Development of a System for Teaching CS1 in C/C++ with Lego™ NXT Robots

A. Delman, A, Ishak, M. Kunin, L. Goetz,
Y. Langsam, T. Raphan

Department of Computer and Information Science,
Brooklyn College of CUNY

Open Source Software

- Code::Blocks
  - www.codeblocks.org
- Ubuntu
  - www.ubuntu.com/
- nxtOSEK
  - lejos-osek.sourceforge.net/
- Oracle VM VirtualBox
  - www.virtualbox.org

Brooklyn College

- Middle-Lower class income
- 17000 students
- CIS Department
  - BS in Computer Science
  - BS in Multimedia Computing
  - BS in Information Systems (Joint with Economics)
  - ~270 majors
  - Coordinated Engineering Honors Program (with Polytechnic University, City College of New York School of Engineering, and the College of Staten Island Engineering Science Program)

CIS 1.5 Introduction to Programming Using C++

1.5 Introduction to Computing Using C++

5 hours; 4 credits

Algorithms, computers, and programs. Writing, debugging, and testing programs. Loops and conditional control structures. Functions and parameter passing. Arrays, strings, and simple classes. Sorting, Searching, and other basic algorithms. Input and output. Programming applications selected from various disciplines. History and basic concepts of computer science.

- Freshman course – no background assumed
- Multiple sections
- 15-25 students per section
- Reserved section for STEM students
- Typical section web page: http://eilat.sci.brooklyn.cuny.edu/cis1_5/CISClassPage.htm

Goals of Project

- Teach C++
- Target STEM students
- Increase motivation
- Increase retention
- Increase relevancy
- Introduce advanced concepts

Method For Achieving Goals

- Utilize robot programming (motivation)
- Science exercises for STEM majors (relevancy)
- Consolidate a simple professional level IDE that students can use throughout careers (retention)

Prior Art

- Mini Language Approach
  - Turtle Graphics (Logo)
  - Karel the Robot
  - Others (Wayfarer, Turingal and Tortoise)
  - MindStorms Lego™ robot
    - immediate visual feedback
    - visual debugging
    - ultimate goal of a program is to accomplish a task
    - introduces STEM students to sensory-motor based computer control

Disadvantages

- Mini languages delay introduction to mainstream programming languages (e.g. C++, Java)
- Relevance to student goals is unclear
- May require semi-expensive hardware
- May be platform dependant
- Do not present a clear path to advanced concepts

Previous Contribution

- CPlusVEBot (C++ Virtual Environment Robot)
  - A. Delman, L. Goetz, Y. Langsam, and T. Raphan, "*Development of a System For Teaching C/C++ Using Robots in the first CS1 Course Using Open Source Software*", Proceedings of the WORLDCOMP'09 - The 2009 World Congress in Computer Science, Computer Engineering, and Applied Computing

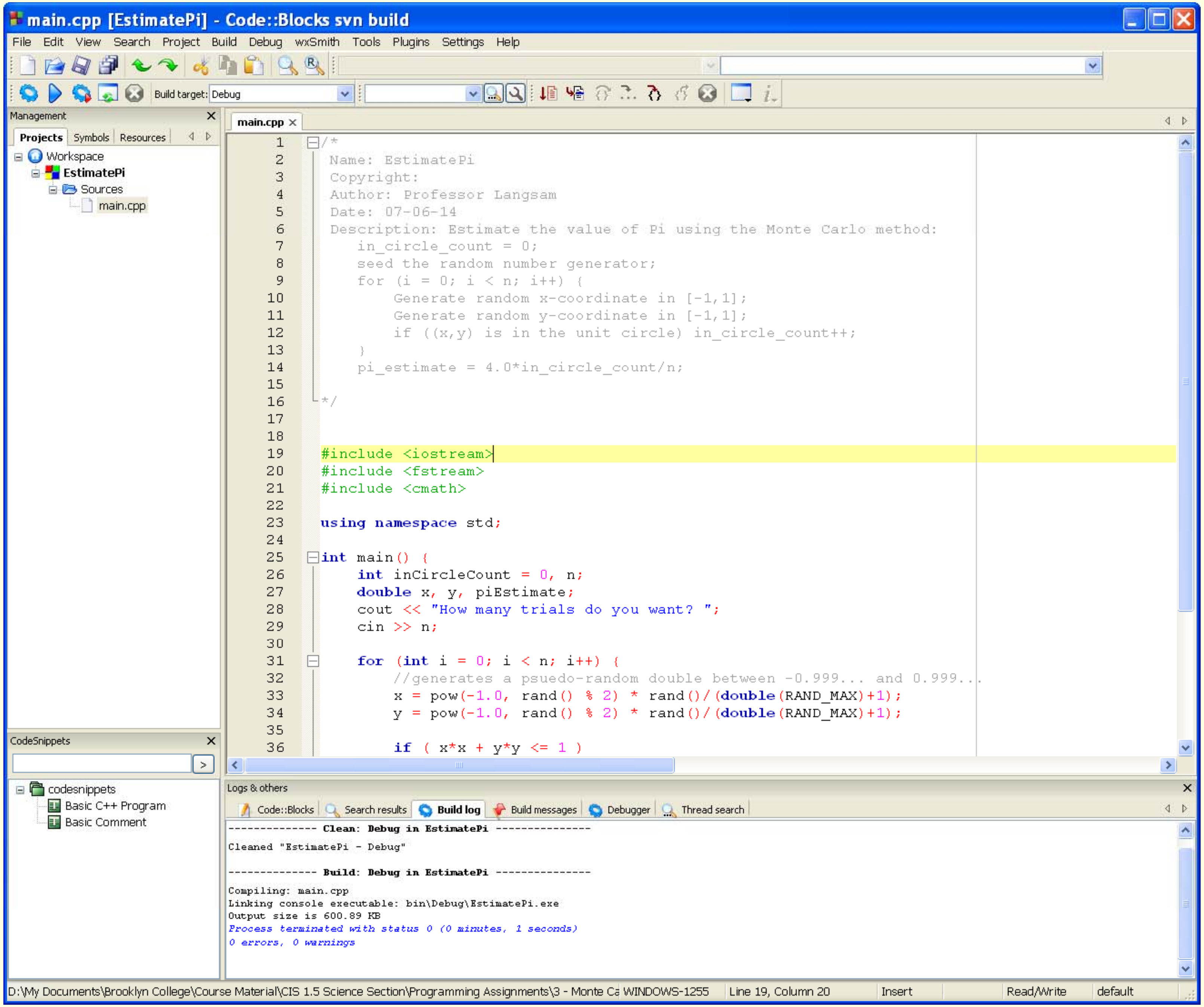
  - However the RCX is underpowered and 1st generation

Upgraded System Components

- Code::Blocks IDE
- The Lego ™ NXT Brick
- nxtOSEK - C/C++ API
- NXT simulator
  - developed by us
- C++ Robot Library (robot.h)
  - developed by us to allow for standard C/C++ syntax
- Oracle VirtualBox

Code::Blocks IDE

- Open Source IDE
- Runs across all platforms
- Active development and user community
- Adds uniformity to the student learning experience
- Flexibility to add compilers, debuggers, and emulators to the environment
- New version (10.05) released


main.cpp [EstimatePi] - Code::Blocks svn build
File Edit View Search Project Build Debug wxSmith Tools Plugins Settings Help
Build target: Debug
Management
Projects Symbols Resources
Workspace
EstimatePi
Sources
main.cpp
CodeSnippets
codesnippets
Basic C++ Program
Basic Comment
Logs & others
Code::Blocks Search results Build log Build messages Debugger Thread search

```
/*
 Name: EstimatePi
 Copyright:
 Author: Professor Langsam
 Date: 07-06-14
 Description: Estimate the value of Pi using the Monte Carlo method:
    in_circle_count = 0;
    seed the random number generator;
    for (i = 0; i < n; i++) {
        Generate random x-coordinate in [-1,1];
        Generate random y-coordinate in [-1,1];
        if ((x,y) is in the unit circle) in_circle_count++;
    }
    pi_estimate = 4.0*in_circle_count/n;

*/


#include <iostream>
#include <fstream>
#include <cmath>

using namespace std;

int main() {
    int inCircleCount = 0, n;
    double x, y, piEstimate;
    cout << "How many trials do you want? ";
    cin >> n;

    for (int i = 0; i < n; i++) {
        //generates a psuedo-random double between -0.999... and 0.999...
        x = pow(-1.0, rand() % 2) * rand()/(double(RAND_MAX)+1);
        y = pow(-1.0, rand() % 2) * rand()/(double(RAND_MAX)+1);

        if ( x*x + y*y <= 1 )
```

```
-------------- Clean: Debug in EstimatePi ---------------

Cleaned "EstimatePi - Debug"

-------------- Build: Debug in EstimatePi ---------------

Compiling: main.cpp
Linking console executable: bin\Debug\EstimatePi.exe
Output size is 600.89 KB
Process terminated with status 0 (0 minutes, 1 seconds)
0 errors, 0 warnings
```

D:\My Documents\Brooklyn College\Course Material\CIS 1.5 Science Section\Programming Assignments\3 - Monte Ca WINDOWS-1255 Line 19, Column 20 Insert Read/Write default

The Lego ™ NXT Brick

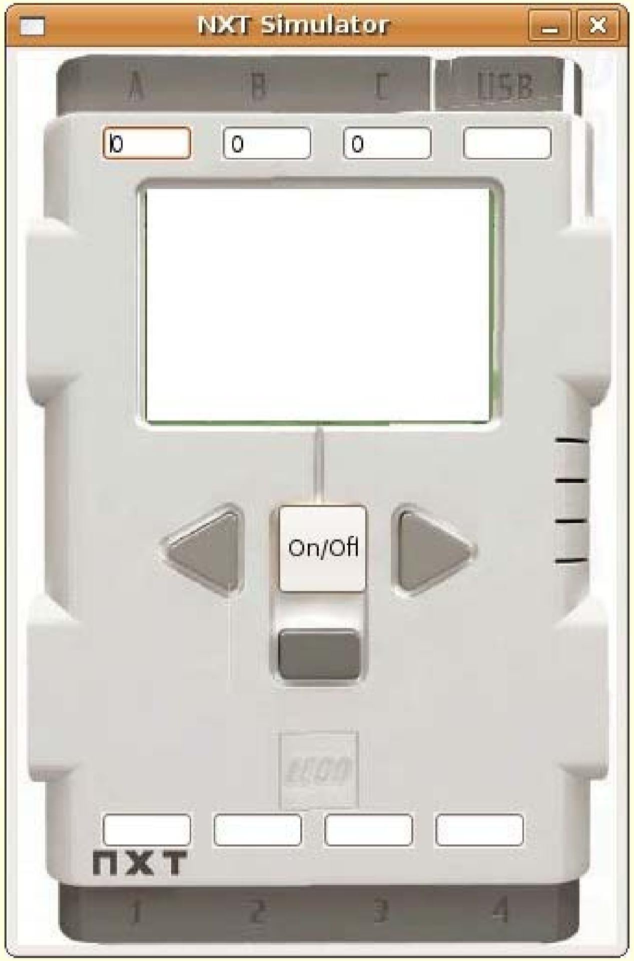
- Receives a program via USB & Bluetooth communications
- Allows for output on an LCD screen
- Four sensor ports
- Three motor ports

- Supports inter-robot communication
- Numerous sensors available

NXT

NXT Simulator

- Open source Linux based simulator
- Status of motors and sensors displayed
- Allows for user interaction



nxtOSEK

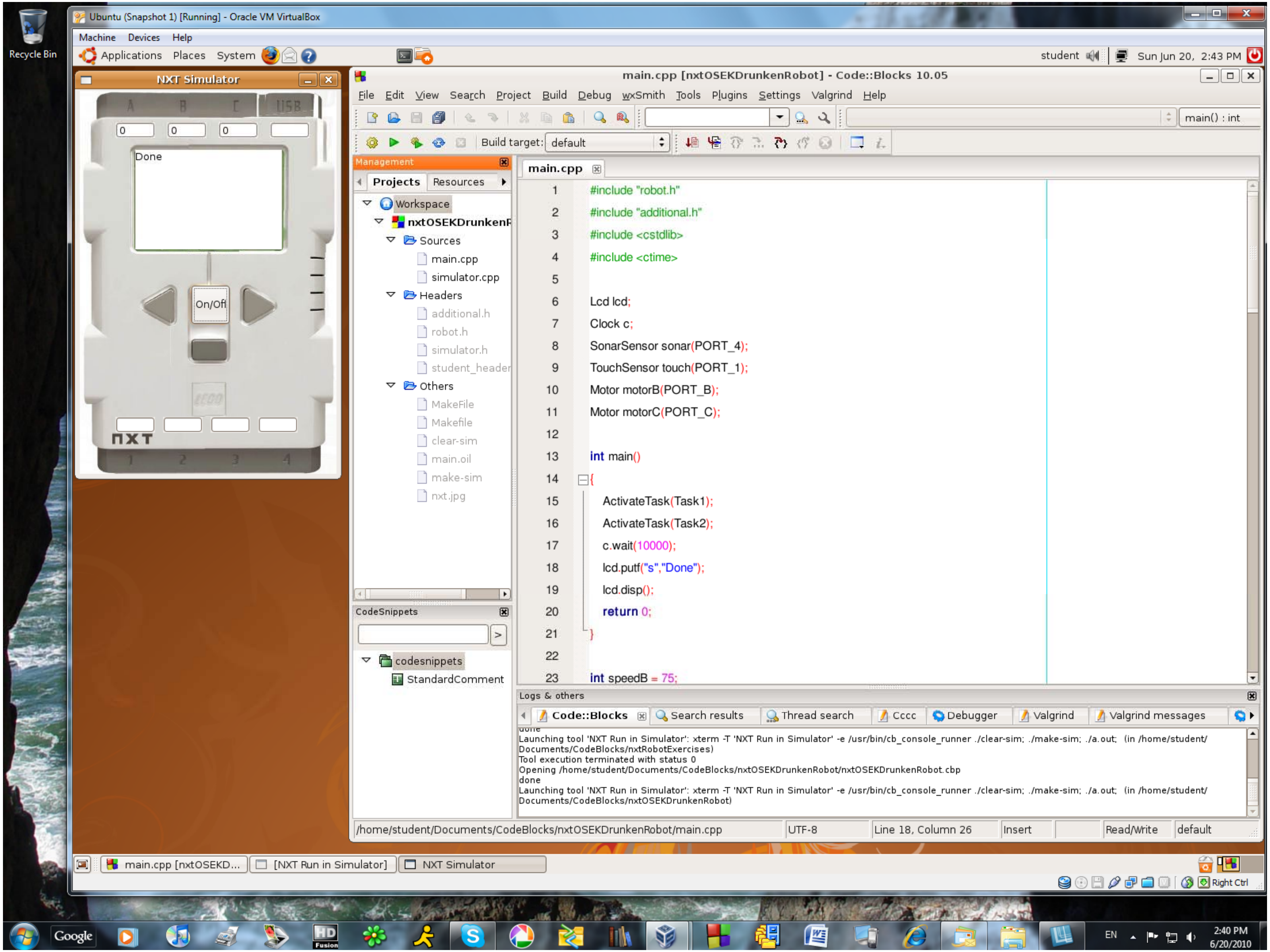
- *John Hansen's Enhanced Firmware* is downloaded into the brick
- nxtOSEK is an open source platform for LEGO Mindstorms NXT
- nxtOSEK provides:
  - ANSI C/C++ programming environment by using GCC tool chain
  - C++ API for NXT sensors and motors, including many third party sensors
  - TOPPERS (Toyohashi OPen Platform for Embedded Real-time Systems), which provides real-time multi-tasking for an nxtOSEK application

Oracle VirtualBox

- Freely available
- Supports multiple hosts (Windows XP, Vista, Windows 7, MacOS, Linux)
- Supports multiple clients
- Provides a consistent user interface
    - Each student works on a common virtual machine (Ubuntu Linux client)

Integrated Environment

- GNU ARM toolchain cross-compiler is integrated into Code::Blocks
- nxtOSEK APIs are integrated into the Code::Blocks IDE
- Provides similar programming experience with console-based environment
- Tools (plugins) for downloading firmware and compiled code into simulator or robot (via USB or Bluetooth) are integrated into Code::Blocks.



```
#include "robot.h"
#include "additional.h"
#include <cstdlib>
#include <ctime>

Lcd lcd;
Clock c;
SonarSensor sonar(PORT_4);
TouchSensor touch(PORT_1);
Motor motorB(PORT_B);
Motor motorC(PORT_C);

int main()
{
    ActivateTask(Task1);
    ActivateTask(Task2);
    c.wait(10000);
    lcd.putf("s","Done");
    lcd.disp();
    return 0;
}

int speedB = 75;
```

```
done
Launching tool 'NXT Run in Simulator': xterm -T 'NXT Run in Simulator' -e /usr/bin/cb_console_runner ./clear-sim; ./make-sim; ./a.out;  (in /home/student/
Documents/CodeBlocks/nxtRobotExercises)
Tool execution terminated with status 0
Opening /home/student/Documents/CodeBlocks/nxtOSEKDrunkenRobot/nxtOSEKDrunkenRobot.cbp
done
Launching tool 'NXT Run in Simulator': xterm -T 'NXT Run in Simulator' -e /usr/bin/cb_console_runner ./clear-sim; ./make-sim; ./a.out;  (in /home/student/
Documents/CodeBlocks/nxtOSEKDrunkenRobot)
```

/home/student/Documents/CodeBlocks/nxtOSEKDrunkenRobot/main.cpp | UTF-8 | Line 18, Column 26 | Insert | Read/Write | default

System Configuration

- CPlusVEBot (C++ Virtual Environment Robot)
  - Preconfigured system (DVD) allows students to install the entire system on a wide range of platforms, including Windows XP/Vista/Windows 7 and MacOS, automatically
  - Code::Blocks IDE and the robotics based tools are installed directly on an Ubuntu client
  - Student manual for CPlusVEBot and Code::Blocks IDE
  - nxtOSEK Reference

- Online scripts are available to install the entire system on any existing Intel-based Ubuntu system

Pedagogical Protocol

- Students are introduced to C++
- Science based programming HWs assigned
- Lego Robot and nxtOSEK specific statements are introduced
- Robot based programming HWs assigned to be done at home
- Students test and modify their programs in the laboratory

Previous Classroom Experience Using RCX

- Students successfully installed system on their home machines
- Students developed simple *Robot Exercise* program using C++ including functions, parameters and loops as their third programming exercise
- Students developed *Brownian Motion* aka *Drunken Robot* program using threads, random, and sensor IO as their fifth programming exercise

Previous Classroom Experience Using RCX

- Student excitement was high
- Students experimented in lab by modifying their programs
- Students requested additional homework

Expected Outcomes Based on Experience with RCX Robots

- Strong positive influence on
  - science, technology, engineering, and mathematics (STEM) based instruction
  - basic computer science instruction
- Motivate students to trace and debug their programs
- Gives the student experience with
  - Sophisticated IDE
  - Linux (No matter what their native platform)
  - Programming for embedded systems
  - Virtualization
  - Interplay between simulation and real systems

Major Contributions

- Developed an Integrated Environment to teach Programming in C++, Utilizing NXT robot.

- Developed header files for maintaining C/C++ program conventions - necessary because of the task oriented programming environment of nxtOSEK, which was based on OSEK OS (embedded system programming OS specification)

- Developed a Simulator for programming NXT robot in C++.

Further Work

- Develop a variety of programming assignments using C++ and the Lego Robot
- Enhance Simulator Capabilities
- Introduce CPlusVEBot into CS2

References

- Code::Blocks Student Manual
  - http://www.sci.brooklyn.cuny.edu/~goetz/codeblocks/codeblocks-instructions.pdf
- nxtOSEK
  - http://lejos-osek.sourceforge.net/
- This Presentation
  - http://eilat.sci.brooklyn.cuny.edu/WorldComp10/WorldComp10Presentation.pdf