EDU BOTS

Learn to Program in Half The Time!

by Allison Liu, Graduate Student, University of Pittsburgh; Jeff Newsom, Technology Teacher, Penn Trafford High School, Pittsburgh, Pennsylvania; Chris Schunn, Cognitive Scientist, University of Pittsburgh; Robin Shoop, Director, Carnegie Mellon Robotics Academy

Virtual Sims make students smarter

Robotic systems make up a hundred-billion dollar emerging industry. Robots are everywhere, we just don't call them robots. We call them cell phones, bank machines, cars, microwaves, the Internet… Robotic technologies are ubiquitous and are making it easier for humans to drive cars, access money, find restaurants via their cell phone, or cook their food using a microwave. We are in the middle of a robotic revolution.

The brains of robotic systems are driven by Computer Science (CS). CS will play a key role in nearly all future innovation, including advancements across all science, technology, engineering, and mathematics (STEM) fields, yet sadly the U.S. has entered into a significant national decline in the number of college graduates with basic and advanced CS-STEM degrees. This downward trend is particularly pronounced in CS. In 2010, the National Science Board reported that the U.S. does not graduate enough computer scientists to meet its own demand, and therefore has to rely on foreign-born talent and this trend will likely continue.

Robots to the rescue in RVW Operation Reset.

At the high school level, the focus on high stakes testing topics coupled with increased emphasis on Advance Placement courses has squeezed out coursework in many areas including computer science. DARPA's CS2N project recently funded a research study conducted by Carnegie Mellon University's Robotics Academy and the University of Pittsburgh's Learning Research and Development Center, and the results proved to be positive from both a teaching and learning perspective.

Robotics appears to be an activity that excites students to consider CS-STEM careers. Between the FIRST and VEX competitions there are over 20,000 US teams, but our surveys indicate that the majority of the kids on those teams do not consider themselves programmers. The Robot Virtual Worlds (RVW) project team hopes to significantly increase the number of kids on robotic teams that identify themselves as programmers. We believe that early programming environments more like current gaming platforms will be motivational and kids will want to play them and formal and informal education systems will want to use them to introduce students to programming.

In the November 2012 issue of ROBOT Magazine, we asked ROBOT magazine readers if they wanted to participate in our RVW research project where we were going to compare computer programming results of classrooms using real robots versus classrooms learning to program using virtual robots. This article reports on the second phase of that study.

Our team worked with Jeff Newsom, a Tech Ed Teacher from Penn-Trafford High School, on the study. At the end of the project he had this to say about RVWs: "When I first heard about RVWs and how they worked, I knew I wanted to try them. During the study I had one class learning to program using the RVWs and the other using the physical robots; by the end of the study my suspicions were confirmed. The RVWs allowed my students to learn the same amount of material in a lot less time. The RVWs provide immediate feedback when the students are focused on programming. They also provide opportunities for them to work from home if they fell behind, or needed more time to learn a concept, or just wanted to learn more.

"With less time spent on setup, communications, electrical and mechanical problems, and cleanup, the RVW class was able to learn programming much more quickly. My students were surveyed at the end of the class and the students' feedback was very positive. As for the class working with the physical robots, their main complaints were around communication or mechanical problems they ran into while trying to test their programs. Any time I have the opportunity to have my students learn more efficiently, I am going to take advantage of it. Needless to say, RVW will be part of my curriculum for all of my future classes."

A DESCRIPTION OF THE STUDY: PHYSICAL VS. VIRTUAL PROGRAMMING

In our first study we validated that RVW software allowed students to learn basic robot programming. In this study we were interested in studying the difference between classes that learned programming using simulation verse physical robots. We were seeking a school that taught multiple sections of robotics and that would agree to have one section of students learn to program via the RVW technology and the other section using physical robots; once programming was taught both sections would complete the course using physical robots. We selected a local public high-school that taught two elective "introduction to robotics" courses and the same teacher taught both classes. One class completed a ROBOTC programming course using physical VEX robots (the Physical class), while the other class completed a ROBOTC programming course using virtual VEX robots (the Virtual class). Thirteen students were in the Physical class, and 17 students were in the Virtual class. Both classes consisted primarily of freshmen and sophomore students with little to no prior programming experience.

Both the Physical class and the Virtual class completed the same pre-test and post-test online. The pre-test and post-test contained the same 50 items, and both classes completed the pre-test around the same date. Eleven students in the Physical class and 15 students in the Virtual class completed both the pre-test and post-test, and were included in the analyses. Analyses investigated whether there were learning differences between students who interacted with physical robots versus students who interacted with virtual robots.

THE EXPERIMENT

Three analyses were performed on the data. First, students' total scores on the pre-test were compared to their total scores on the post-test. To control for students' differing pre-test scores, an ANCOVA was run using condition (Physical or Virtual) as the independent variable, post-test score as the dependent variable, and pre-test score as the covariate. Second, we examined whether learning differed across topic sub-categories. Four sub-categories were defined, into which all problems on the pre-test and post-test could be placed. These sub-categories were:

Algorithmic thinking: Problems that involved thinking through the process of the programming problem (e.g., planning the program, using pseudocode, predicting what a program would do) or more abstract concepts of programming. Example: "Given the program above, the robot will do __"

General programming: Problems that involved syntax or concepts that are applicable to multiple programming languages. Example: "If the condition of an If statement is true, then all of the code inside of its curly braces will run: True/False"

ROBOTC Syntax: Problems that involved ROBOTC syntax or the ROBOTC program (e.g., how to use menus in the ROBOTC application). Example: "To make the robot stop, you set its motor values equal to __"

Physical Robot: Problems that involved the physical VEX robot's functioning. Example: "The VEX Ultrasonic Rangefinder (sonar sensor) measures distance using __"

The number of problems in each sub-category and the Cronbach's alpha (?; calculated using both conditions' post-test scores) for each category are shown in Table 1. Note that the sub-categories were not mutually exclusive; that is, the same problem could fit into multiple sub-categories.

Problem Sub-Category	Number of Problems	α
Algorithmic Thinking	4	.54
General Programming	13	.56
ROBOTC Syntax	37	.84
Physical Robot	19	.81

Due to the uneven number of problems in each category, we used the proportion of correct answers within each category as a measure of accuracy. An ANCOVA was performed to control for pre-test scores, using condition as the independent variable, post-test score as the dependent variable, and pre-test score as the covariate.

Thirdly, we looked at the amount of days between participants' pre-test attempt and post-test attempt. This was used as a measure of the time needed to complete the course, to see whether one condition required less time than the other to learn the same amount of information. A one-way ANOVA was performed, using condition as the independent variable and number of days as the dependent variable.

THE RESULTS

Overall Scores. No differences were found between the Physical and Virtual class in their overall post-test scores. Both classes began with similar pre-test scores and ended with similar post-test scores. Figure 1 shows that overall learning gain did not differ by pre-test score, as almost all participants improved regardless of their pre-test score. The average pre-test and post-test scores for both classes can be found in Table 2.

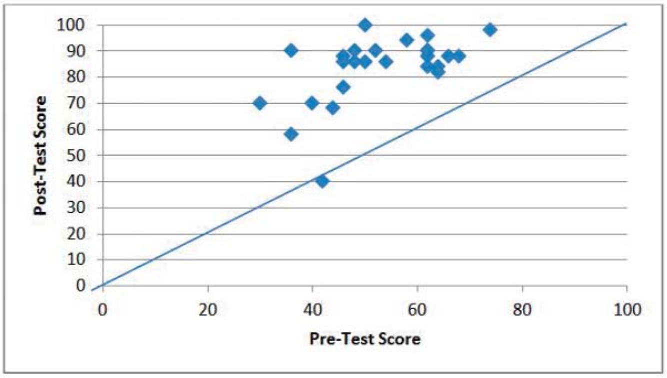


Figure 1. Pre-test score vs. post-test score. Points above the line improved on the post-test compared to pre-test.

Condition	Pre-Test Average	Post-Test Average	Average Time Taken
Physical	50.2 (*SD*=11.2)	82 (*SD*=10.6)	85.0 (*SD*=0.0)
Virtual	55.9 (*SD*=11.5)	84.5 (*SD*=14.6)	54.7 (*SD*=18.2)

Table 2. Averages (and standard deviations) of pre-test score, post-test score, and time taken, separated by condition.

Time Taken. The average time taken for both classes to complete the programming course can be seen in Table 2; the Physical class took significantly more time than the Virtual class. All students in the Physical class completed the course in the same amount of time, as working with Physical robots did not afford them the same freedom of students in the Virtual class, who could work independently through the course. Overall, the Physical class took an extra 30.3 days (approximately one month) to complete the course than the Virtual class (see Figure 2).

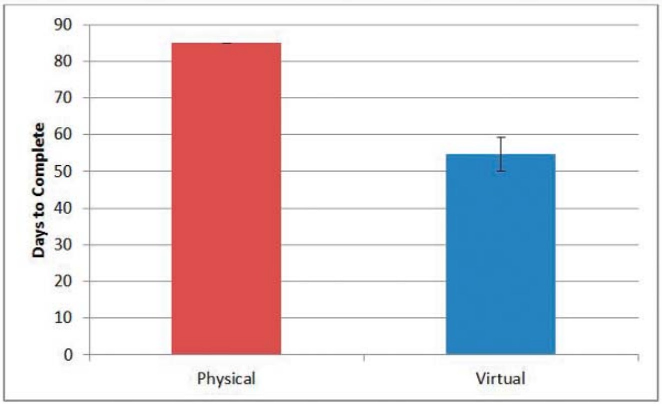


Figure 2. Days taken to complete the course, separated by condition.

SUMMARY

Both the Physical class and the Virtual class showed equal learning gains. The type of learning did not differ between the two classes either, as evidenced by the equal learning gains seen across all four sub-categories. However, the Virtual class did show a time reduction benefit, as they completed the course about a month earlier than the Physical class, with no effect on their overall learning. This suggests that working with the virtual robots allowed students to learn more efficiently in this context when compared to the physical robots.

The teacher's informal observations support this conclusion. The teacher noted that students in the Physical class had to deal with the daily robot setup, additional mechanical issues, and the cleanup that comes with working with a physical robot. Consequently, the teacher spent much more of his time in the Physical class helping students with robot communication, mechanical, and class organization issues. In the Virtual class, he and his students were able to focus 100% of their time on learning programming.

To confirm that the learning gains and time savings seen in the Virtual class were consistent, we also looked at two additional classes who completed the same programming course with virtual VEX robots. A graph comparing the three courses' pre-test scores, post-test scores, and days to complete the course can be seen in Figure 3. The graph suggests that Virtual robots in all three Virtual classes allowed students to complete the course in significantly less time than the Physical class in the study above.

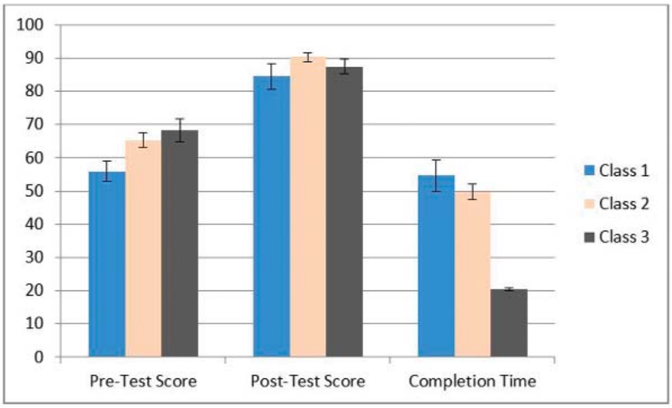


Figure 4. Average pre-test score, post-test score, and days to complete the course, separated by class.

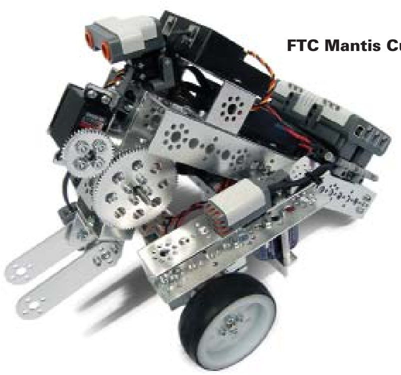
Class	Average Pre-Test Score	Average Post-Test Score	Average Days Taken
Class 2 (N=23)	65.3 (*SD*=10.5)	90.3 (*SD*=6.7)	49.8 (*SD*=12.3)
Class 3 (N=13)	68.3 (*SD*=12.3)	87.4 (*SD*=8.1)	20.5 (*SD*=1.7)

Table 4. Average pre-test score, post-test score, and days taken to complete the course (and their standard deviations), separated by class.

To see a more comprehensive version of the study, please visit: www.cs2n.org/teachers/research. ◎

LEGO Mining Bot.

LEGO Doggy.



FTC Mantis Cutout.

VEX Sensorbot.

Links

Carnegie Mellon Robotics Academy, www.education.rec.ri.cmu.edu, (412) 681-7160

DARPA CS2N, www.cs2n.org

Robot Virtual Worlds Download, www.robotc.net/download/rvw

For more information, please see our source guide on page 80.