

CMP 108: Programming for Non-Majors

Lecture Notes

2 February 2007

Prof. Katherine St. John

Lehman College & the Graduate Center

City University of New York

Overview

- First Day Details (web pages)
- Opportunities
- Class Overview
- Introduction to Computers
- Checking out & Building Robots

First Day Details

See course webpages.

Opportunities for Students in Technology-Related Majors

- Computer Science & Mathematics
Mentorship & Scholarship Program
- Internships (On-Campus & Industry)
- Research Assistantships

Computer Science & Mathematics Mentorship & Scholarship Program



- Provides scholarships, a laptop mentoring and career planning
- Majoring in technology-related field
- Good grades in first year courses
- US Citizen or permanent resident



Funded by the National Science Foundation.

Internships (On-Campus & Industry)

Some recent internships:



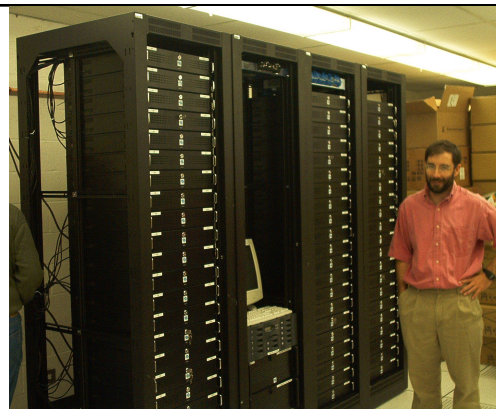
- ABC Television
- Einstein School of Medicine
- FAA– La Guardia
- IBM Research
- NYC MTA
- National Medical Library
- New York Public Library

Research Assistantships

A few examples:



Computational Biology:
Tree of Life Project

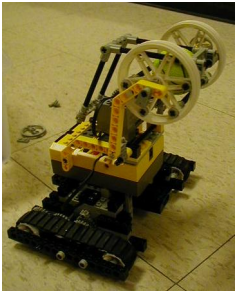


Wildebeest Cluster
(132 processors)



Morphometrics
with AMNH

Course Overview



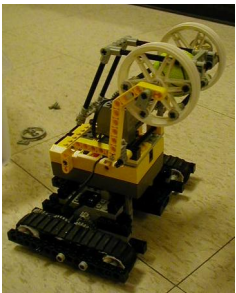
- Part I: Lego Robots



- Part II: Animation Programming (Alice)

Course Overview

More formally:

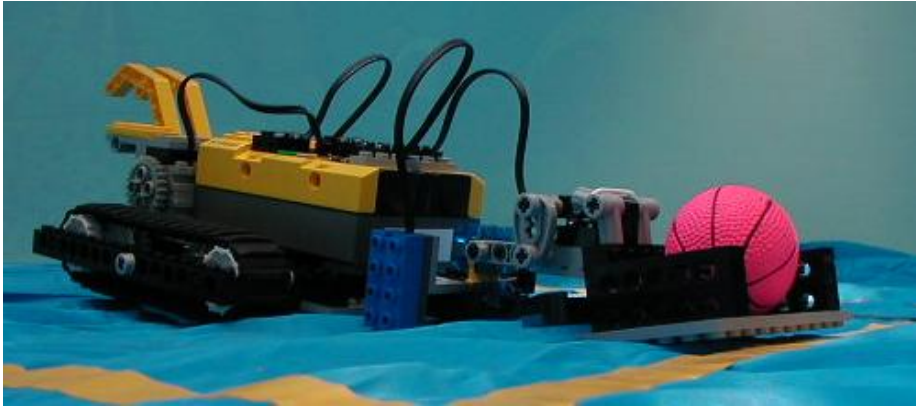


- Part I: Structured Programming



- Part II: Object Oriented Programming (Alice)

Introduction to Computers & Robots



- What is a computer?
- What is a program?
- Compiling Programs

What is a computer?

A basic computer has a

- central processing unit (CPU) or “brain”,
- registers to keep track of next instruction & where data is stored
- Two kinds of memory:
 - Read Only Memory (ROM): Often contains directions that programs the CPU and can't be erased,
 - Random Access Memory (RAM): Used to store firmware and programs the computer runs.

What is a computer?

There's also input devices:

- keyboard, mouse, scanner,...

and output devices:

- screen, printer, speaker,...

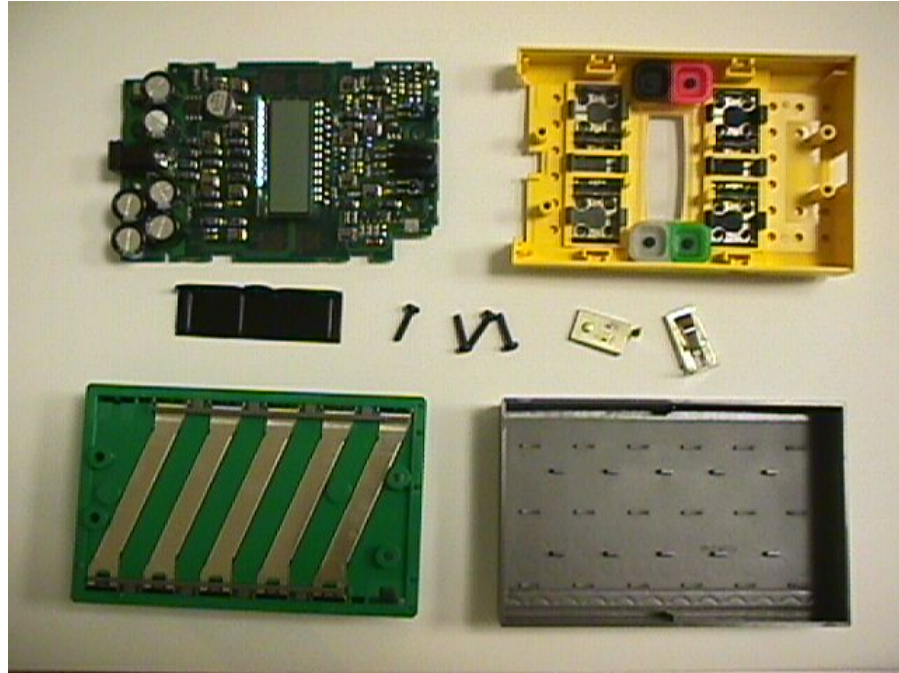
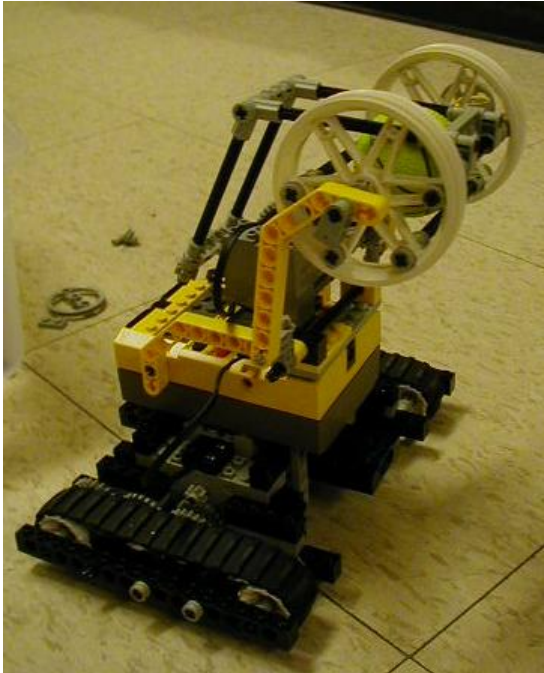
Lego Mindstorm Robot



The lego robot has:

- 8-bit CPU (in bulky lego block, called the RCX)
- 16K Internal ROM & 32K static RAM (6K for programs)
- Input Devices: 2 touch sensors, 1 light sensor, I/R port
- Output Devices: 2 motors, screen, speaker, I/R port

Lego Mindstorm Internals



(Pictures of RCX internals from: <http://graphics.stanford.edu/kekoa/rcx/>)

CPU Directions

- Most CPUs understand directions written in **machine language**— strings of 0's and 1's
- Each instruction corresponds to an “operation code” or **opcode** that consists of commands like:
‘ ‘Increment value in register AX’ ’
- Very hard to write program in machine language.
- Most programs are written in a **high level language**, like Java, Visual Basic, C or C++.

Programming

- The general process is:

You write a program that looks like English (with lots of rules) \Rightarrow “compiling” \Rightarrow Gives a binary file the computer can understand \Rightarrow You “run” the binary to execute the program

- A **program** is a set of instructions for the computer to follow.
- Programs implement **algorithms**— step-by-step directions for performing a task (ex: a recipe to make cookies, directions to make the robot turn 360°).

Not Quite C

- For the robot, we're going to write programs in a variant of C, called **Not Quite C** (NQC).
- Legos come with a language called RCX– it's very simple, but doesn't allow you to do a lot.
- By using NQC, you can do more sophisticated programs and it will help you learn C/C++ and Java.

A Simple Program

```
// tankbot1.nqc - drive straight ahead
```

```
#define LEFT OUT_A
```

```
#define RIGHT OUT_C
```

```
task main()
```

```
{
```

```
    On(LEFT+RIGHT);
```

```
    until(false);
```

```
}
```

Some Useful NQC Commands

Command	Definition	Example
<code>On(<i>outputs</i>)</code>	turn on outputs	<code>On(LEFT+RIGHT);</code>
<code>Off(<i>outputs</i>)</code>	turn off outputs	<code>Off(LEFT+RIGHT);</code>
<code>Fwd(<i>outputs</i>)</code>	sets to forward direction	<code>Fwd(LEFT);</code>
<code>Rev(<i>outputs</i>)</code>	sets to reverse direction	<code>Rev(RIGHT);</code>
<code>Wait(<i>time</i>)</code>	wait for time $\frac{\text{time}}{100}$ seconds	<code>Wait(100);</code>

(Much more on this in the next two lectures.)

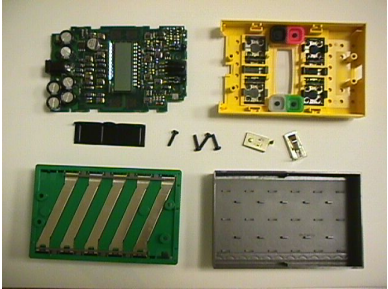
Checking Out & Building Robots



- Robots are stored in Gillet 137.
- After a short break, need a few volunteers to help transport them to our classroom.

Checking Out & Building Robots

Each student will get:

-  RCX brick



- Small plastic shoebox to store assembled robot
- Labels to place on the robot RCX and the shoebox.

Checking Out & Building Robots

- Everyone will put a robot together and test the 5 built-in programs, described in the Lego instruction manual.
- You need to build a car-like robot with a bumper and light sensor, either of your own design, or the Roverbot or Acrobot from the *Constructopedia*.
- We will begin cleaning up at 12:15pm and class ends at 12:30pm
- If you do not finish today, you may come during my office hours (W 11-1, F 1-2) or when Gillet 137 suite is open (most T,W,Th 1-5).