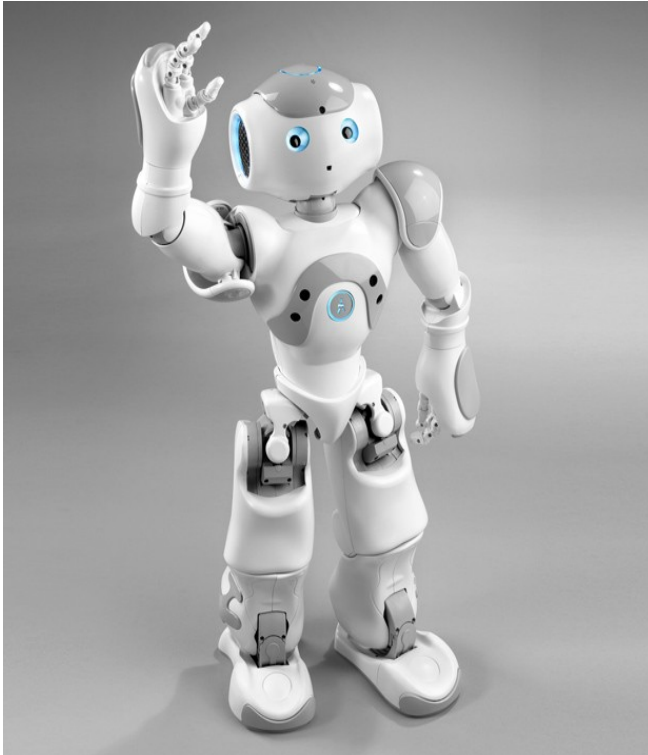
- Created Python modules in order to control the robot.
- Navigation, posture.
- Built program to control the robot with keyboard.
- Tested the use of speech recognition control. (forward vs backward?)





# NAO Image processing

---

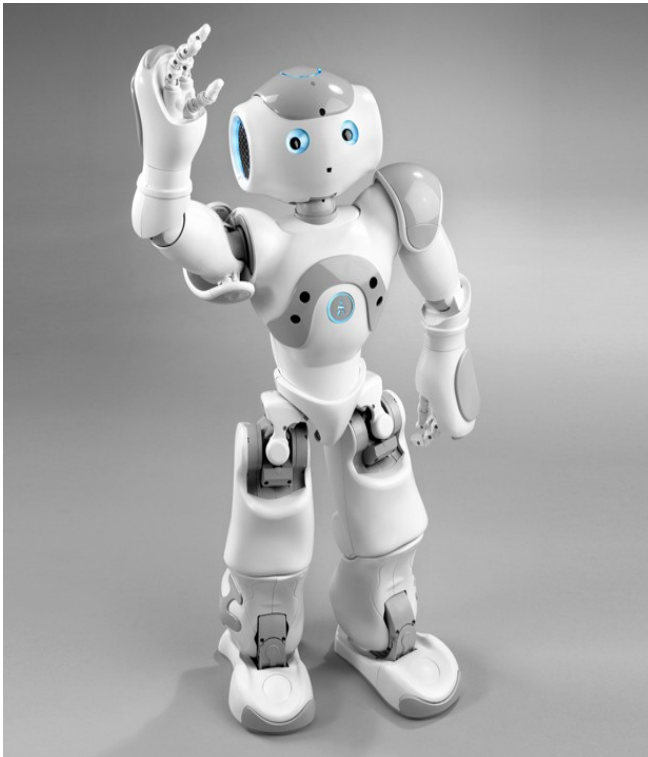


- Follow a line marked on the floor instead of a wall.
- Use NAO Camera in order to take pictures.
- Found instructions and code for NAO line detection from Aldebaran.
- Tried to make rough implementation work.



# NAO Line Follower

---



- Code and data flow analysis.
- Image processing understanding.
- Lighting issues, glossy floor created reflections.
- Tweaked constants in algorithm.
- SOLUTION 1: Change environment.
- SOLUTION 2: Better algorithm? (Hough transform)



# NAO Line Follower Continuation

---



- Extended the program, making NAO try to find a new line when finished following the last line.



# NAO Project Conclusion

---

- Face recognition implemented.
- Greeting implemented.
- Rough smart corridor walking implemented.
- Integration not done due to lack of time.





**LUND**  
**UNIVERSITY**