Introduction to IT
Software Development

David Eklov

Division of Computer Systems
Department of Information Technology
Uppsala University
david.eklov@it.uu.se

September 29, 2010
Class Overview

- This is a project course:
  - work independently in groups of four
  - design, build and program a LEGO robot
  - 1hp ≈ 1 week of full time studies ≈ 40 hours

- One lecture (today)
  - Introduction to software development

- One lab (next week)
  - Help you get started with your project

- Class webpage
  - General information
  - http://www.it.uu.se/edu/course/homepage/sts/ht2010

- Office hours
  - Ask questions and get help
  - Schedule is on webpage

- Examination
  - Project report and demo
1. What is a Computer?
   - What is Hardware?
   - What is Software?

2. Lego NXT

3. Tutorials

4. Project Report

5. Project Schedule

6. Demo
What is a Computer?

**Hardware**
- Intel Core™ i7
- RAM modules
- Hard drive
- Network interface card

**Software**
- Windows Vista
- Linux
- Firefox
- Microsoft Office
What is Hardware?

Central Processing Unit (CPU)
- The heart of the computer
- Performs simple computations (+,-,*,/)
- Moves data (e.g. from harddrive to memory)

Main Memory
- Temporary data storage
- Data is lost when the power is turned off
- Fast, Small and Expensive

Harddrive
- Permanent data storage
- Data is retained when the power is turned off
- Slow, Large and Cheap
What is Software?

"Computer software [...] is the collection of computer programs and related data that provide the instructions telling a computer what to do." – wikipedia.org

- **Program**
  - Instructs the computer (CPU) what to do
  - Typically stored in files on hard drive

- **Data**
  - The data that the program is processing
  - File on hard drive, read from keyboard, read from network card
What is a (Computer) Program?

- A (computer) program is a sequence of instructions telling the CPU what to do.
- The CPU only understands ones (1) and zeros (0).
  - Instructions are coded as a string of ones and zeros.
- This makes programming hard.
  - 0100001110011100010100011100110011...
- High level languages:
  - Easier for humans to understand and write.
  - Translated by a compiler to a program that the CPU understands (ones and zeros).
  - Examples: C, C++, C#, Java, Python, ML...
1 What is a Computer?
   - What is Hardware?
   - What is Software?

2 Lego NXT

3 Tutorials

4 Project Report

5 Project Schedule

6 Demo
Lego NXT Hardware

- Computer LEGO Brick (middle)
- Three Servo Motors
- Four Sensors
  - Two Touch Sensors
  - One Ultrasonic Sensor
  - One Light/Color Sensor
The Lego NXT (high level) Programming Language

- Graphical programming language
  - Instructions are represented by boxes
  - Integrated development environment (IDE)
  - Compile and download program with a single mouse click
Basic instructions
- Rotate motor
- Read sensor value
- Timers

Programming Abstractions
- Variables
- Loops
- Branches (If-statement)
Variables

Store data for later use

Use as counter

- Count how many times the robot hits a wall

Store a sensor value

- Read and save light sensor now, read it again 10s later and compute difference – did it get darker?

Example

```c
int x; // Create variable x
x = 5; // Set x to 5
int y; // Create variable y
y = x; // Set x to y (5)
```
Loops

Repeats its "body"

- Loop while a condition is true

Example

```java
int x = 0;
while (x <= 5) {
    x = x + 1;
}
```
Branches

Choose between alternative paths

- Which path to execute is determined by a condition

Example

```c
int x, y;
x = 3;
if (x == 2) {
y = 5 * x;
} else {
y = x / 7;
}
```
1. What is a Computer?
   - What is Hardware?
   - What is Software?

2. Lego NXT

3. Tutorials

4. Project Report

5. Project Schedule

6. Demo
The LEGO NXT IDE has built-in tutorials – "Robot Educator" – that we highly recommend.
All tutorial have a *building guide* and a *programming guide*

There are three categories of tutorials

- Common palette
- Complete palette
- (Data logging)

To get you started, we recommend do (at least) the tutorials listed on the course webpage before the lab session

Start by building the basic robot described in the booklet that comes with the lego box – ”Driving Base” (DB)

In the tutorials you will add various sensors to the DB robot

You can use these sensors for your own project (so do not tear them down unless you need to bricks for something else)

The help menu in the bottom right corner of the IDE is very helpful. Hover the mouse over a block to get help
1. What is a Computer?
   - What is Hardware?
   - What is Software?

2. Lego NXT

3. Tutorials

4. Project Report

5. Project Schedule

6. Demo
Sections:

- Introduction
- Functional Requirements
- Test Plan
- Architecture / Design
- Test Report
Introduction

- Short summary of what you will build
- No longer than half a page

Example

The robot should follow a straight line of a length of up to 1m drawn on a flat and level surface and stop at an end marker. After having been placed on the line, it should be started manually (by pressing a button) and should then follow the line and stop at the end marker where it stays in rest until it is turned off.
Requirements Analysis

“Requirements analysis in [...] software engineering, encompasses those tasks that go into determining the needs or conditions to meet for a [...] product, taking account of the possibly conflicting requirements of the [...] users.” – wikipedia.org

Example

Assumptions The line is black on a white surface and is 1 cm thick and 1m in length. The end marker is a red circle with a radius of 10cm placed with its center on the line.

Requirement 1 The robot should follow the line for at least 1 m after having been placed centered over the line facing in the direction of the line towards the end marker.

Requirement 2 The robot should come to a full stop at a distance within 5cm from the end marker.
During development of the robot, you need to test that it works as expected.

Need to design test cases that cover all the robot’s functionalities.

**Example**

**Purpose** Verify requirement R1

**Test Setup** After placing the robot on the line facing in the direction of the line, start the robot by pressing the start button.

**Passing Requirement** The robot successfully follows the line for at least 1m.
Describe the architecture / design you have chosen to fulfill the functional requirements

This section is written in free form, i.e. you can write it any way you like
Summary of the results of test results

Example

**Test 1**
Passed 5 out of 5 tries.

**Test 2**
Passed 4 out of 5 tries. Failure occurred when the robot was placed at a distance of 15cm from the end marker.
Project Schedule

- Before the lab
  - Do the recommended tutorial (see course webpage)
  - Read the documentation guide (see course webpage)
  - Write the first to sections of the project report ("Introduction" and "Functional Requirements")
  - Build your robot

- During the lab
  - Show your project report to the teaching assistant
  - Start programming your robot

- After the lab
  - Write the "Test Plan" section
  - Write the "Architecture" section
  - Finnish programming your robot
  - Test the robot one last time
  - Write the "Test Report" section
Demo