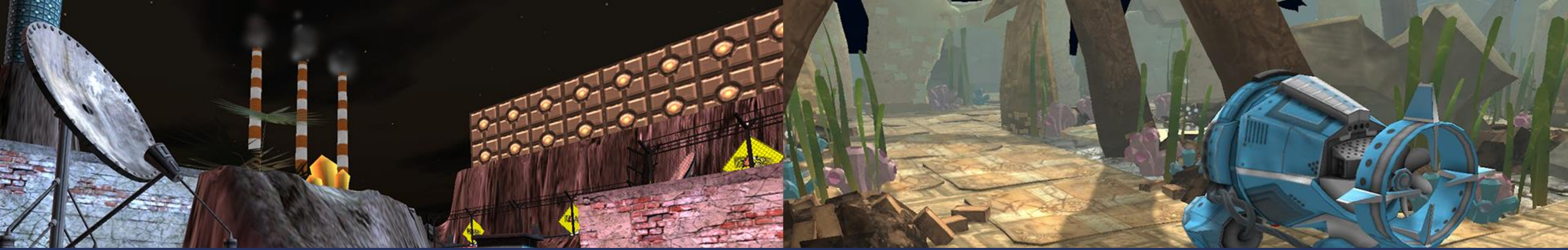
Presenters:

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

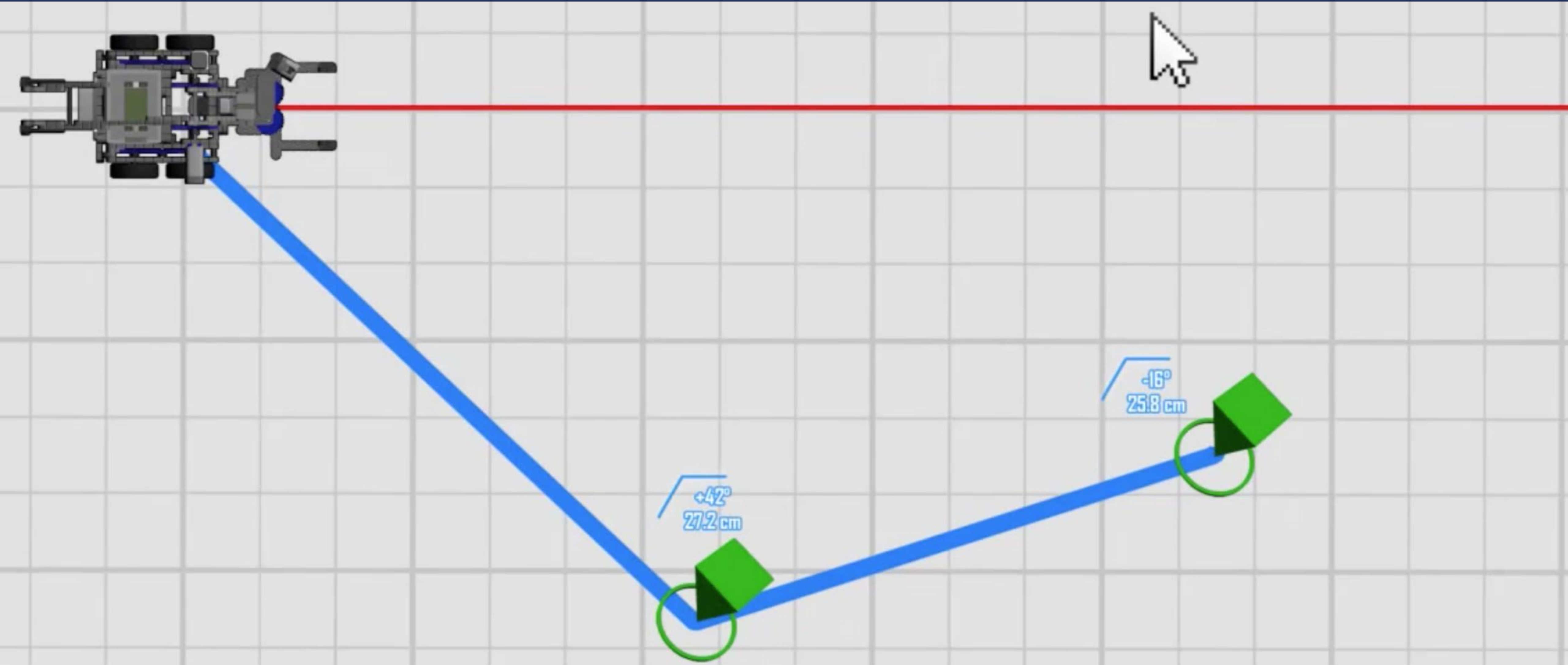
Jason McKenna: Director Educational Strategy Robomatter
Email: jmckenna@robomatter.com
Twitter: @McKennaj72



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

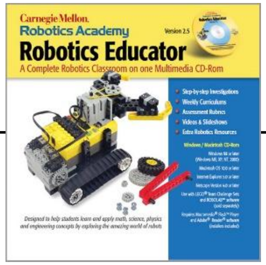
Key Partners

- 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.

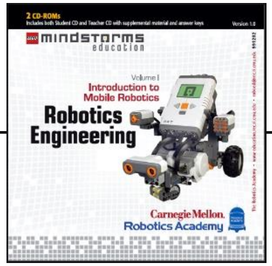
Prior Related Research

Teaching Math with Robots

2002



2005 - 06



Mobile Robotics Curriculum

2007

Abstraction Bridges

2008 - 09

Robot Synchronized Dancing

2009 - 10

Robots in Motion

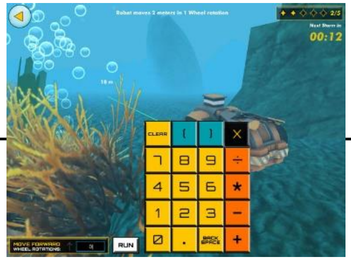
2010

Cognitive Tutor Enabled Robots in Motion

2011

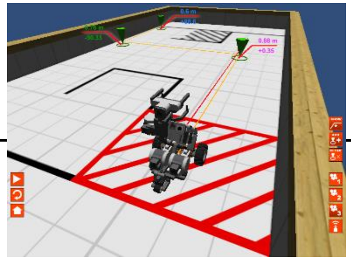
NSF Funded Robot Algebra Project

2012 - 13



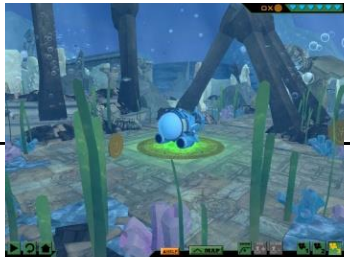
Expedition Atlantis Math Movement Game

2012 - 13



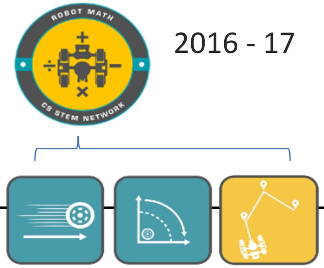
Robot Virtual World Math Toolkit

2014



Ruins of Atlantis Math Programming Game

2016 - 17

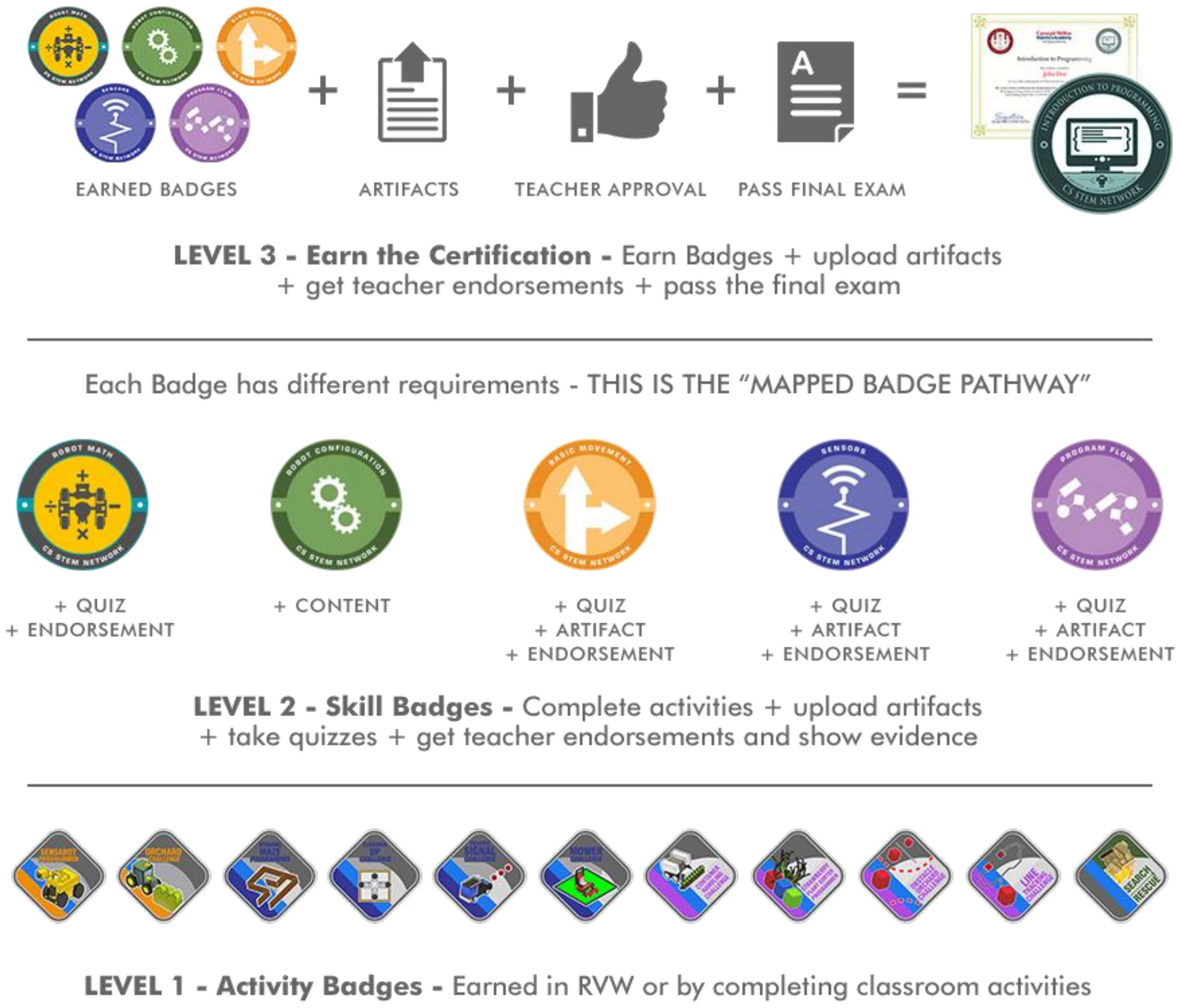


Robot Math CS2N Badge Pathway

2017 - 18

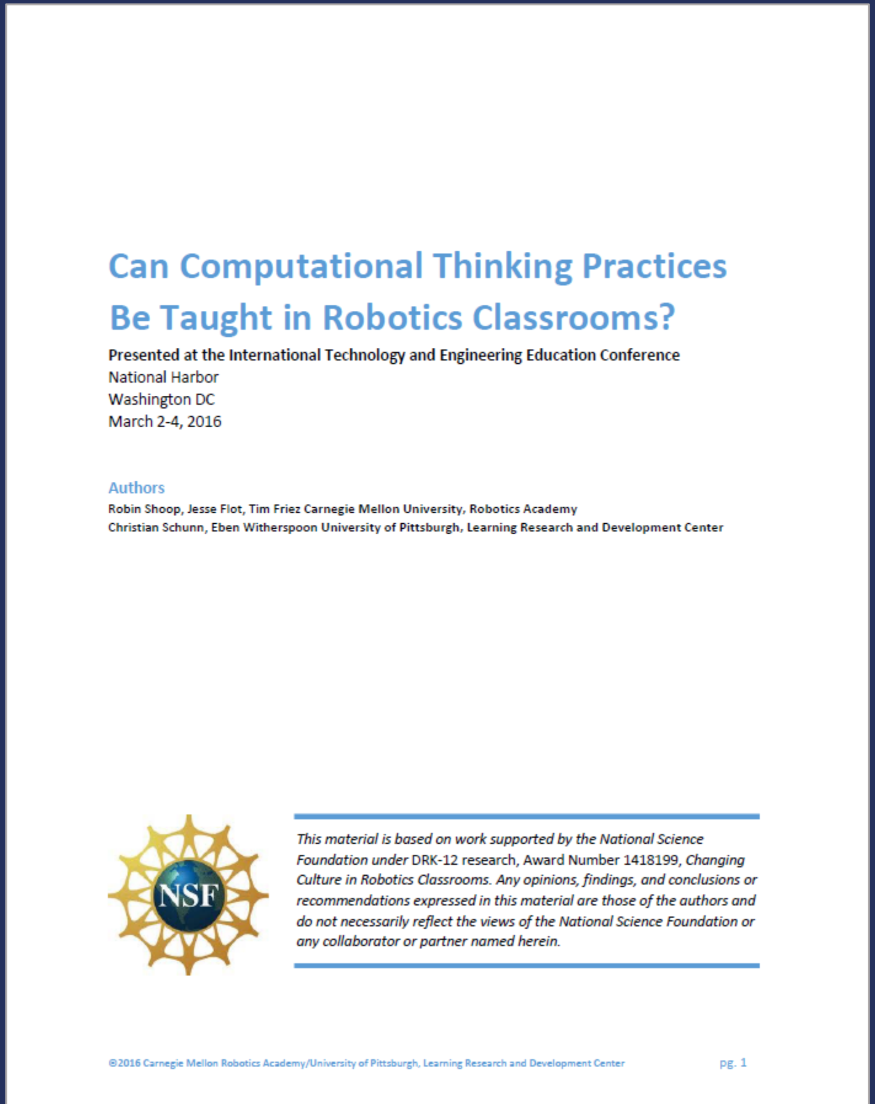
Boulder Math Web-enabled Math Game

Mapped Badged Pathways to Certification



- Abramovich, S., Schunn, C.D., Higashi, R. (2013) ***Are Badges Useful in Education?: it depends upon the type of badge and expertise of Learner.*** Educational Technology Research & Development, March 2013. DOI: 10.1007/s11423-013-9289-2
- Higashi, R., Abramovich, S., Shoop, R., Schunn, C.D.(2012, June) ***The Roles of Badges in the Computer Science Student Network.*** 2012 GLS Conference
- Abramovich, S., Higashi, R., Hunkele, T. Schunn, C.D., Shoop, R. (2011, July) ***Achievement Systems to Boost Achievement Motivation.*** 2011 GLS Conference

Can CTP Be Taught in Robotics Classrooms?



Can Computational Thinking Practices Be Taught in Robotics Classrooms?

Presented at the International Technology and Engineering Education Conference
National Harbor
Washington DC
March 2-4, 2016

Authors
Robin Shoop, Jesse Flot, Tim Friez Carnegie Mellon University, Robotics Academy
Christian Schunn, Eben Witherspoon University of Pittsburgh, Learning Research and Development Center

This material is based on work supported by the National Science Foundation under DRK-12 research, Award Number 1418199, Changing Culture in Robotics Classrooms. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation or any collaborator or partner named herein.

- 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 all 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:
Eben B. Witherspoon, Ross M. Higashi, Christian D. Schunn, Emily C. Baehr, and Robin Shoop. 2017. Developing Computational Thinking Through a Virtual Robotics Programming Curriculum. *ACM Trans. Comput. Educ.* 18, 1, Article 4 (October 2017), 20 pages.
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 (Grover 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

This work is supported by the National Science Foundation, under grant DRL1418199.
Authors' addresses: E. B. Witherspoon, R. M. Higashi, C. D. Schunn, and E. C. Baehr, Learning Research and Development Center, University of Pittsburgh, 3939 O'Hara Street, Pittsburgh, PA 15260; emails: {ebw13, rmh57, schunn, ecb42}@pitt.edu; R. Shoop, National Robotics Engineering Center, 10 40th Street, Pittsburgh, PA 15201; email: rshoop@andrew.cmu.edu.

https://doi.org/10.1145/3104982

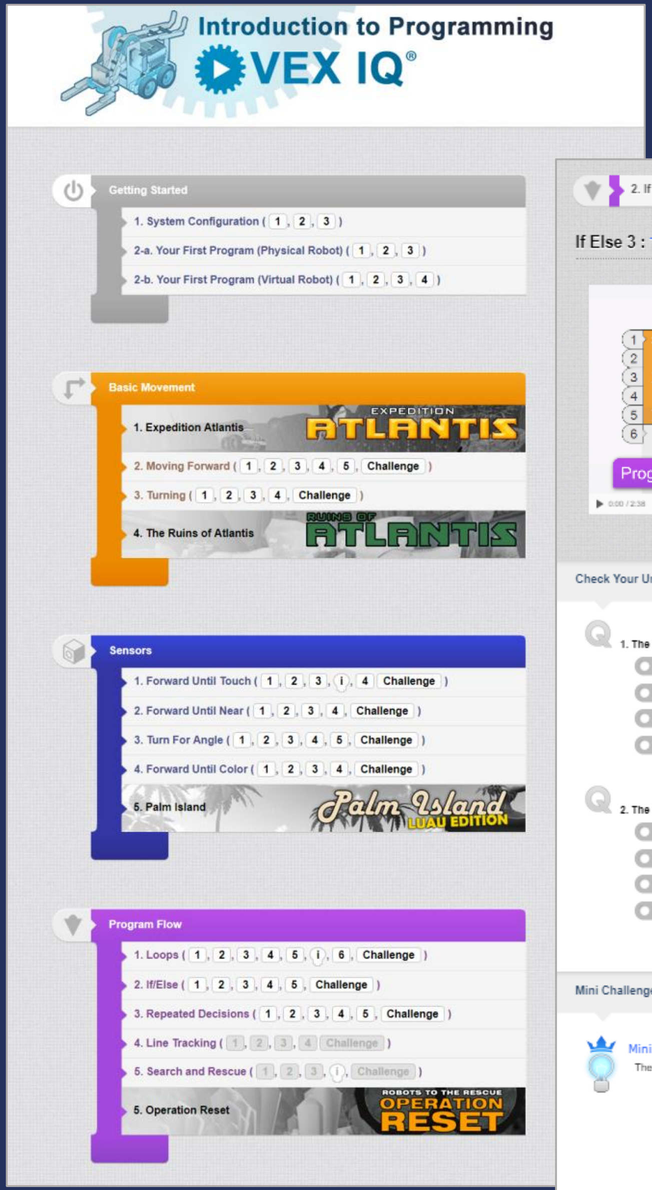
Organization of the Study

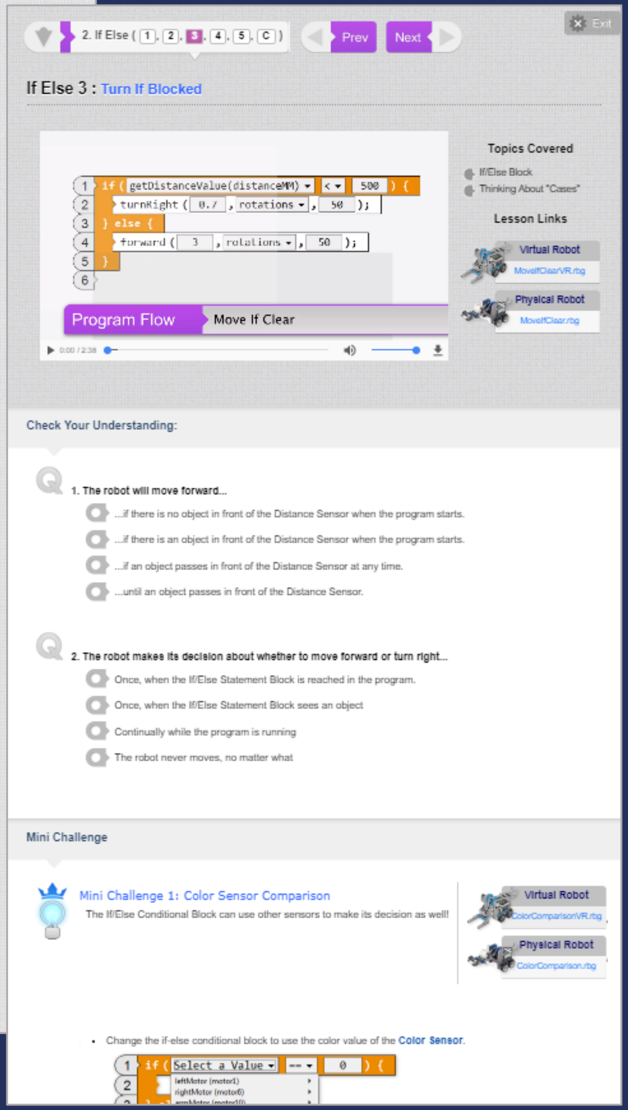
- Two Studies
  - Single school district for feedback and debugging
  - 4 districts and 26 classrooms
- Tracked student progress with three versions of Computational Thinking assessment
- Pre and Post tests

Research Study

Key Technologies

Curriculum



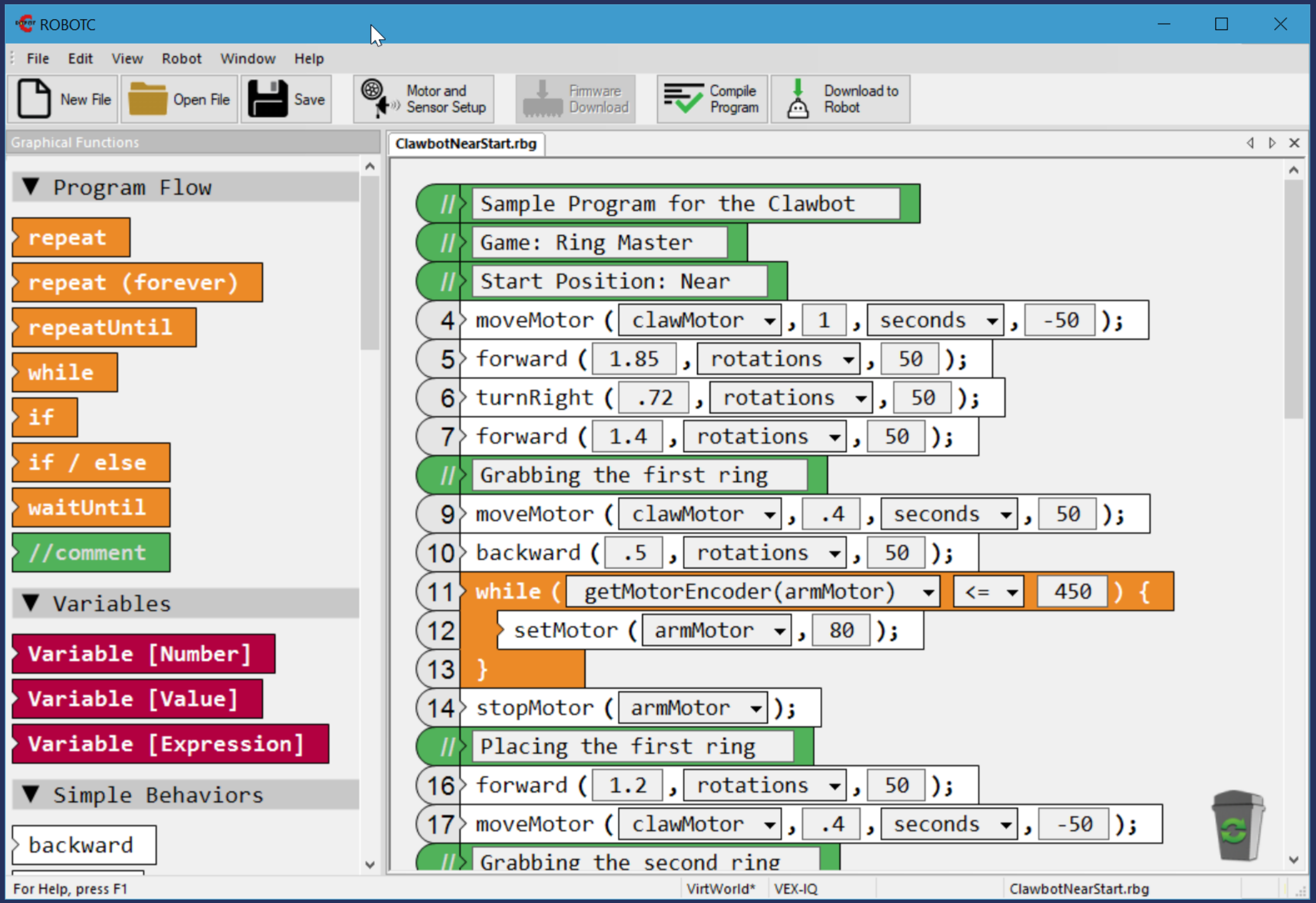


- Structured in 4 units
  - Getting Started, Basic Movement, Sensors, Program Flow
- Provides multi-media driven direct instruction to students
- Scaffolds Programming Concepts and Big Ideas in Computational Thinking in robotics-driven activities
- Includes embedded authentic Formative and Summative Assessment opportunities

CS-STEM Network (CS2N)

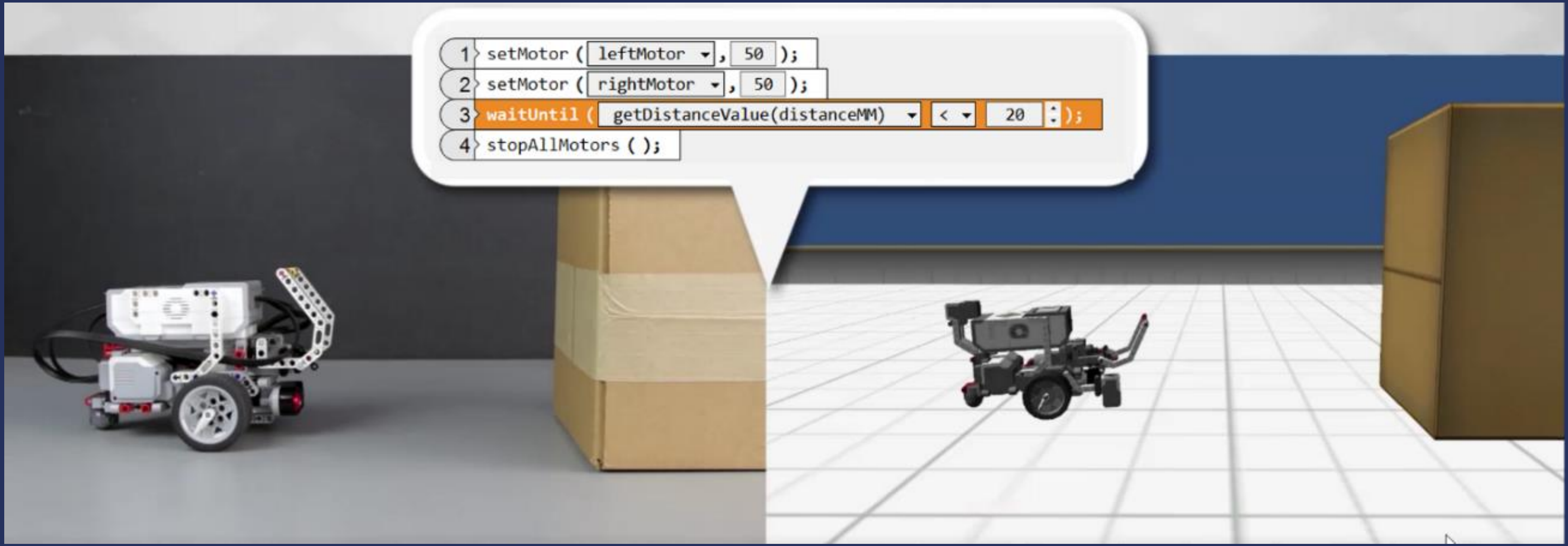
- An education platform where students can earn badges and certificates for CS-STEM related skills.
- Create student progress reports for teachers and researchers.
- Progress measured includes:
  - Curricular Material Consumption
  - Computational Artifact Uploads
  - Endorsements
  - Quiz Assessments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

ROBOTC Programming Environment



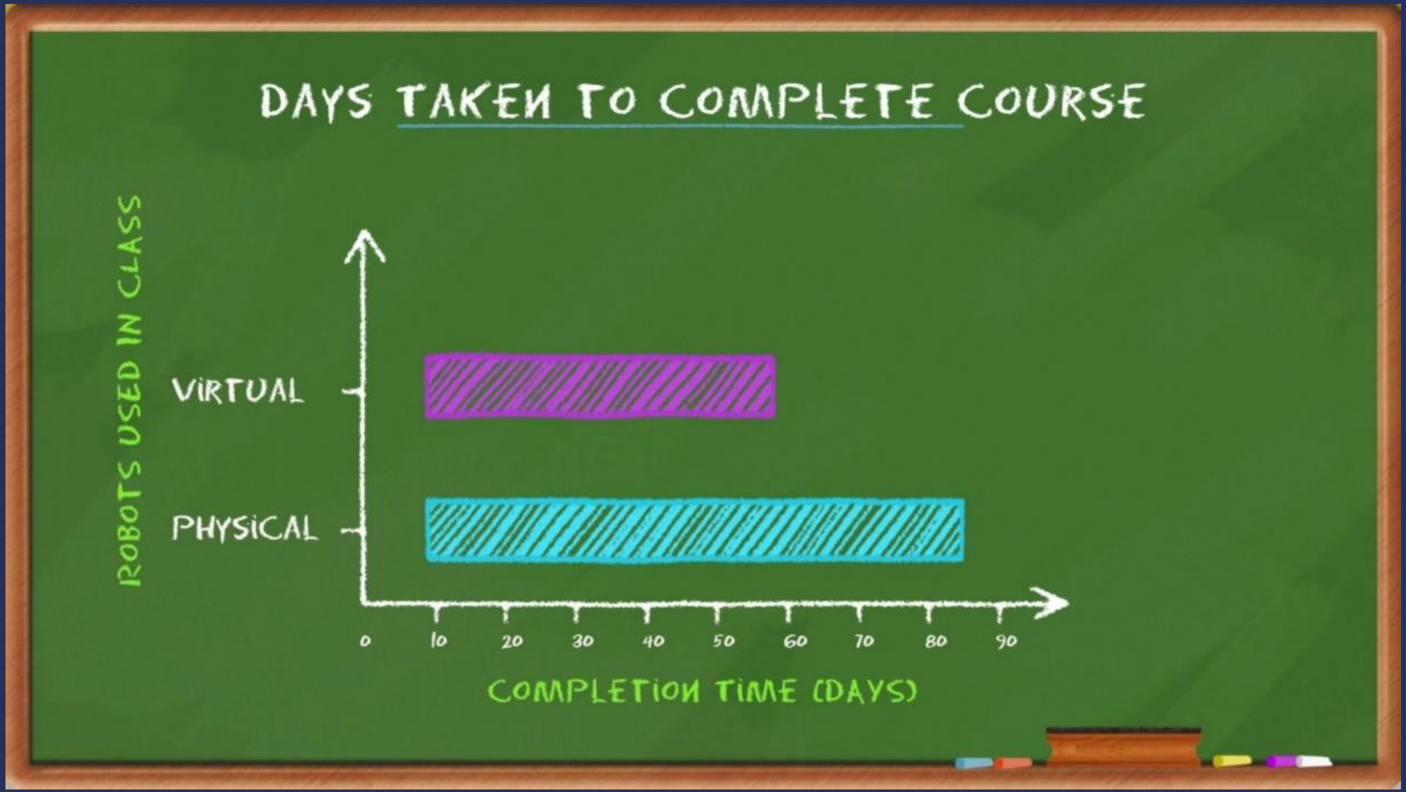
- C-Based programming language with robotics extensions
- Graphical and text-based modes
- Includes natural language and standard language constructs
- Supports multiple robot platforms popular in education, including robot simulations (RVW)

Robot Virtual Worlds



- 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

Robot Virtual Worlds: Results



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

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:
Eben B. Witherspoon, Ross M. Higashi, Christian D. Schunn, Emily C. Baehr, and Robin Shoop. 2017. Developing Computational Thinking Through a Virtual Robotics Programming Curriculum. *ACM Trans. Comput. Educ.* 18, 1, Article 4 (October 2017), 20 pages.
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 (Grover 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

This work is supported by the National Science Foundation, under grant DRL1418199.
Authors' addresses: E. B. Witherspoon, R. M. Higashi, C. D. Schunn, and E. C. Baehr, Learning Research and Development Center, University of Pittsburgh, 3939 O'Hara Street, Pittsburgh, PA 15260; emails: {ebw13, rmh57, schunn, ecb42}@pitt.edu; R. Shoop, National Robotics Engineering Center, 10 40th Street, Pittsburgh, PA 15201; email: rshoop@andrew.cmu.edu.

https://doi.org/10.1145/3104982

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?

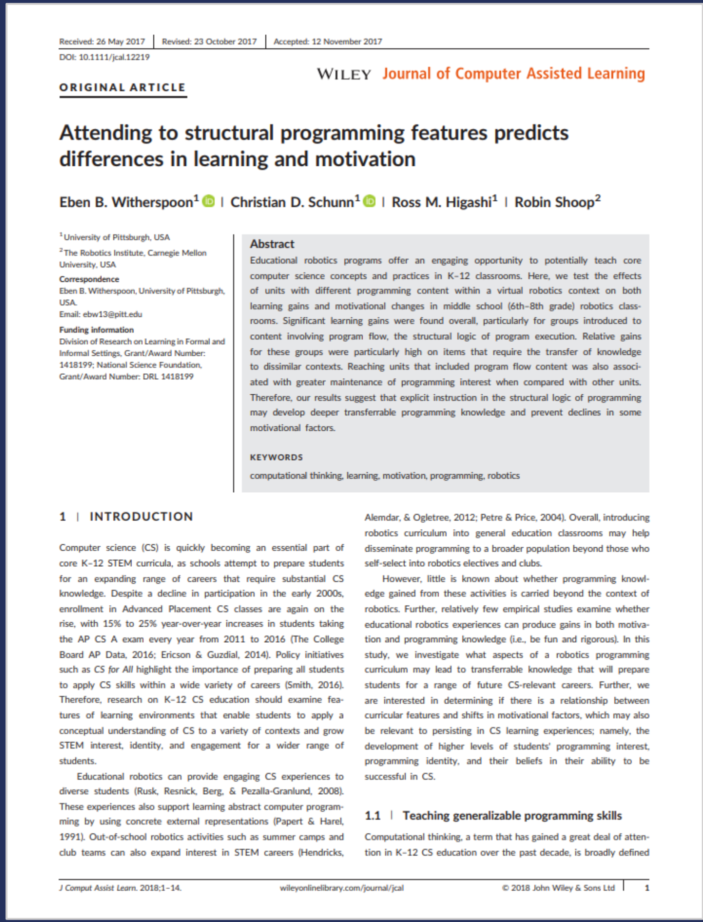
Practical Considerations

- 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



Received: 26 May 2017 | Revised: 23 October 2017 | Accepted: 12 November 2017
DOI: 10.1111/jcal.12219

WILEY Journal of Computer Assisted Learning

ORIGINAL ARTICLE

Attending to structural programming features predicts differences in learning and motivation

Eben B. Witherspoon[1] | Christian D. Schunn[1] | Ross M. Higashi[1] | Robin Shoop[2]

[1] University of Pittsburgh, USA
[2] The Robotics Institute, Carnegie Mellon University, USA

Correspondence
Eben B. Witherspoon, University of Pittsburgh, USA.
Email: ebw13@pitt.edu

Funding information
Division of Research on Learning in Formal and Informal Settings, Grant/Award Number: 1418199; National Science Foundation, Grant/Award Number: DRL 1418199

Abstract
Educational robotics programs offer an engaging opportunity to potentially teach core computer science concepts and practices in K–12 classrooms. Here, we test the effects of units with different programming content within a virtual robotics context on both learning gains and motivational changes in middle school (6th–8th grade) robotics classrooms. Significant learning gains were found overall, particularly for groups introduced to content involving program flow, the structural logic of program execution. Relative gains for these groups were particularly high on items that require the transfer of knowledge to dissimilar contexts. Reaching units that included program flow content was also associated with greater maintenance of programming interest when compared with other units. Therefore, our results suggest that explicit instruction in the structural logic of programming may develop deeper transferrable programming knowledge and prevent declines in some motivational factors.

KEYWORDS
computational thinking, learning, motivation, programming, robotics

1 | INTRODUCTION

Computer science (CS) is quickly becoming an essential part of core K–12 STEM curricula, as schools attempt to prepare students for an expanding range of careers that require substantial CS knowledge. Despite a decline in participation in the early 2000s, enrollment in Advanced Placement CS classes are again on the rise, with 15% to 25% year-over-year increases in students taking the AP CS A exam every year from 2011 to 2016 (The College Board AP Data, 2016; Ericson & Guzdial, 2014). Policy initiatives such as *CS for All* highlight the importance of preparing all students to apply CS skills within a wide variety of careers (Smith, 2016). Therefore, research on K–12 CS education should examine features of learning environments that enable students to apply a conceptual understanding of CS to a variety of contexts and grow STEM interest, identity, and engagement for a wider range of students.

Educational robotics can provide engaging CS experiences to diverse students (Rusk, Resnick, Berg, & Pezalla-Granlund, 2008). These experiences also support learning abstract computer programming by using concrete external representations (Papert & Harel, 1991). Out-of-school robotics activities such as summer camps and club teams can also expand interest in STEM careers (Hendricks, Alemdar, & Ogletree, 2012; Petre & Price, 2004). Overall, introducing robotics curriculum into general education classrooms may help disseminate programming to a broader population beyond those who self-select into robotics electives and clubs.

However, little is known about whether programming knowledge gained from these activities is carried beyond the context of robotics. Further, relatively few empirical studies examine whether educational robotics experiences can produce gains in both motivation and programming knowledge (i.e., be fun and rigorous). In this study, we investigate what aspects of a robotics programming curriculum may lead to transferrable knowledge that will prepare students for a range of future CS-relevant careers. Further, we are interested in determining if there is a relationship between curricular features and shifts in motivational factors, which may also be relevant to persisting in CS learning experiences; namely, the development of higher levels of students' programming interest, programming identity, and their beliefs in their ability to be successful in CS.

1.1 | Teaching generalizable programming skills

Computational thinking, a term that has gained a great deal of attention in K–12 CS education over the past decade, is broadly defined

- 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

Education Tech Research Dev
DOI 10.1007/s11423-017-9528-z

CrossMark

RESEARCH ARTICLE

Different underlying motivations and abilities predict student versus teacher persistence in an online course

Ross M. Higashi[1] · Christian D. Schunn[1] · Jesse B. Flot[2]

Abstract Free online courses, including Massively Open Online Courses, have great potential to increase the inclusiveness of education, but suffer from very high course dropout rates. A study of 172 K-12 students and 114 K-12 teachers taking the same free, online, summertime programming course finds that student and teacher populations have different underlying motivational models that predict rates of persistence in the course despite having generally similar motivational levels. Student persistence is predicted by prior programming knowledge, intrinsic interest in the subject matter, and mastery approach goals. By contrast, 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. This sub-population discrepancy in predictive factors is novel, and may be reflective of differing environmental conditions or internal mechanisms between students and teachers. Future design of free choice learning environments can take these factors into account to increase rates of user persistence for different target user populations.

Keywords MOOC · Programming · Students · Teachers · Motivation · Persistence

Free online courses, including Massively Open Online Courses (MOOCs), are an increasingly popular form of online instruction offering for large audiences. They typically offer access to live, video-recorded, or online text instruction with no associated fee, and often cover high-demand topics like introductory sciences or computer programming. Since the cost to offer a course to diverse audience scales only with the costs

✉ Ross M. Higashi
rmh57@pitt.edu

[1] Learning Research and Development Center, University of Pittsburgh, 3939 O'Hara Street, Pittsburgh, PA 15260, USA

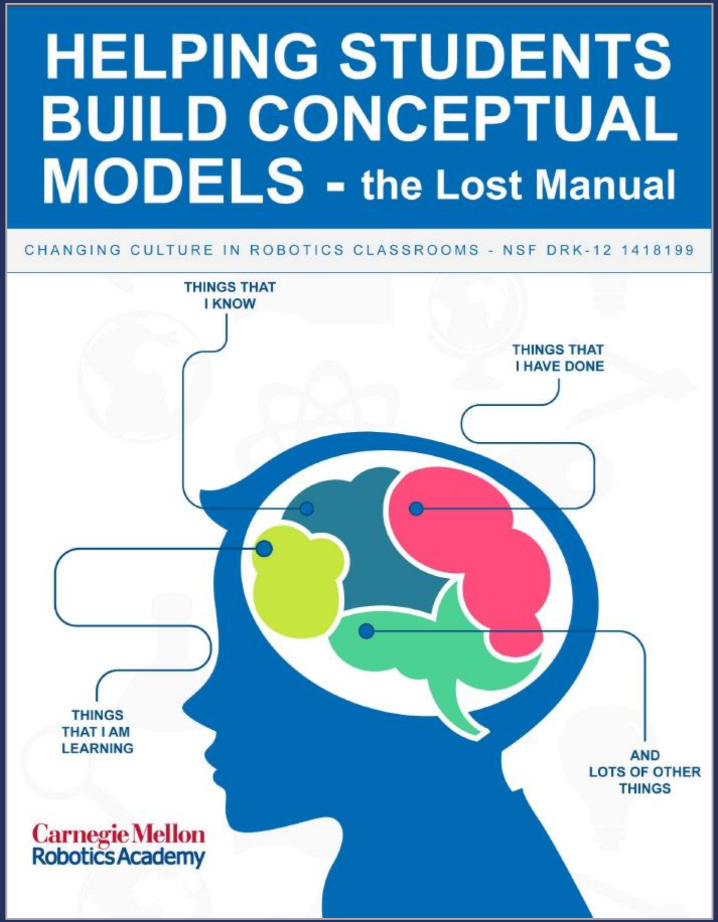
[2] Robotics Academy, Carnegie Mellon University, Ten 40th Street, Pittsburgh, PA 15201, USA

Published online: 02 May 2017

Springer

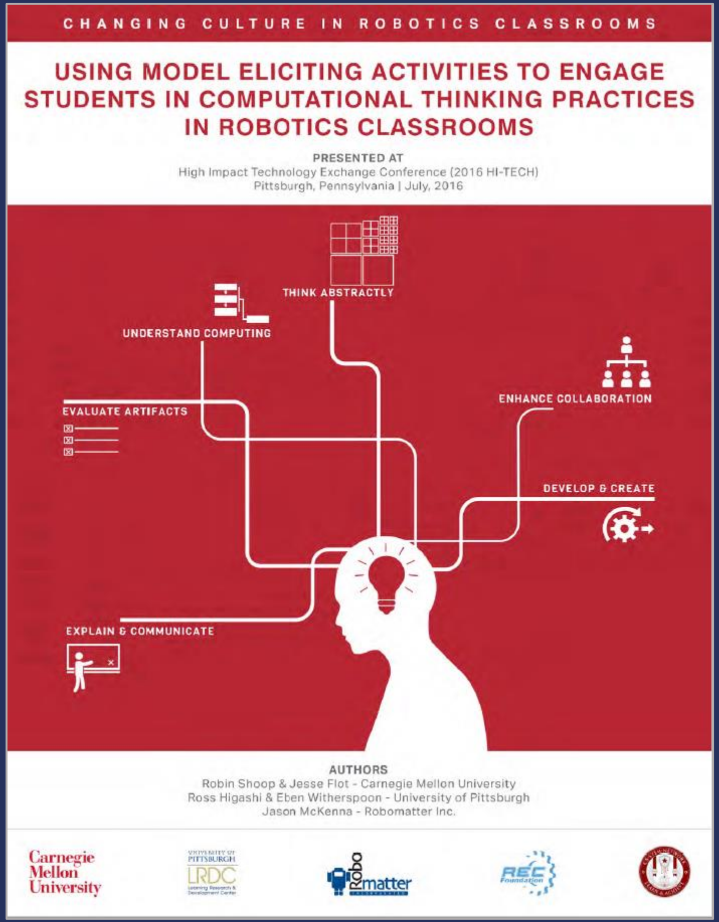
- Summary:
  - Student persistence is predicted by prior programming knowledge, intrinsic interest in the subject matter, and mastery approach goals.
  - Teacher persistence is similarly predicted by intrinsic interest, but then also by self-identity as a programmer, performance approach goals, and negatively 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
- Flot, J., McKenna, J., Shoop, R. (2016)
  Helping Students Build Conceptual Models – the Lost Manual
  Carnegie Mellon Robotics Academy, Pittsburgh, PA.

Using MEAs to Engage Students in CTP



- 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
  Presented at the High Impact Technology Exchange Conference (2016 HI TEC), Pittsburgh, Pennsylvania.

Using MEAs to Engage Students in CTP

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

- 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.