# Project Plan

How to train your Nao

Version 1.0

*TSBB11 HT2015*

# Project identity

| Name | Responsibility | Phone | Email |
| --- | --- | --- | --- |
| Madeleine Stein | Project leader (PL) | 0709383490 | madst314@student.liu.se |
| Susanna Gladh | Quality assurance (QA) | 0735030125 | susgl621@student.liu.se |
| Anton Ågren | Testing (TES) | 0768026950 | antag110@student.liu.se |
| Fredrik Kvillborn | Design (DES) | 0703050656 | frekv134@student.liu.se |
| Elin Andersson | Scrum master (SM) | 0736466026 | elian253@student.liu.se |
| Richard Bondemark | Documentation (DOC) | 0762464454 | ricbo818@student.liu.se |
| Fredrik Löfgren | Nao Expert (NE) | 0703571658 | frelo223@student.liu.se |

**Customer:** Michael Felsberg, LiU (ISY), michael.felsberg@liu.se
**Project supervisor:** Fahad Khan, LiU (ISY), fahad.khan@liu.se
**Examiner:** Michael Felsberg, LiU (ISY), michael.felsberg@liu.se

# Contents

# 1 Introduction

This is a document describing the way of work for the project group during the project *How to Train Your Nao* in the course TSBB11 *Images and Graphics, Project Course CDIO* at Linköping University during HT2015. At the end of the document an initial draft of a system overview is given.

## 1.1 Background

We live in a world where technology is constantly and rapidly evolving. Robots are becoming an increasing part of our everyday life. Humanoid robots are a hot subject, since they can provide assistance in many areas, as well as companionship.

## 1.2 Involved parties

The customer is Michael Felsberg at the department of Electrical Engineering (ISY) at Linköping University. The supervisor of the project is Fahad Khan at the department of Electrical Engineering (ISY) at Linköping University.

The end user will be the staff and students at ISY who continues to develop the platform and use it for demonstrations at different events, but also the onlookers at these events since they might interact with the robot.

## 1.3 Goal

The goal of this project is to implement computer vision algorithms on the Nao platform. The algorithms will be one part of the camera related artificial intelligence developed for the platform. This will in turn contribute to developing the platform for ISY so that they can perform interesting and inspiring demos and will get an initial system to continue building on. The result might also be used by the LiU robocup team, who uses the robot in different competitions around the world.

# 2 Organization

Each member of the project group will get a role, which is presented below.

| **Name** | **Responsibility** |
|---|---|
| Madeleine Stein | Project leader (PL), responsible for organizing the group and keeping in contact with supervisor and customer. |
| Susanna Gladh | Quality assurance (QA), responsible for assuring the product is good enough to satisfy all the requirements. |
| Anton Ågren | Testing (TES), responsible of making sure the product is properly tested and that the tests are adequate. |
| Fredrik Kvillborn | Design (DES), responsible for making sure the design of the product is coherent during development as well as for the final product. |
| Elin Andersson | Scrum master (SM), responsible for planning the sprints and keeping the team on the right track. |
| Richard Bondemark | Documentation (DOC), responsible for handling all documents and to make sure all the necessary documents are produced. |
| Fredrik Löfgren | Nao Expert (NE), responsible for sharing knowledge and insight about Nao platform. |

# 3 Development

The model used in this project will be a version of Scrum. The project group will use 14 day sprints, with sprint planning in the beginning and a review in the end of each sprint. A digital scrum board will be used in Trello but due to the lack of burndown functionality in Trello, this will be kept in a separate spreadsheet.

The project group will use Gitlab for revision control and Google Drive as the primary text editor for documents written in this project.

# 4 Resources

The project group will have access to two Nao robots provided by ISY. Further, the project group will have access to, and are allowed to use and build upon, all present implemented functionalities of the Nao robot. All online open source functionality packages will also be free to use during the project. Finally, the project group is supplied with a tutor to consult if needed.

# 5 System overview

This section will give an overview of the system, including hardware and software components, usability and installation, external dependencies, compatibility and limitations.

## 5.1 Rough description of the system

The Nao robot will collect images and/or video using its two cameras. This data will be used either on its on-board computer or on an external computer to compute a decision. The robot will then act accordingly.

## 5.2 Components

The core component is the Nao robot. The robot can both collect images, detect sensory input, move around in the environment as well as run the software on its on-board computer. An external computer will be used for extra computational power and will communicate data and results with the robot over WiFi. See more specifications of hardware, software and dependencies in Appendix A.

The project group will connect to the robot by its proxies and do all demanding image processing offline on an ordinary computer (either a stationary, or preferable a laptop to be able to do demos outside of the lab).

## 5.3 Limitations

This section will describe the hardware and software limitations of the final product.

### 5.3.1 Hardware

There will be some practical hardware limitations. That is, among other things, the lack of computing power on board the robot as well as the limitation of the WiFi communicating data and decisions between the external computer and the robot.

**Gyro:** The Nao V4 are missing the Z-axis in the gyroscope and can therefore not detect its turn angle using dead-reckoning. However it has a 3-axis accelerometer.

**Bandwidth:** Video streaming is possible in full resolution (960p) in 20-25 frames per second (fps) when connected to the robot with a cable. When using wireless connection the expected frame rate is only between 1-5 fps in full resolution. When several robots are connected to the same network the frequency will drop even more. Some video preprocessing can however be done on the robot before sending the data to an offline computer. It is also possible to reduce the resolution to 640x480 pixels, thereby allowing a higher frame rate.

**Processor:** The available CPU on the robot is just a single core, so not much performance can be gained by parallelizing computations on it. Since most of the calculations are intended to be done on an external computer this should not be a major concern.

**WiFi:** The router must allow 2.4GHz clients, since that is the only bandwidth supported by the robot.

For more information and links to the manufacturer's home page see Appendix A.

### 5.3.2 Software

The Nao computer lacks vector processing assembly instructions SIMD, used by OpenCV to do parallel calculations. The consequence is slower image processing then techniques using parallel calculations.

The code written for RoboCup Standard Platform League (SPL) was intended to run on the Nao robot itself, and is therefore adapted to its limited processing power. In this project more advanced algorithms can be implemented since the processing will be done offline.

## 5.4 Compatibility

The project should run on a Linux Ubuntu 14.04 computer, connected to a local network. The robot should be connected wirelessly to the same network.

# 6 System overview

From the main loop the robot should be able to enter into four modes. Figure 1-4 shows the flowchart for each mode. An arrow pointing into no box indicates that the program goes back into the main loop.

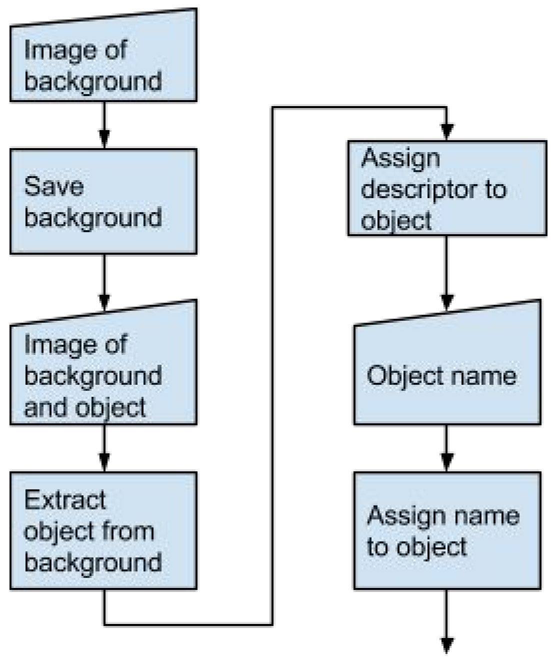


***Figure 1:*** *Flowchart over implementation of learning a new object. The Nao will get an image of the background. An object is then placed in the foreground and extracted from the background. A descriptor and object name is then assigned to the object.*

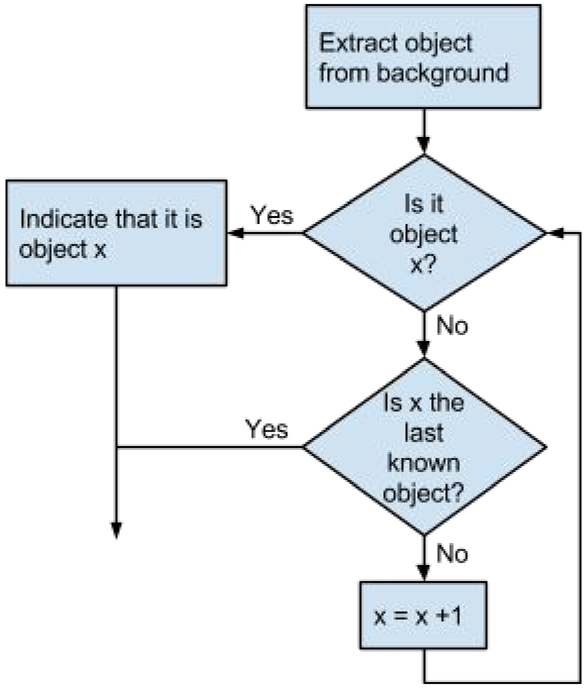


***Figure 2:*** *Flowchart over implementation of identifying which object is being shown to the Nao robot. The object is extracted from the image background. The robot then compare the extracted object to all objects the robot knows.*

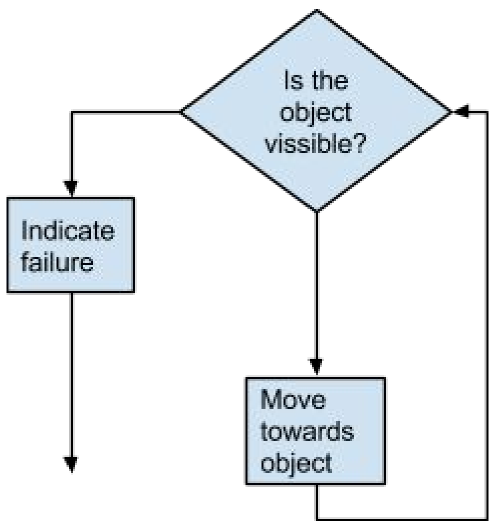


***Figure 3:*** *Flowchart over the implementation of the follow object routine. The Nao robot is given a known object to search for. Each image is then scanned for the object. If the object is visible the Nao robot should move closer to the robot. If the object is not visible the Nao robot should indicate that the search failed.*

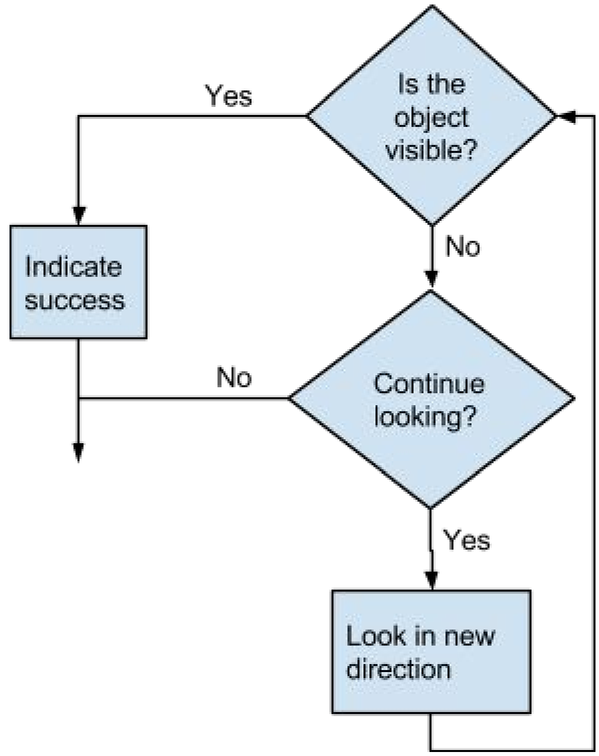


***Figure 4:*** *Flowchart over the implementation of the "hide and seek"-routine, where the robot is searching the environment for a specific object. The Nao robot is given a known object to search for. Each image is then scanned for the object. If the object is visible the Nao robot should indicate that the object was found successfully. If the object is not visible the robot should check if he should continue looking at a new angle or stop the search. The search is stopped when the robot has rotated 360 degrees without finding the object.*

# 7 Testing, applications and scenarios

Testing and applications are two important aspects of this project. The purpose of testing is mainly to be able to evaluate and see improvements in algorithms during the development, where the applications will be important for future development and identify the potential of the project and usefulness.

The testing of the object association algorithm will be made both through a demonstration as well as a quantitative evaluation.

## 7.1 Quantitative evaluation

The quantitative evaluation will be performed on the object association algorithm to measure its robustness in a statistical way. The evaluation itself will be a cross-evaluation, where one set of data will act as the training data while the rest will be used for evaluation. All of the data will once be used as training data and an average success rate will then be calculated.

## 7.2 Common scenarios and backgrounds

While testing and demonstrating the system there will be a range of scenarios with different difficultness. In the requirement specification document, three types of background scenarios are mentioned. An example of a *new environment* is a more or less clean room with minimal amount of disturbing objects. In the *new environment* light can change between the learning process and the evaluation or even that the learning takes place in one room and the evaluation in another. Examples of *advanced environment* is in open and crowded areas where there can be a lot of motion or change in light during the evaluation. The *advanced environment* can for humans be seen as everyday scenarios where nothing may be known about the nature of the environment.

The project group has access to a NIST arena and a small indoor soccer field, which will be used to create the different environments for training, and for demonstrations.

## 7.3 Applications

One example of application of the project is Hide-n-seek, where a known object is to be found somewhere in a room where the object may or may not be in the field of view. Another example is follow the leader, where the Nao should first learn to recognize a person and then follow him as he moves around.

An application of the classification could be sorting fruits, where the robot is given a pile of fruit and sort them by type.

# Appendix A

## Robot hardware specification

The project group will be using a Nao V4 with body type H25 from Aldebaran. The robot has the following sensors and actuators:

- 2x cameras, model MT9M114 (details: http://doc.aldebaran.com/1-14/family/robots/video_robot.html)
- 2x Ultrasonic sensors (http://doc.aldebaran.com/1-14/family/robots/sonar_robot.html)
- 1x Inertial Unit (IMU) (http://doc.aldebaran.com/1-14/family/robots/inertial_robot.html)
- Buttons (http://doc.aldebaran.com/1-14/family/robots/contact-sensors_robot.html):
    - 3 on the head
    - 1 on the stomach
    - 2 on each foot
    - 3 on each hand
- 4x Force Sensitive Resistors (FSR) on each foot (http://doc.aldebaran.com/1-14/family/robots/fsr_robot.html)
- LEDs (http://doc.aldebaran.com/1-14/family/robots/leds_robot.html):
    - 8x RGB in each eye
    - 10x blue in each ear
    - 1x RGB in each foot
    - 1x RGB in stomach
- 2x 940nm IR sensors (http://doc.aldebaran.com/1-14/family/robots/infrared_robot.html)
- 4x microphones (http://doc.aldebaran.com/1-14/family/robots/microphone_robot.html)
- 2x speakers (http://doc.aldebaran.com/1-14/family/robots/loudspeaker_robot.html)
- WiFi: 2.4GHz (5GHz is not supported)
- Processor: ATOM Z530 1.6GHz CPU
- RAM: 1GB
- Secondary memory: 2GB (expandable with USB memory stick)
- Battery:
    - Old battery: 27.6 Wh (http://doc.aldebaran.com/1-14/family/robots/battery_robot.html)
    - New battery: 48.6 Wh (http://doc.aldebaran.com/2-1/family/robots/battery_robot.html)
- 25x Servos (http://doc.aldebaran.com/1-14/family/robots/motors_robot.html):
    - 22NT82213P
    - 17N88208E
    - 16GT83210E

## Robot software specifications

The Nao is running Aldebaran API 1.14.5 (http://doc.aldebaran.com/1-14/index.html). The project group will be using Nao_bridge (http://wiki.ros.org/Nao) from Freiburg's Humanoid Robots Lab to do the connection with ROS, together with the ROS framework developed by IDA at Linköping University.

## External dependencies

Nao_bridge: http://wiki.ros.org/Nao
ROS Indigo: http://wiki.ros.org/indigo
LiU Humanoids: https://gitlab.ida.liu.se/groups/liu-humanoids
OpenCV 2.4.8: http://opencv.org/opencv-2-4-8.html
Operating system offline computer: Ubuntu 14.04 (http://releases.ubuntu.com/14.04/)
Operating system Nao: NaoQi OS (https://community.aldebaran.com/en/resources/software)
Python version: 2.7
Visualization tool: rviz (http://wiki.ros.org/rviz)
Record and play video and sensor data: rosbag (http://wiki.ros.org/rosbag)

## Other Links

http://doc.aldebaran.com/1-14/family/robots/video_robot.html
http://doc.aldebaran.com/1-14/Naoqi/vision/alvideodevice-indepth.html