Presenters:
Jesse Flot: Research Programmer Carnegie Mellon Robotics Academy
Email: jbflot@nrec.ri.edu

Jason McKenna: Director Educational Strategy Robomatter
Email: jmckenna@robomatter.com
Twitter: @McKennaj72
Developing Computational Thinking through a Virtual Robotics Programming Curriculum
Motivation
Goal of the Study
How are we defining effective?
Study Results to Classroom Application

1. Educational Robotics
2. Effective Tools
3. Curriculum, not just content
Outline

• Background information & Prior Research
• Examination of the Study
  • Overview
  • Key Technologies
  • Results
• Summary and Practical Considerations
• Future Work
• Q&A
Background Information
Key Partners

• Carnegie Mellon Robotics Academy leads development of CS-STEM curricular materials.
• Robomatter leads development of interactive programming tools such as ROBOTC and Robot Virtual Worlds
• The Learning Research & Development Center of PITT serves as project evaluator, observing classroom implementations, conducting surveys, etc.
• Partners recruit classrooms from local school districts and competitions to participate in the CCRC research project.
Prior Related Research
Teaching Math with Robots

- Mobile Robotics Curriculum
- Robots in Motion
- Cognitive Tutor Enabled Robots in Motion

- Abstraction Bridges
- Robot Synchronized Dancing
- Robots in Motion

- NSF Funded Robot Algebra Project
- Expedition Atlantis Math Movement Game
- Robot Virtual World Math Toolkit

- Ruins of Atlantis Math Programming Game

- Robot Math CS2N Badge Pathway
- Boulder Math Web-enabled Math Game
MAPPED BADGES PATHWAY

**LEVEL 3 - Earn the Certification** - Earn Badges + upload artifacts + get teacher endorsements + pass the final exam

Each Badge has different requirements - THIS IS THE “MAPPED BADGE PATHWAY”

**LEVEL 2 - Skill Badges** - Complete activities + upload artifacts + take quizzes + get teacher endorsements and show evidence

**LEVEL 1 - Activity Badges** - Earned in RVW or by completing classroom activities

Can CTP Be Taught in Robotics Classrooms?

• Summary:
  • There is a growing recognition that computer science and computational thinking are new basic skills that all K-12 students must learn.
  • Robotics provides opportunities to integrate and teach programming, engineering design, and mathematics—areas that benefit from computational thinking.

• Flot, J., Friez, T., Schunn, C., Shoop, R., Witherspoon, E. (March 2016) *Can Computational Thinking Practices Be Taught in Robotics Classrooms?*
  Presented at the International Technology and Engineering Education Conference, National Harbor, Washington DC.
Research Study

Overview
Developing Computational Thinking through a Virtual Robotics Programming Curriculum

EBEN B. WITHERSPOON, ROSS M. HIGASHI, CHRISTIAN D. SCHUNN, and EMILY C. BAEHR, University of Pittsburgh
ROBIN SHOOP, The Robotics Institute, Carnegie Mellon University

Computational thinking describes key principles from computer science that are broadly generalizable. Robotics programs can be engaging learning environments for acquiring core computational thinking competencies. However, few empirical studies evaluate the effectiveness of a robotics programming curriculum for developing computational thinking knowledge and skills. This study measures pre-post gains with new computational thinking assessments given to middle school students who participated in a virtual robotics programming curriculum. Overall, participation in the virtual robotics curriculum was related to significant gains in pre- to posttest scores, with larger gains for students who made further progress through the curriculum. The success of this intervention suggests that participation in a scaffolded programming curriculum, within the context of virtual robotics, supports the development of generalizable computational thinking knowledge and skills that are associated with increased problem-solving performance on nonrobotics computing tasks. Furthermore, the particular units that students engage in may determine their level of growth in these competencies.

CCS Concepts: → Applied computing → Interactive learning environments;

Additional Key Words and Phrases: Computational thinking, robotics, programming, K-12, curriculum design

ACM Reference format:
https://doi.org/10.1145/3104982

1 INTRODUCTION
In the last decade, computational thinking has gained a great deal of attention in K-12 computing education. It is typically construed as an essential 21st-century skill that draws on algorithmic thinking and design processes, but especially in ways that may be generalizable across various contexts (Grower and Pea 2013; Wing 2006). In 2011, a committee of computer science (CS) experts, examining the role that CS would play in bringing computational thinking to K-12, broadly

Organization of the Study

- Two Studies
- Single school district for feedback and debugging
- 4 districts and 26 classrooms
- Tracked student progress with three versions of Computational Thinking assessment
- Pre and Post tests
• Structured in 4 units
  • Getting Started, Basic Movement, Sensors, Program Flow
• Provides multi-media driven direct instruction to students
• Scaffolds Programming Concepts and Big Ideas in Computational Thinking in robotics-driven activities
• Includes embedded authentic Formative and Summative Assessment opportunities
CS-STEM Network (CS2N)

• An education platform where students can earn badges and certificates for CS-STEM related skills.
• Create student progress reports for teachers and researchers.
• Progress measured includes:
  • Curricular Material Consumption
  • Computational Artifact Uploads
  • Endorsements
  • Quiz Assessments
    • including pretests and posttests
  • Virtual Activity Badges and Scores
ROBOTC Programming Environment

- C-Based programming language with robotics extensions
- Graphical and text-based modes
- Includes natural language and standard language constructs
- Supports multiple robot platforms popular in education, including robot simulations (RVW)
• Robot Virtual Worlds (RVW) are simulation environments that allow virtual robots to be programmed with the same languages as physical robots
• Virtual robots emulate their real-world counterparts
• As students progress through the curriculum, they can download their code to a physical or virtual robot
A study found that classes using virtual robots learned just as much as classes using physical robots, but completed the course an average of 30.3 days (40%) faster.

Liu, A., Newsom, J., Schunn, C., Shoop, R. *Learn to program in half the time!* *Robot Magazine*, 49-51.
Technology Demo!
Research Study

Results
Key Findings

• “The increasing contextual distance of the items (from robotics) was intended to assess whether participation in the robotics curriculum developed problem-solving strategies that could transfer to non-robotics tasks.”

• “When examining these effects by the amount of progress that students are able to make through the curriculum, however, we observed significantly larger learning gains occurred for groups of students who reach the more content-rich Sensors and Program Flow units. Thus, students were able to learn generalizable skills, despite being embedded in a context that placed strong emphasis on a particular context (i.e., robotics), suggesting that a robotics context can be used in an extended fashion for instruction on computational thinking, rather than just as a short application included within a CS course.”
Summary & Considerations
Limitations

- Lack of a random assignment control group
- Differing implementations
- What supplemental items did the teachers use?
Practical Considerations

• The lack of professional development for teachers to teach higher order programming and its perceived impact on the study
• The challenges of incorporating robot-specific activities into a programming curriculum
Conclusion

• There is a current effort to broaden the scope of CS learning opportunities in K-12
• New technologies and effective curriculum design can facilitate the learning of generalizable computational thinking skills
Attending to Structural Programming Features Predicts Differences in Learning and Motivation

• Summary:
  • Results suggest that explicit instruction in the structural logic of programming may develop deeper transferrable programming knowledge and prevent declines in some motivational factors

• Witherspoon, E., Higashi, R., Schunn, C., Shoop, R (December, 2017)

Attending to Structural Programming Features Predicts Differences in Learning and Motivation in a Virtual Robotics Programming Curriculum
Journal of Computer Assisted Learning, DOI.10.111/jcal.12219
Underlying Motivations Predict Persistence in an Online Course

• Summary:
  • Student persistence is predicted by prior programming knowledge, intrinsic interest in the subject matter, and mastery approach goals.
  • Teacher persistence is similarly predicted by intrinsic interest, but then also by self-identity as a programmer, performance approach goals, and negatively by performance avoidance goals.

• Higashi, R., Schunn, C., Flot, J (May, 2017) Different underlying motivations and abilities predict student versus teacher persistence in an online course. Education Tech Research Dev DOI 10.1007/s11423-017-9528-z
Helping Students Build Conceptual Models

• Summary:
  • Describes how to help students develop a conceptual model of what computing is as they learn to program.
  • This approach moves students away from memorizing code snippets and reserved words, to a conceptual framework that enables them to understand how computers make decisions

• Summary:
  • Model Eliciting Activities create rich tasks for a diverse set of middle, high school, and college classrooms, shown to be critical to thinking and learning in science and engineering.

• Flot, J., Higashi, R., McKenna, J., Shoop, R., Witherspoon, E. (July 2016)
  *Using Model Eliciting Activities to Engage Students in Computational Thinking Practices*
MEAs are a class of problems in which students must develop a “mental model” representing and incorporating key aspects of a given problem scenario in order to reason about it and produce a solution.

- This framing shifts instructional emphasis to conceptual understanding and model-building rather than searching for the “right answer”.
- Mental modeling is a critical component of mathematical thinking and learning that has also been shown to be critical to thinking and learning in science and engineering.

<table>
<thead>
<tr>
<th>Table 1. MEA Design Principle</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Reality Principle</strong> – Can students make sense of the problem based on prior experience?</td>
</tr>
<tr>
<td><strong>Model Construction</strong> – Does the task need students to create a mental model of the solution?</td>
</tr>
<tr>
<td><strong>Model Documentation</strong> – Will the response require students to explicitly reveal how they are thinking about the problem?</td>
</tr>
<tr>
<td><strong>Self-Evaluation</strong> Does the statement of the problem strongly suggest criteria that enables students to judge when their response is complete?</td>
</tr>
<tr>
<td><strong>Model Generalization</strong> Is the model not only good enough for the specific situation, but can be repurposed for other situations?</td>
</tr>
<tr>
<td><strong>Simple Prototype</strong> Is the problem as simple as possible given the instructional goals?</td>
</tr>
</tbody>
</table>