by Jesse Flot, Computer Scientist, CMU; Dr. Chris Schunn, Cognitive Scientist, the University of Pittsburgh; Allison Lui, Graduate Student, the University of Pittsburgh; and, Robin Shoop, Director of CMU's Robotics Academy

Learning How to Program via Robot Simulation

Robot Virtual Worlds (RVW) opens doors for learning robotics

Researchers at Carnegie Mellon University (CMU) conjecture that a robotic simulation environment will prove to be a better tool to teach computer science principles than a physical robot. It is unlikely that robotic simulation will convey all key aspects of engineering, electronics, and design, but it may be a better tool for teaching robot programming. This article describes the collaborative research between CMU and the University of Pittsburgh (PITT), the results of prior research when teachers have used physical robots to teach programming, the results of our pilot study of learning with simulated robots, new tools available to students learning how to program, and how you can participate in our upcoming studies.

CAN ROBOTS BE USED TO TEACH NOVICE PROGRAMMING?

Many teachers have brought robots into their computer science classrooms, and there have been numerous studies designed to measure the effectiveness of using robots to motivate and promote computer science education. According to a recently published review of research in this area, 74% of studies[i] find that robots are an effective tool for teaching introductory computer science principles. *The review also suggested that simulation environments may increase robotics' ability to motivate learning computer science by removing some of the difficulty associated with physical robots (e.g. cost, setup time, space).* However, there has not yet been much research investigating robotics simulation as a tool for teaching computer programming.

Table 1

The Ideal Simulation Environment Should Teach:

- Algorithmic Thinking
- Programming Syntax, Statements, and Structures
- Robot Mathematics
- Controlling Motors and Servos
- Using Sensor Feedback
- Boolean Algebra/Conditional Operators
- Variables/Passing Parameters
- Functions
- Programming User Interfaces
  - Buttons,
  - Joysticks
  - Digital Displays
- Debugging Programs

Over the last year, CMU and Robomatter Inc. have developed an inexpensive, high-end robotic simulation software named Robot Virtual Worlds (RVW). RVW is a multiple hardware platform (LEGO, VEX, Arduino), engaging simulation environment that enables students without access to physical robots to practice programming with the same motivational effects as students that do have access to robots. The RVW environment has been embedded into curricula designed to teach the competencies listed in Table 1. CMU is now working collaboratively with PITT along with formal and informal educators to measure the effectiveness of the new RVW simulation environment.

WHAT IS A RVW?

RVW enables students to use code written in the ROBOTC programming environment to control simulated robots in physics enabled digital worlds. ROBOTC can be used to program Arduino, LEGO, and VEX educational robots. It is also a legal programming language for VEX, BEST, FTC, Robofest, and the National Robotics Challenge robotics competitions. ROBOTC is a complete development environment and includes a powerful runtime debugger; it has free curriculum and support at the ROBOTC website and is used in over 8,000 schools. A key feature of ROBOTC and RVW is that students can use the exact same code to control both simulated and physical robots. There are nearly one hundred physics-enabled RVW simulated environments including: typical classroom tabletop activities (Figure 1), competition environments (Figure 2), and game-like fantasy environments (Figure 3). These features enable researchers to setup parallel studies and compare the results. Study one will extend promising pilot results of learning using the ROBOTC curriculum with the RVW simulation software; study two will compare RVWs and physical robots using a shared ROBOTC curriculum.

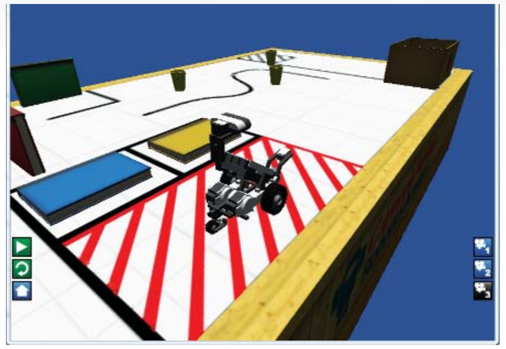


Figure 1.

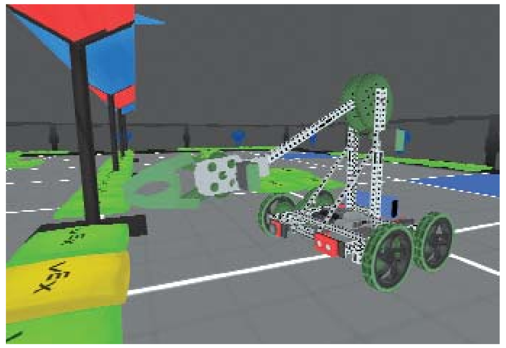


Figure 2.

Figure 3.

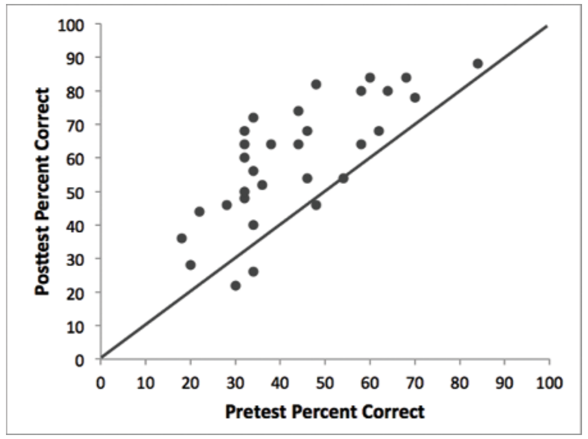
PILOT STUDY, CAN RVW BE USED TO TEACH COMPUTER SCIENCE PRINCIPLES?

In January 2012, the research team recruited two middle school teams that recently participated in CMU's FLL competition to be part of a RVW pilot study; the research team wanted to know if RVW could be used to teach novice programming skills. Neither teacher in the study had experience using ROBOTC. Students in the study used a curriculum that was integrated into a Learning Management System (LMS) that is part of the DARPA-sponsored Computer Science Student Network (CS2N). Students used a combination of the RVW tabletop simulations and the fantasy based Palm Island programming environment to learn basic programming. Each student was required to complete a 50-question pretest prior to the course, and an identical 50-question post-test at the end of the course. Across the two classes, 31 students completed both pretest and post-test.

LEARNING IMPROVEMENTS

There was a statistically significant improvement from pretest scores to post-test scores in both schools, with an average increase of 16 points out of 100. Even though there were differences in the kinds of schools, with differing teaching styles and student backgrounds, students across these settings were equally likely to show performance improvements (just below 16 points in one school and just above 16 points in the second school).

The scatterplot below of pretest scores versus post-test scores suggests that equal learning occurs across all performance levels. Any point on the y=x line indicates a student with no gain from pre to post, and all points above the line are students showing gains. The majority of students showed evidence of learning after taking the course, and the extent of learning did not vary based on their pretest scores. That is, students who already knew a fair amount of programming gained about the same as students who had almost no prior programming skills.



Comparison chart

WHAT TYPES OF QUESTIONS DID WE ASK?

The questions on the pretest and post-test were categorized into four category types (and a few problems involved combinations):

***Robot-Specific Behaviors*:** questions about the physical robot's functions (e.g., "Choose the picture that best illustrates a point turn")

***ROBOTC Syntax*:** questions about programming commands that are used in ROBOTC (e.g., "Which lines in the following code are responsible for controlling how long the robot moves?")

Algorithmic Thinking: questions that require thinking about how to solve a problem or evaluate a program (e.g., "Which example of pseudocode would be most helpful for solving the maze?")

General Programming: questions that involve programming knowledge that is applicable to many other languages, not just ROBOTC (e.g., "Which data type is used to store numbers like 3.1415, 6.234, and 9.323?")

All categories showed significant learning gains, but there was some variation in gains across categories. In general, problems with significant improvement involved knowledge that students gained while writing their own programs, such as programming syntax, whitespace and comment usage, understanding code (what will a given program cause the robot to do), and common variables and operators (such as floats, integers, and logical not).

There was also some noticeable variation among algorithmic thinking problems. Students showed improvements on ROBOTC-specific algorithmic thinking (e.g., what will a given program do), but appeared to have difficulties with broader algorithmic thinking (e.g., how to avoid future flaws in a program, or how to plan a program using pseudocode).

For each question category, the number of problems, the Cronbach ? (how reliable the measure is), the mean % improvement from pre to post, and the effect sizes (Cohen's d; change measured in standard deviation units; greater than 0.8 is considered a large effect) of the pre-to-post improvements:

Data Chart

	Robot-Specific	ROBOTC Syntax	Algorithmic Thinking	General Programming
# of Problems (out of 50)	5	29	17	32
Cronbach α (Reliability)	.61	.85	.77	.84
Mean % Improvement	23%	18%	19%	15%
Effect Size of Improvement (*d*)	0.8	1.1	0.9	1.2

NEW TOOLS AVAILABLE TO MOTIVATE AND ASSESS LEARNING

The research team has recently developed a set of automated web-based tools designed to report student progress as it motivates and assesses student learning (Figure 4); these tools are being integrated into DARPA's CS2N (www.cs2n.org), we are calling this CS2N function a Group. CS2N Groups combines a guided learning experience using an LMS, the RVW software, and automated reporting allowing informal and formal education teachers to track students' progress as they work through the lessons. CMU has iteratively tested and developed groups during their summer teacher professional development courses. This fall over 100 certified teachers will test the CS2N's Groups feature; CMU is offering free training and access to RVW through CS2N to all FTC and VEX robotic competition coaches, when they complete the course they will be able to offer Robotics Academy ROBOTC Certi-fication to their students. During the fall study, PITT and CMU will collect anonymized data to help answer the question "Do students learn to program robots better via simulation than with physical robots?", read below to see how you can become involved in the study.

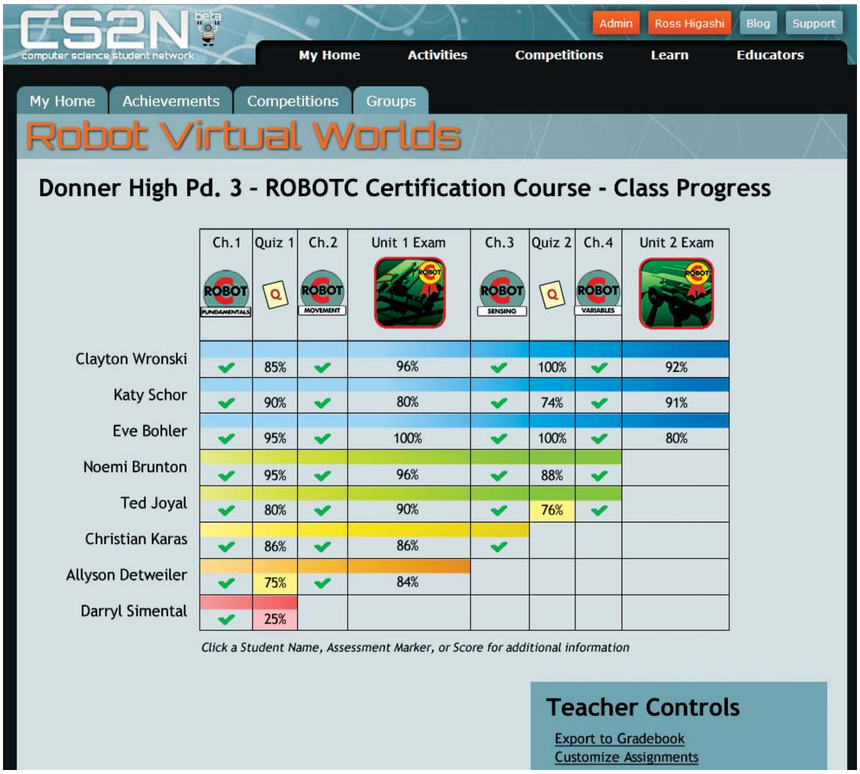


Figure 4, CS2N Groups Interface

HOW CAN YOU PARTICIPATE IN THE NEXT STUDIES?

CMU and PITT are looking for informal and formal education partners to participate in one of two different studies. In Study 1, we want to look for even larger learning gains of virtual-only instruction using the new CS2N Groups tools. In Study 2, we want to investigate the overall learning gains from physical versus virtual robots. Research partners will receive free RVW software, curricular materials, and access to CS2N's LMS. In order to participate, partners will need to commit to the following:

Have their students take the pre and posttests on CS2N

Use the "Groups" feature in CS2N in order to track your students' progress.

Use the CS2N LMS to learn ROBOTC LEGO or VEX programming.

Interested partners can find more information at www.cs2n.org/educators/rvw.

[i] *Systematic Literature Review: Teaching Novices Programming Using Robots*
L. Major, T. Kyriacou and O. P. Brereton, (2011). 15th Annual Conference on Evaluation & Assessment in Software Engineering, p 21-30.