Development of a System for Teaching C/C++ Using Robots and Open Source Software in a CS1 Course

A. Delman, L. Goetz, Y. Langsam, T. Raphan

Department of Computer and Information Science, Brooklyn College of CUNY
Open Source Software

- Code::Blocks
  - www.codeblocks.org

- BrickEMU
  - http://hoenicke.ath.cx/rcx/

- Sun VirtualBox
  - www.virtualbox.org
Brooklyn College

- Middle-Lower class income
- 16000 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
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
System Components

- Code::Blocks IDE
- The Lego ™ RCX Brick
- BrickEMU
- BrickOS
- Sun 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
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 pseudo-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)
The Lego ™ RCX Brick

- Receives a program via infrared communications
- Allows for output on an LCD screen
- Three sensor ports
- Three motor ports
BrickEMU

- Open source Linux based emulator
- Status of motors and sensors displayed
- Allows for user interaction
- Remote control emulator also available
BrickOS

- BrickOS firmware is downloaded into the brick
- Supports standard C++
- BrickOS cross-compiler is integrated into the Code::Blocks IDE
- Provides similar programming experience with console-based environment
- Compiled code downloaded by Code::Blocks into robot or emulator.
Sun VirtualBox

- Freely available
- Supports multiple hosts (Windows XP, Vista, MacOS, Linux)
- Supports multiple clients
- Provides a consistent user interface
  - Each student works on a common virtual machine (Ubuntu Linux client)
```cpp
#include "core++.h"
#include "robot++.h"
#include "sensor++.h"
#include "oled++.h"
#include "button++.h"

using namespace std;

int main() {
    // Display "Hello".
    oled.println("Hello!");
    sleep(5);
    // Display "World!"
    oled.println("World!");
    sleep(5);
    // Roll forward and backward.
    robot.roll();
    while (true) {
        // Display "Hello" whenever sensor 2 is activated.
        if (sensor[2].isTouched())
            oled.println("Hello! ");
        else if (sensor[1].isTouched())
            oled.println("World! ");
    }
}
```
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 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
  - BrickOS Reference
Classroom Experience

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

- 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.
- Student excitement was high.
- Students requested additional homework.
Student’s YouTube Video
Conclusions & Further Work

- 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
Conclusions & Further Work

- Extend CPlusVEBot for the Lego™ NXT robot
- Develop additional programming assignments using C++ and the Lego Robot
References

- Code::Blocks Student Manual

- CPlusVEBot Student Manual
  - http://eilat.sci.brooklyn.cuny.edu/cis1_5/Programming%20the%20LEGO.pdf

- This Presentation
  - http://eilat.sci.brooklyn.cuny.edu/WorldComp09/WorldComp09Presentation.pdf