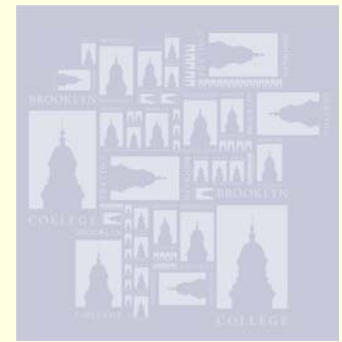
The background features a repeating pattern of the Brooklyn College logo, which includes a silhouette of a building with a dome and the text 'BROOKLYN COLLEGE'. The pattern is arranged in a grid-like fashion, with some elements slightly offset or faded to create a sense of depth and movement.

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

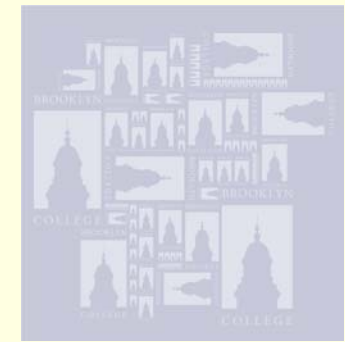
- BrickEMU

- <http://hoenicke.ath.cx/rcx/>

- Sun VirtualBox

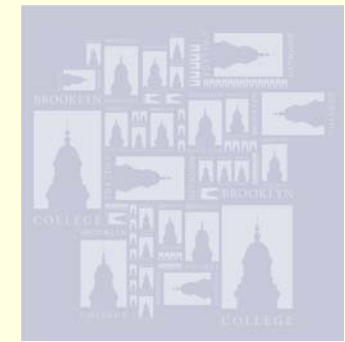
- www.virtualbox.org

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++

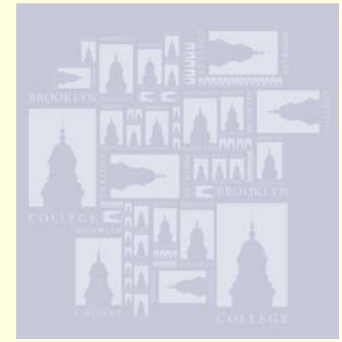
5 hours; 4 credits

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.

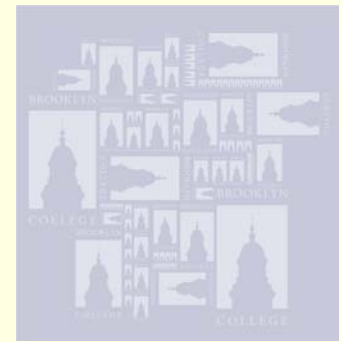
- Freshman course – no background assumed
- Multiple sections
- 15-25 students per section
- Reserved section for STEM students
- Typical section web page: http://eilat.sci.brooklyn.cuny.edu/cis1_5/CISClassPage.htm

Goals of Project

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

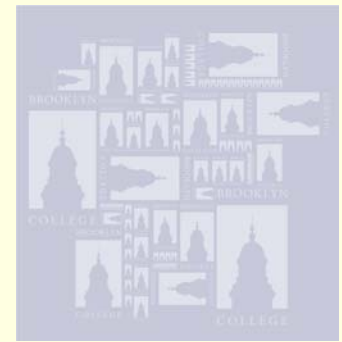


Prior Art



- 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
-

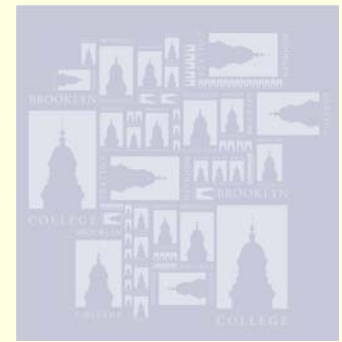
Disadvantages



- 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 TM 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
-

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

```
1  /*
2  Name: EstimatePi
3  Copyright:
4  Author: Professor Langsam
5  Date: 07-06-14
6  Description: Estimate the value of Pi using the Monte Carlo method:
7      in_circle_count = 0;
8      seed the random number generator;
9      for (i = 0; i < n; i++) {
10         Generate random x-coordinate in [-1,1];
11         Generate random y-coordinate in [-1,1];
12         if ((x,y) is in the unit circle) in_circle_count++;
13     }
14     pi_estimate = 4.0*in_circle_count/n;
15 */
16
17
18
19 #include <iostream>
20 #include <fstream>
21 #include <cmath>
22
23 using namespace std;
24
25 int main() {
26     int inCircleCount = 0, n;
27     double x, y, piEstimate;
28     cout << "How many trials do you want? ";
29     cin >> n;
30
31     for (int i = 0; i < n; i++) {
32         //generates a psuedo-random double between -0.999... and 0.999...
33         x = pow(-1.0, rand() % 2) * rand() / (double)(RAND_MAX)+1;
34         y = pow(-1.0, rand() % 2) * rand() / (double)(RAND_MAX)+1;
35
36         if ( x*x + y*y <= 1 )
```

CodeSnippets

codesnippets

Basic C++ Program

Basic Comment

Logs & others

Code::Blocks Search results Build log Build messages Debugger Thread search

Clean: Debug in EstimatePi

Cleaned "EstimatePi - Debug"

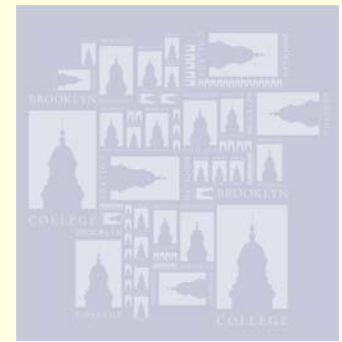
Build: Debug in EstimatePi

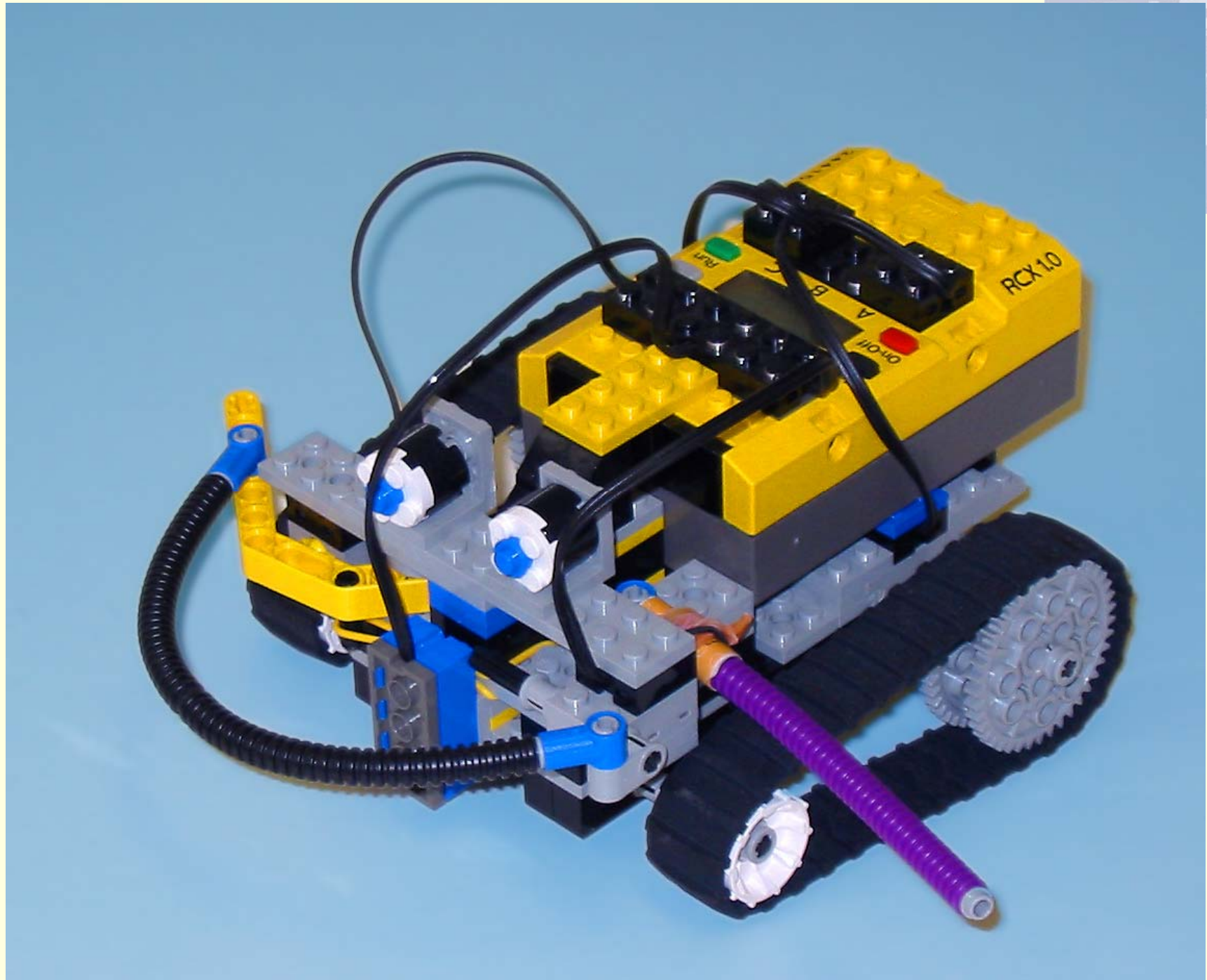
Compiling: main.cpp
Linking console executable: bin\Debug\EstimatePi.exe
Output size is 600.89 KB
Process terminated with status 0 (0 minutes, 1 seconds)
0 errors, 0 warnings

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

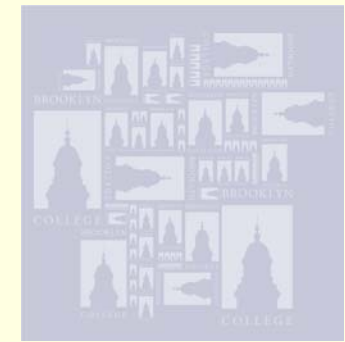
The Lego™ RCX Brick

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

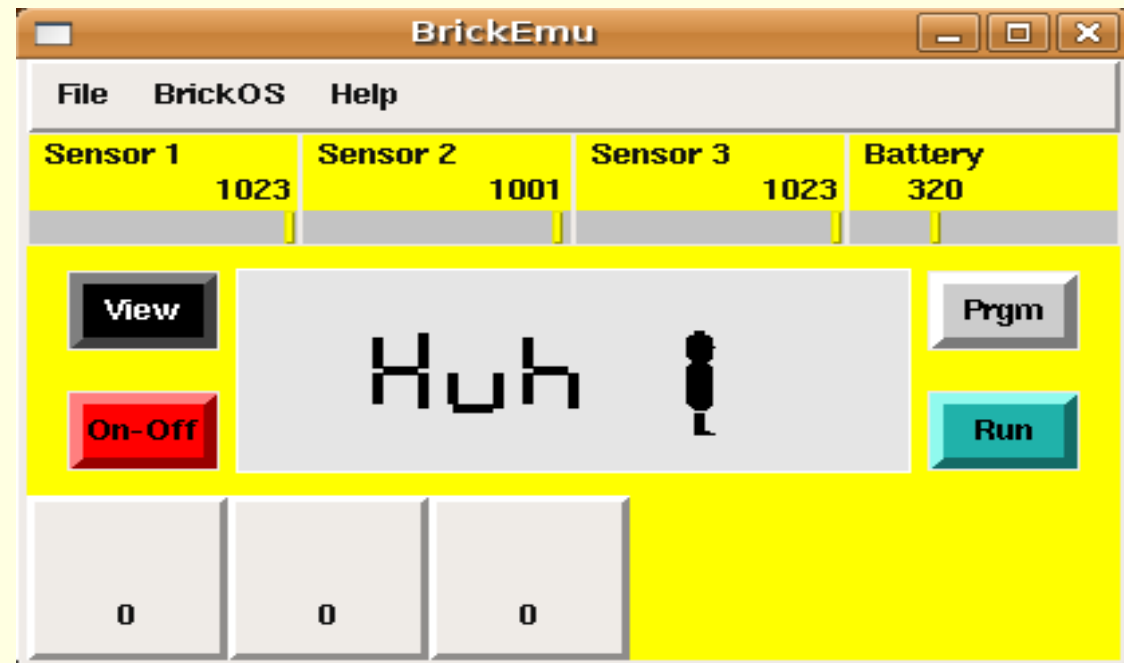




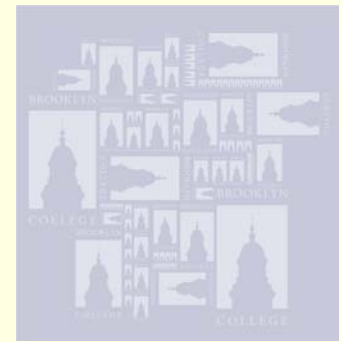
BrickEMU



- 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)
-

Ubuntu (Snapshot 1) [Running] - Sun xVM VirtualBox


Machine Devices Help Applications Places System student Sun Mar 1, 10:59 AM

BrickEmu

File BrickOS Help

Sensor 1	Sensor 2	Sensor 3	Battery
1023	1023	1023	320

View



Prgm Run

0 0 0

main.cpp [HelloRobotWorld] - Code::Blocks 8.02

File Edit View Search Project Build Debug wxSmith Tools Plugins Settings Help

Build target: Debug

main(): int

```

1  /* Name: HelloRobotWorld
2  Description: A simple program that illustrates the features of BrickOS.
3  The program will:
4  1. Display "Hello World" on the LCS
5  2. Roll forward and then back
6  3. Wait until the either Sensor 1 or Sensor 2 is "touched"
7  */
8
9  // Include files containing the BrickOS library functions
10 #include<conio.h>
11 #include<motor.h>
12 #include<sensor.h>
13 #include<unistd.h>
14
15 using namespace std;
16
17 // Function prototypes
18 void hello();
19 void world();
20 void roll();
21
22 int main() {
23
24     hello();           // Display "Hello"
25     sleep(5);         // Pause for 5 seconds
26     world();          // Display "World"
27     sleep(5);         // Pause for 5 seconds
28
29     roll();           // Roll forward and backwards
30
31
32     // Display "HMM" until such time as either Sensor 1 or Sensor 2 is activated.
33     // Display "OUCH" whenever sensor 2 is activated.
34     // Display "BYE" and end the program when sensor 1 is activated.
35     while (1) {
36         if (TOUCH_2)
37             cputs("OUCH ");
38         else if (TOUCH_1) {
39             cputs("BYE ");
40             break;

```

Management

Projects Resources

- Workspace
 - HelloRobotWorld
 - Sources
 - main.cpp

Logs & others

Code::Blocks Search results Thread search Debugger Build log Build messages

Running startup script
 ScriptFunction 'edit_startup_script' registered under menu '&Settings/Edit startup script'
 Opening /home/student/Documents/CodeBlocks/helloworld/helloworld.cbp
 done
 Opening /home/student/Documents/CodeBlocks/HelloRobotWorld/HelloRobotWorld.cbp
 done

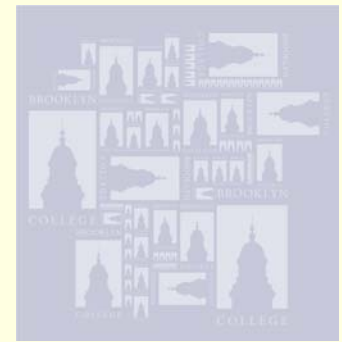
/home/student/Documents/CodeBlocks/HelloRobotWorld/main default Line 29, Column 48 Insert Read/Write default

main.cpp [HelloRobo... BrickEmu

Right Ctrl



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



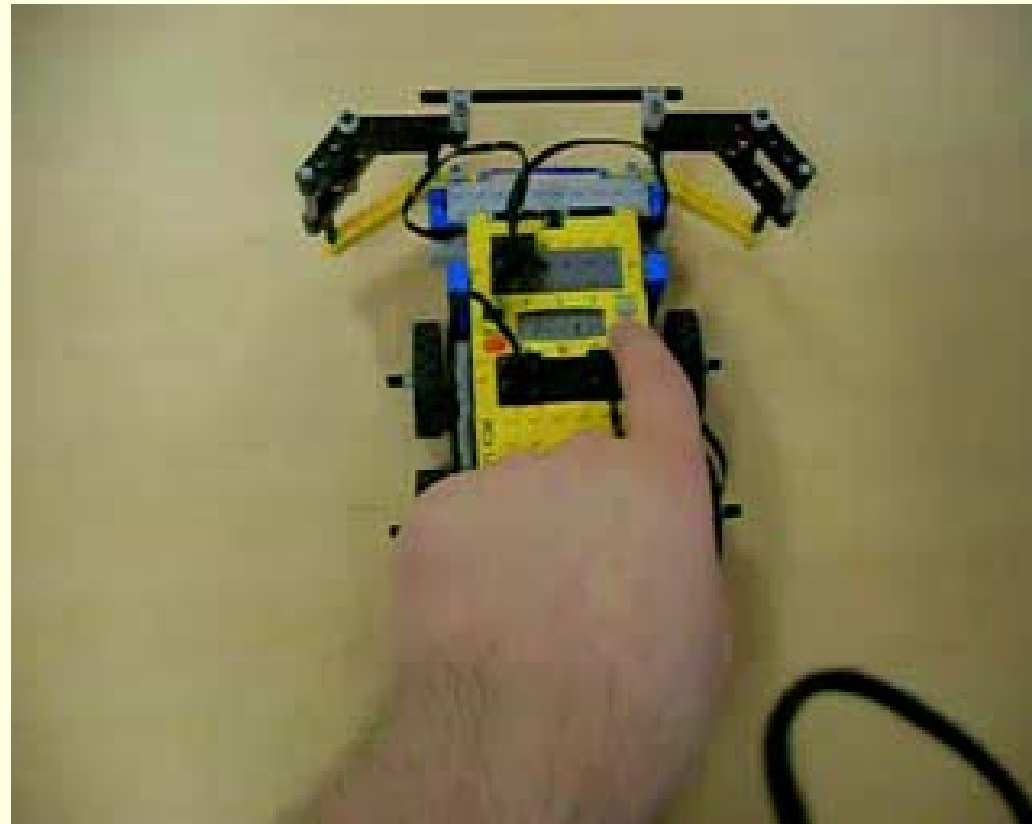
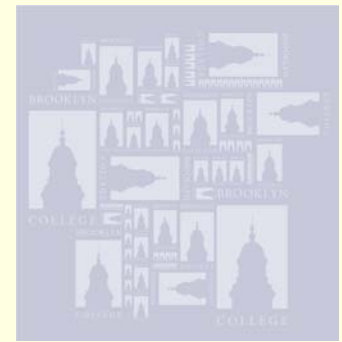
- Students are introduced to C++
- Science based programming HWs assigned
- Lego Robot and BrickOS specific statements are introduced
- Robot based programming HWs assigned to be done at home
- Students test their programs in laboratory

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



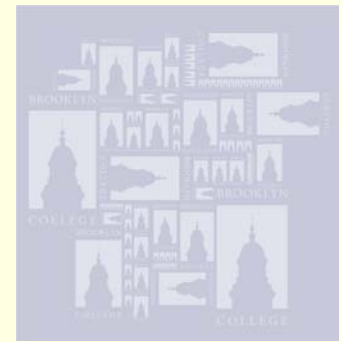
—

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

- <http://www.sci.brooklyn.cuny.edu/~goetz/codeblocks/codeblocks-instructions.pdf>

- 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>