designing and implementing a STEM summer camp

by Allyson Hallman-Thrasher,
Chang Lui, Courtney Koestler,
Susanne Strachota, and Jen Thompson

The siloed nature of schooling where each discipline exists in its own class and teachers in upper grades are often certified for only one subject area and in lower grades are not expected to be content experts, does not foster meaningful integration of STEM disciplines.

Introduction

Schools across the U.S. report widespread teaching vacancies and difficult-to-staff positions in mathematics and science (Malkus et al., 2015). This challenge has impacted already marginalized and under-resourced schools, particularly rural schools. Monk (2007) reports that many rural science teachers lack strong coursework in the sciences when compared to their urban counterparts. In response, we have developed a program designed to prepare individuals with STEM degrees, hereinafter referred to as fellows, to become mathematics and science teachers especially well-positioned to teach in rural districts. Because our fellows have strong content knowledge from their STEM degrees, our program focuses on providing high-quality teacher preparation that attends to multiple STEM disciplines, develops pedagogical content knowledge, and provides cohesive, coherent support for learning about teaching across the middle/high school and the university classroom. Here we report on the lessons we learned from bringing a group of STEM experts together to develop and implement a summer camp for rural middle school students. Before we share about this experience, we first describe what we mean by STEM because the meaning of STEM and the purpose of STEM education continue to be unclear (Bybee, 2010).

Used as an acronym for the four disciplines, science, technology, engineering, and mathematics, "STEM education" or "STEM literacy" are phrases often used by policymakers when discussing shortcomings of U.S. education and areas to which attention, and thus funding, should be diverted. Before engaging in conversations about developing students' STEM literacy, ambiguity about the meaning of the term STEM should be addressed. Two ways that the term STEM is used are particularly problematic: 1) STEM is used as a synonym for mathematics and science and 2) STEM is used to refer to the integration of the four disciplines. The first definition downplays the importance of the technology and engineering components of STEM, while the second assumes that all ways of integrating the disciplines are equal. Technology and engineering are of special concern because, as noted, they are often eclipsed by an overemphasis on math and science. Moreover, often they are not a part of the regular curriculum and viewed as extra content for students to learn and teachers to plan for.

Efforts to improve STEM education have called for discipline integration as a way to promote student learning, show practical applications of math and science concepts, and elevate the position of the underrepresented T and E in STEM. However, the siloed nature of schooling where each discipline exists in its own class and teachers in upper grades are often certified for only one subject area and in lower grades are not expected to be content experts, does not foster meaningful integration of STEM disciplines. In a survey of the state of computer science education, "Sixty-three percent of K-12 principals and 74% of superintendents who do not have CS in their school or district say a reason they do not offer CS is the lack of teachers available at their school with the necessary skills to teach it" (Google Inc. & Gallup Inc, 2016).

Our work aims to address some of these issues by preparing individuals with STEM degrees to become math and science teachers. We approach STEM education through a focus on STEM Practices, the ways in which scientists, mathematicians, and engineers engage in doing and thinking about their work, and the methods, values, and ideas of STEM fields (Lowrie et al., 2018). This approach is supported by a focus on the Technology and Engineering Practices (International Technology and Engineering Educators Association [ITEEA], 2020), the mathematics practices of the Common Core State Standards (National Governors Association Center for Best Practices [NGA] & Council of Chief State School Officers [CCSSO], 2010) and the Next Generation Science and Engineering Practices (Next Generation Science Standards Lead States [NGSS], 2013). (See Figure 1.) We draw on practices that cut across all disciplines, particularly modeling and constructing arguments, which while common to all four STEM disciplines, have a slightly different implementation in each. We also frame problem solving and engineering design as analogous cyclical processes.

We use these practices to guide our work with teachers and students, unify our program, and highlight differences across disciplines. We share this conceptualization of STEM because we believe this approach is pragmatic in that it clearly highlights *how* students can engage with STEM and how their engagement with one of the four disciplines translates to another discipline.

Figure 1.
Technology and Engineering, Mathematics, Science & Engineering, and STEM Practices

Technology and Engineering Practices (International Technology and Engineering Educators Association, 2020)

1. Systems Thinking
2. Creativity
3. Making and Doing
4. Critical Thinking
5. Optimism
6. Collaboration
7. Communication
8. Attention to Ethics

Mathematics Practices (CCSSO, 2010)

1. Make sense of problems and persevere in solving them.
2. Reason abstractly and quantitatively.
3. Construct viable arguments and critique the reasoning of others.
3. Model with mathematics.
5. Use appropriate tools strategically.
6. Attend to precision.
7. Look for and make use of structure.
8. Look for and express regularity in repeated reasoning.

Science and Engineering Practices (NGSS, 2013)

1. Asking questions (for science) and defining problems (for engineering).
2. Developing and using models.
3. Planning and carrying out investigations.
4. Analyzing and interpreting data.
5. Using mathematics and computational thinking.
6. Constructing explanations (for science) and designing solutions (for engineering).
7. Engaging in argument from evidence.
8. Obtaining, evaluating, and communicating information.

STEM Practices (Lowrie et al., 2018)

Ideas	Methods	Values
▪ Problem finding	▪ Generating ideas	▪ Curiosity
▪ Finding and validating evidence	▪ Processing information	▪ Integrity
▪ Questioning	▪ Encoding and decoding information	▪ Imagination
▪ Proposing	▪ Using appropriate language and vocabulary	▪ Creativity
▪ Designing and building	▪ Using tools to produce artefacts	▪ Teamwork
▪ Exploring and challenging	▪ Thinking critically	▪ Persistence

Our Program

In response to the critical shortage of highly qualified mathematics and science teachers in rural areas, our institution has partnered with local school districts and community nonprofit organizations to offer a teacher preparation program for individuals who have already earned a degree in a STEM field and want to become certified teachers. Throughout its activities, our program adopts a STEM focus as opposed to a mathematics or science focus to deepen teachers' learning of STEM content and practices and develop their identities as STEM teachers. We address the crucial need of a well-prepared, highly qualified STEM teaching force in the rural regions of the United States because rural students are least likely to experience science and technology in their schools and daily lives (State of Computer Science Education, 2018). By increasing the cadre of effective and highly qualified mathematics and science teachers in our rural underserved area, we hope to raise the profile of STEM disciplines in rural communities, increase access to effective mathematics and science teaching for middle and high school students, and ultimately improve preparation in STEM fields for these students and an increased capacity for a STEM workforce in rural areas.

As part of the partnership program, we co-hosted a one-week STEM Summer camp for students 11 to 14 years old in a high-need school in rural Appalachia. The camp ran for one week in July 2019, as part of students' regular summer school programming. Our fellows, preservice teachers who were just beginning their masters'-level teacher preparation program, and faculty served as co-instructors. The school in which we hosted our STEM camp is one of the university's high-need local education agency partners. The school also served as a site for some of our fellows' year-long clinical placements, which began the subsequent fall. We use the term high need as it is defined in section 201 of the Higher Education Act of 1965 (20 U.S.C. 1021). The camp was a part of our fellows' introductory field experience for our program, which involved designing, implementing, and reflecting on the week-long STEM camp.

We designed the introductory field experience with three tenets in mind. First, interventions should "model how to teach in ways that recognize and adjust to the cultural differences of school children" and preservice teachers need to explore and reflect on how knowledge is culturally bound (Terrill & Mark, 2000, p. 154). One way we designed our introductory field experience to address the first tenet is by meeting for several hours each day after the STEM camp. This supported fellows in recognizing and reflecting on their practice, attending specifically to cultural strengths of the students and community. It also gave them the opportunity to discuss and modify their plans based on what they were learning about this specific group of students. We also chose topics for investigation that related to students' experiences in their communities. In our rural context a focus on agriculture and a history of mining meant that our students were connected to the land and geography of our region, so several of our activities focused on the geology of our area.

Second, teacher education programs should increase preservice teachers' experiences with diverse youth by providing diverse field experiences (Baldwin et al., 2007; Whitaker & Valtierra, 2018), One way this introductory field experience addressed the second tenet was by differentiating it from all other field experiences provided in our program. Most field experiences are traditional in that they occur in a classroom setting. However, we have incorporated community events and this camp in order to give our fellows diverse experiences. Fellows facilitated informal learning opportunities with students and their families at events held at a local community center. In this camp fellows were not restricted by the sometimes rigid classroom schedule that necessitates covering particular standards or particular sections of a textbook. They were also afforded the flexibility to experiment with different modes of instruction. We also purposefully chose to involve students who have struggled with the traditional structures of school, so fellows had the opportunity to think about how to motivate learners to engage meaningfully in content. The third tenet guiding our design was the use of cross-cutting STEM practices: constructing arguments and identifying and solving problems, and modeling to unify our diverse topics. Each activity, regardless of which discipline took a central focus, drew on one or more of these practices. Throughout the camp we also provided multiple opportunities to use discipline-specific technology (e.g., mathematical software for modeling geometric and algebraic objects, spreadsheets for data analysis, and various measurement tools to capture geological and environmental data).

STEM Camp

During the first four days of camp, our fellows and project faculty designed, implemented, and reflected on STEM lessons, which addressed topics such as weathering, erosion, forces, and motion, classifying rocks, density, and slope. On the final day we focused primarily on programming robots. One goal in planning the activities was to engage students in STEM practices and encourage them to explore STEM careers (e.g., geologists, engineers). Students spent four of the five days investigating common phenomenon from a mathematical perspective, justifying their reasoning by computing and testing formulas, and a science perspective, justifying their conjectures using data from their experiments. Students rotated through these two sessions each day. In the science session, students had to justify the conjectures using data from their experiments, whereas in the math session they justified their reasoning by computing and testing formulas. For example, in a science session students explored landslides, a common phenomenon for our region's geology, through an experiment that led students to make a hypothesis about which conditions caused landslides (e.g., type of soil, how and how much water is added) and what angle of elevation is needed for different types of soil to fall down a slope. When students explained and justified their process, they showed that they are able to defend findings related to a design problem, which is a benchmark from the new ITEEA 2020 Technological and Engineering Literacy *(STEL)* standards and provided them an opportunity to engage in argumentation. This benchmark is listed as benchmark S under Standard 2. In that day's corresponding mathematics session, students used technology to explore different slopes, develop a formula for measuring slope, and compare angles of elevation to slopes calculated using their formulas. Hereinafter, when we include STEL followed by a number and letter, we are indicating the standard number and the letter of the benchmark. For instance, the aforementioned benchmark was STEL 2S.

Programming Activities

As noted, the final day of the camp focused on programming activities that built on technology used earlier in the week. Coding requires students to not only develop a solution to a problem but communicate that solution in a specific way such that it can be carried out by a computer. To engage students in computational thinking, we used cognitively demanding problems (Smith & Stein, 2011), those in which a solution or a solution strategy is not immediately obvious, and strategies are developed and implemented by the students. Small robots provided students with immediate feedback and a richly engaging experience, since they could immediately test their code and see real results. We used a "use-modify-create" progression (Lee et al., 2011) through the coding activities, which has been shown to foster students' computational thinking. We also observed that through students' engagement with this progression, they demonstrate that they met several of the 2020 *STEL* standards' benchmarks. For example, in the progression, students first interact with a computer to learn how it is coded. Next, they develop computational thinking by modifying and revising instructions that have been written for a computer. Finally, students develop a project of their own in the create stage. The progression approach supported students' learning but also gave teacher candidates a structure for scaffolding student problem solving. Through this process students demonstrated meeting the ITEEA (2020) benchmarks for *STEL*, "apply the technology and engineering design process" (7Q) and "refine design solutions to address criteria and constraints" (7R).

In the programming activity, students were given Ozobots, small robots, or bots, that can be programmed two ways, either by color coding a surface that the robot can scan and read or by using a block-style coding software (called OzoBlockly). The code is entered in a web-based platform, then downloaded to the Ozobot using a system of colored LED flashes called flash loading. The bots can be coded to travel a set path, change direction or speed depending on conditions, change the light colors that flash on the bot, and play sounds. At more advanced levels of coding, variables and functions can be defined and complex logical statements and loops can be used. To begin, we encouraged students to tinker with drawing tracks and using different colors to program paths for the bots (see Figure 2). The bot is able to drive on a "track" drawn by the students and, depending on addition of sequences of dots of different colors, the bots will do different actions (e.g., change speed, change path, etc.). In this use stage, as students explored with bots and began to notice the effects of different dot sequences, we began to form a code glossary on the whiteboard (see Figure 3). In developing a code glossary, students were learning to "predict outcomes of a future product or system" prior to beginning their own designs (ITEEA, 2020). As students tinkered with the Ozobots, our fellows moved between groups of students and supported them in exploring the different abilities of these robots and proposing "what if" questions such as "What will happen if you change the order of these color codes?" and "What if the bot reads the dots backwards?" As students were prompted to address these questions, they "differentiated between inputs, processes, outputs and feedback in technological systems" (ITEEA, 2020).

Once students learned what the dot sequences did through their exploration, the activity moved to the modify phase, where fellows shared additional codes the students may have missed and also created

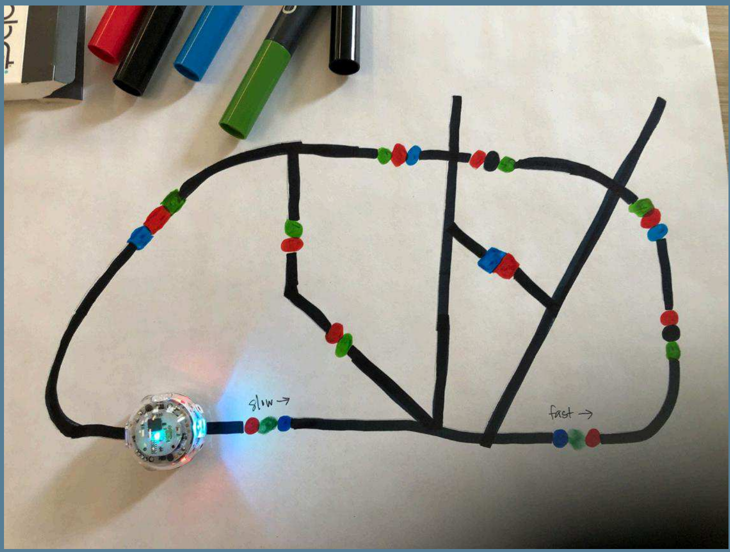


Figure 2.
Example track to explore effects of colored dot codes on bots.

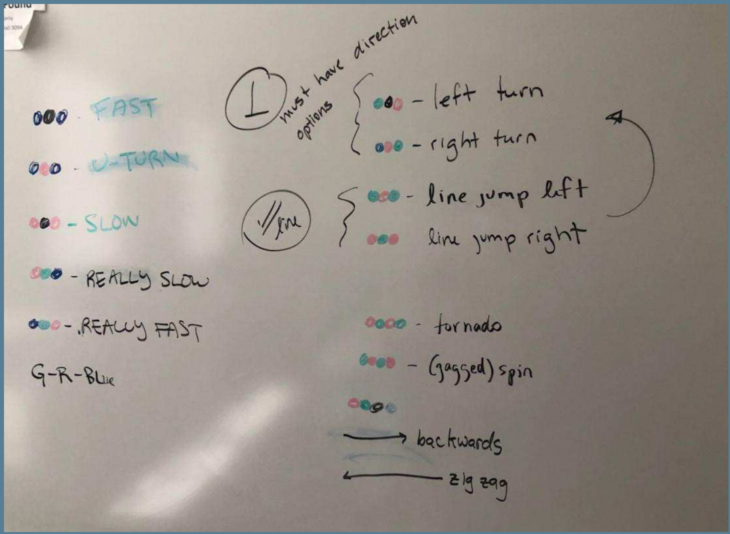


Figure 3.
Glossary of codes developed by students after exploring bots on different racks.

simple tracks for students to revise (e.g., make the bot go left here instead of right, make the bot drive this track backwards, or change speeds). Finally, students were given the opportunity to create their own tracks to navigate obstacles, make decisions on which tracks to follow, and use a preset time or number of moves (see Figure 4). In accounting for these constraints students meet the *STEL* standard benchmark: "create a closed-loop system that has a feedback path and requires no human interaction" (2P). Moreover, in this create stage, fellows were able to assess student understanding and observe STEM practices in action. For example, the students who created the track shown in Figure 4 wanted to create a ramp for their bot to traverse. They had to experiment to determine what level of incline was shallow enough for the bots to go up, what moves it could execute at the top of the ramp before going down the other side, what speed codes were needed to get up and down the ramp, and how tight a turn it could make and how to use line jump codes to randomize the bot's path. In doing so, they questioned, proposed, designed, and built and then explored and challenged their solutions.

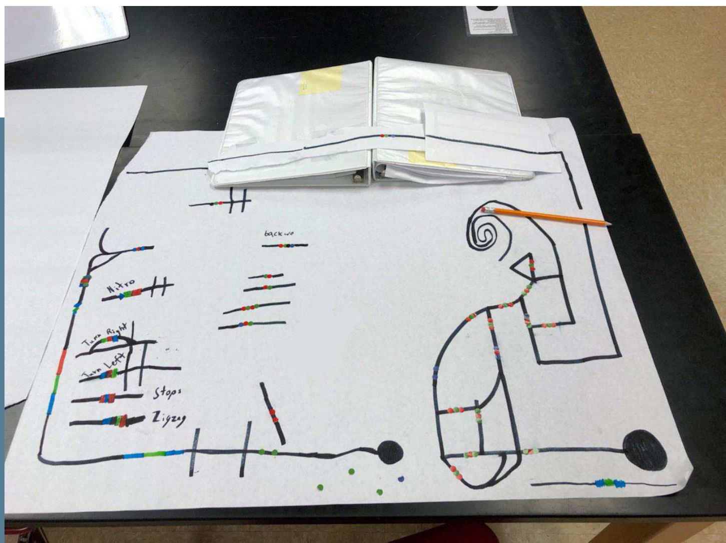


Figure 4.
Student-created track with obstacles.

As individual students developed different levels of expertise around color code programming the robots, we encouraged them to explore more sophisticated programming with the OzoBlockly software, which was another opportunity to create a closed-loop system. We started this by turning on a program that we had pre-loaded on the bots. The bots would execute the program given the correct inputs and no longer needed a track to follow to know what to do. In the use phase, students were given a sheet of paper with large colored dots and told to get the bot on one of the dots (if they did the bot would say the color of the dot and keep a running tally of the dots it touched). Typically, as they reached out to touch the bots, the bots moved. Through exploration, students then learned the bots had motion sensors and depending on where they placed their hands near the bots, the bots moved in particular (not always intuitive) directions. For example, students learned they could make the bot move forward by keeping their hand behind the bot a specific distance. However, placing their hand to the right made the bot turn left. We then facilitated a whole class discussion about what instructions the program must be giving to the bots. Next, we shared the actual block code with the class and asked them to map their descriptions to the line of code. In doing so we supported their technological literacy to translate between the programming language and everyday descriptions. After brief instructions on how to drag and drop different codes, in the modify phase we gave students time to tinker with the software and see how changing the code changed the robot's actions and in doing so engaged students in "predicting outcomes of a future product or system at the beginning of the design process" (ITEEA, 2020). We noticed that students struggled to create a code for the bot that did anything more than what they had already observed, so to keep students engaged the team made the decision to stay at the modify stage for this portion of the activity, as students were able to modify and share many complex revised programs.

Lessons Learned

As novices, most of our project team experienced learning about bots and programming in the same way that the summer school students would—as new content. This experience gave team members insight as to how students would think about the activities, the kinds of questions they would ask, and the issues that might arise. We felt this was a unique scenario because it allowed for authentic problem solving, whereas when project team members and preservice teachers are teaching content they have mastered, they often have to pretend to not understand in order to encourage students in engaging in authentic problem solving. That was rarely the case with this activity.

Much like classroom teachers, our schedule was determined by our school partner and one issue we encountered with all lab components and the programming activities, was that students ran out of time. Our students needed ample time to explore, tinker, and eventually delve into the content to begin creating or producing on their own. In our "use-modify-create" progression (Lee et al., 2011), typically, students had time for the exploratory phase in which they were using and modifying, but too often they ran out of time as soon as they seemed to start creating and producing on their own. One way we would mitigate this tension in the future is to spread the activities in smaller chunks across multiple days.

One beneficial aspect of our schedule was that it allowed us time to debrief and revise our plans for the next day based on the outcomes of the preceding day. In short, the structure of the program allowed our team to implement plan-teach-debrief cycles. Our Fellows had the opportunity to analyze their teaching, discuss how they might revise their instruction moving forward, and then put those ideas into practice the following day. We suggest moving forward to allow students more time to work, with a gap between the end of camp and the debrief session, so students who were interested in dedicating additional time to these tasks could.

We understand that lessons vary across contexts but hope that by sharing observations about our experience we support other teachers and teacher educators who are planning similar activities to better prepare and implement.

Conclusion

Our project was intended to serve as an opportunity for students to develop an interest in STEM topics using our local environment and technology and prospective teachers. We did so by introducing them to an array of topics and content areas and engaging them in STEM Practices, mathematical problem solving. Through their interaction with these topics and content areas we observed students continuously engaging in the Common Core Mathematical Practices of problems, the *NGSS* science practices, the STEM practices outlined by Lowrie et al., (2018) and demonstrating the ability to meet *STEL* standard benchmarks. We feel the activities we have described address the goals of today's teachers while introducing students to new and relevant content.

Acknowledgement

The research reported here was supported in part by National Science Foundation Award #1758484. 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.

References

Baldwin, S. C., Buchanan, A. M., & Rudisill, M. E. (2007). What teacher candidates learned about diversity, social justice, and themselves from service-learning experiences. *Journal of Teacher Education, 58*(4), 315-327.

Bybee, R. W. (2010). What is STEM education. *Science, 329,* 996. doi: 10.1126/science.1194998

Google Inc. & Gallup Inc. (2016). *Trends in the State of Computer Science in U.S. K-12 Schools.* Retrieved from http://goo.gl/j291E0

International Technology and Engineering Educators Association. (2020). *Standards for technological and engineering literacy: The role of technology and engineering in STEM education.* www.iteea.org/STEL.aspx

K–12 Computer Science Framework. (2016). Retrieved from http://www.k12cs.org

Lee I., Martin, F., Denner, J., Coulter, B., Allan, W., Erickson, J., Malyn-Smith, J., & Werner, L. (2011). Computational thinking for youth in practice. ACM Inroads, 2(1), 32–37.

Lowrie, T., Leonard, S. & Fitzgerald, R. (2018). STEM Practices: A translational framework for large-scale STEM education design. *Educational Design Research, 2*(1), 1-20. http://dx.doi.org/10.15460/eder.2.1.1243

Malkus, N., Hoyer, K. M., & Sparks, D. (2015). *Teaching Vacancies and Difficult-to-Staff Teaching Positions in Public Schools.* Stats in Brief. NCES 2015-065. National Center for Education Statistics.

Monk, D. H. (2007). Recruiting and retaining high-quality teachers in rural areas. *Excellence in the Classroom, 17*(1), 155–174. http://doi.org/10.1353/foc.2007.0009

National Governors Association Center for Best Practices & Council of Chief State School Officers. (2010). *Common core state standards for mathematics.* Washington, DC: Council of Chief State School Officers. Retrieved from http://www.corestandards.org/assets/CCSSI_Math%20Standards.pdf

Next Generation Science Standards Lead States. (2013). *Next generation science standards: For states, by states.* Washington, DC: The National Academies Press

Smith, M. S., & Stein, M. K. (2011). *Five practices for orchestrating productive mathematics discussions.* Reston, VA: National Council of Teachers of Mathematics.

State of Computer Science Education. (2018). Retrieved from https://advocacy.code.org/

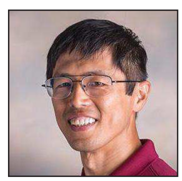
2018 State of Computer Science Education. (2018). Retrieved from https://advocacy.code.org/

Terrill, M. M., & Mark, D. L. (2000). Preservice teachers' expectations for schools with children of color and second-language learners. *Journal of Teacher Education, 51*(2), 149-155.

White, Adrienne. (2019, April). Lessons. 19 *Ozobot Lessons for the Hour of Code.* https://ozobot.com/blog/19-ozobot-lessons-for-the-hour-of-code

Whitaker, M. C., & Valtierra, K. M. (2018). Enhancing preservice teachers' motivation to teach diverse learners. *Teaching and Teacher Education, 73*, 171-182.

Allyson Hallman-Thrasher, Ph.D., *is an Associate Professor of Mathematics Education at Ohio University. Her research focuses on teachers' learning to facilitating to classroom mathematics discourse and the preparation of K-12 mathematics teachers. She can be reached at hallman@ohio.edu*



Chang Lui, Ph.D., *is a professor in computer science at Ohio University, obtaining his doctoral degree in Information & Computer Science from the University of California at Irvine in 2002. His research focuses on software engineering, artificial intelligence, immersive technologies, and their use in learning and medical applications.*

Courtney Koestler, Ph.D., *is the Director of the OHIO Center for Equity in Mathematics and Science and an Associate Professor of Instruction in the Department of Teacher Education at Ohio University.*

Susanne Strachota, Ph.D., *is a post-doctoral researcher at Tufts University. Her research focuses students' early algebra learning and teacher actions to promote algebraic thinking.*

Jen Thompson, M.Ed., *is a Ph.D. student in The Patton College of Education at Ohio University. She has 20 years of experience as an elementary educator and mathematics instructional coach.*

This is a refereed article.