



Challenges and Unexpected Affordances of Physical Computing Going Remote

Alexandra Gendreau Chakarov
University of Colorado Boulder
Boulder, CO, United States
agc@colorado.edu

Jeffrey Bush
University of Colorado Boulder
Boulder, CO, United States
jeffrey.bush@colorado.edu

Quentin Bidy
University of Colorado Boulder
Boulder, CO, United States
quentin.bidy@colorado.edu

Jennifer Jacobs
University of Colorado Boulder
Boulder, CO, United States
jennifer.jacobs@colorado.edu

Colin Hennessy Elliott
Utah State University
Logan, UT, United States
colin.hennessyelliott@usu.edu

Tamara Sumner
University of Colorado Boulder
Boulder, CO, United States
sumner@colorado.edu

ABSTRACT

Engaging in physical computing activities involving both hardware and software provides a hands-on introduction to computer science. The move to remote learning for primary and secondary schools during the 2020-2021 school year due to COVID-19 made implementing physical computing activities especially challenging. However, it is important that these activities are not simply eliminated from the curriculum. This paper explores how a unit centered around students investigating how programmable sensors that can support data-driven scientific inquiry was collaboratively adapted for remote instruction. A case study of one teacher's experience implementing the unit with a group of middle school students (ages 11 to 14) in her STEM elective class examines how her students could still engage in computational thinking practices around data and programming. The discussion includes both the challenges and unexpected affordances of engaging in physical computing activities remotely that emerged from her implementation.

CCS CONCEPTS

• **Social and professional topics** → **Computational thinking; K-12 education; Model curricula.**

KEYWORDS

Computational Thinking, K-12 Computing Education, Physical Computing, Data Science

ACM Reference Format:

Alexandra Gendreau Chakarov, Jeffrey Bush, Quentin Bidy, Jennifer Jacobs, Colin Hennessy Elliott, and Tamara Sumner. 2021. Challenges and Unexpected Affordances of Physical Computing Going Remote. In *Interaction Design and Children (IDC '21)*, June 24–30, 2021, Athens, Greece. ACM, New York, NY, USA, 7 pages. <https://doi.org/10.1145/3459990.3460711>

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

IDC '21, June 24–30, 2021, Athens, Greece

© 2021 Association for Computing Machinery.

ACM ISBN 978-1-4503-8452-0/21/06...\$15.00

<https://doi.org/10.1145/3459990.3460711>

1 INTRODUCTION

Integrating computer science and computational thinking activities into existing parts of the primary and secondary school curriculum allows students to see how computing is broadly applicable. Science classes provide an obvious point for integration because it is a discipline that is increasingly reliant on the power of computational concepts and tools [2, 14]. Some of these computational tools involve the development of computational models and simulations [26, 36], but others depend on physical devices to collect and display information [5, 11, 13, 24]. The Next Generation Science Standards (NGSS) [21] specifically call out computational thinking as one of the eight science and engineering practices [34]. In addition, current best practice in science education emphasizes making science class resemble the work of real scientists [21, 25, 30]. One strategy for doing this is introducing complex computational tools into the classroom to support students in conducting meaningful investigations.

Programmable sensors are an example of physical devices used by real scientists. They are becoming increasingly accessible to primary and secondary educators through their decreasing costs and increasingly friendly user interfaces [1, 6]. Collecting information about their environment using these sensors can prompt students to ask questions such as *why here?* and *so what?* [8] and relate the information they are learning to their own local context [1, 20, 35].

Integrating computing into science classes enables the exploration of students' computational thinking in the context of their science classes. The Computational Thinking in Mathematics and Science Practices Taxonomy (CT Practices) [37] provides a framework defining what computational thinking looks like in science and mathematics classes. The CT Practices describes four categories of practices that students exhibit when they engage in computational thinking: *Data Practices*, *Computational Modeling and Simulation Practices*, *Computational Problem Solving Practices*, and *Systems Thinking Practices* [37, p. 135]. The term practice illustrates that both knowledge and skill are required for students to deeply engage in computational thinking. Each category contains a set of specific practices.

This paper describes how an in-person physical computing unit designed to be implemented in science classrooms was adapted for remote instruction due to the COVID-19 pandemic. Like most units involving physical computing, this unit was designed assuming

that students would have access to all the materials they need to assemble and manipulate the hardware/software system. In this case, the students were provided with the materials in their homes, but all instruction took place within a remote learning environment. This paper examines the teacher's implementation of an introductory unit in which students learned to program sensors to collect and display information about the world around them. This paper focuses on students' engagement with four of the computational thinking practices, Data Practices: *Collecting Data and Visualizing Data*, and Computational Problem Solving Practices: *Programming and Troubleshooting and Debugging*, which are core practices students engage in during the unit. This paper addresses the following research questions (1) *How can students engage in computational thinking practices around data and programming in a remote physical computing unit?* and (2) *What challenges and affordances emerge from engaging in physical computing activities remotely?*

2 BACKGROUND

Physical computing has a long tradition of supporting youth to engage with computing concepts [6]. Physical computing devices have evolved over the years through explicitly designing for children and the expansion of the hobbyist market into the educational realm [6]. Examples include robotics like LegoMindstorms [31, 32], wearable devices such as the LilyPad [7], and kits designed specifically for education such as Dash and Dot¹. These tools vary in their design and intended use, with some intentionally including programming interfaces to expose students to complexities from working with hardware/software systems. In contrast, others focus on electronics and assembly.

Working with both hardware and software requires teachers to support students working in both mediums simultaneously. Different strategies exist to support students to engage with these hardware/software systems ranging from free play where computer science concepts are introduced as they arise due to exploration and trial and error [27] to direct instruction of computing topics followed by independent projects [23] to a combination of both [17, 19]. This paper uses a form of guided inquiry often used to build science curricula [29, 30]. Units are centered around students designing investigations to answer their questions through the creation of a sequence of coherent lessons that support incremental knowledge building [28, 33].

Sensors are important physical computing tools that allow information to be taken in about the world. The increased availability of low-cost mobile sensors [1] is making it easier to measure and display information that was previously impossible for students to see. Microcontrollers such as the micro:bit [3] enable students to easily collect and respond to information from the environment through the incorporation of built-in sensors, such as temperature or light sensors. These sensor technologies provide new and innovative ways to understand and experience the world around us [18]. Collecting and creating data using sensors provides opportunities for students to reason and think critically about the accuracy and reliability of such information [22]. Students need to develop the knowledge and skills to evaluate the benefits and limitations of

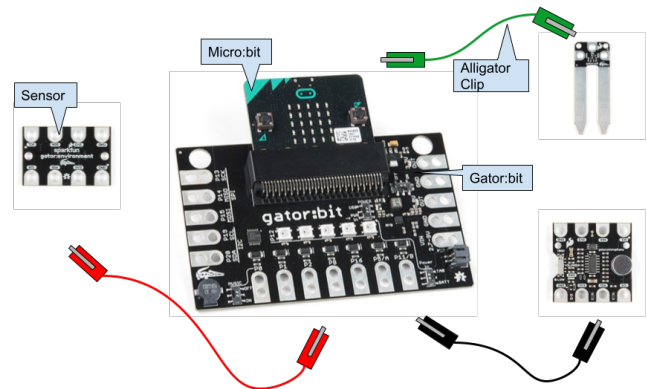


Figure 1: The micro:bit is a microcontroller that can be programmed using MakeCode, a block-based interface to support programming physical devices. The gator:bit enables alligator clippable sensors to be attached to the micro:bit for easy prototyping. Sensors include a sound sensor, environmental sensor, and soil moisture sensor. The gator:bit also includes a speaker and five neopixel LEDs.

sensor technologies [22], especially when selecting tools to use for their scientific investigations [16].

3 RESEARCH CONTEXT

This study is part of an existing research-practice partnership (RPP) [9, 10] between a large public university in the southwestern United States and a large, urban school district near the university. This RPP focuses on the problem of practice of providing all middle school students (ages 11 to 14) with equitable exposure to computational thinking by developing curriculum and professional learning activities to support inservice teachers to integrate computational thinking into required science and STEM classes² [4, 15]. These activities involve using programmable sensors built around the micro:bit, see Figure 1, that allow students to collect and display data from their local environment.

A design based implementation research (DBIR) approach [12] is used to study and iteratively refine the process. Each school year corresponds to one design cycle, and the project is currently in the middle of its fourth design cycle. Each design cycle involves teachers and researchers working together to design and adapt activities that the teachers implement in the classroom. After implementation, teachers reflect on their experiences and suggest improvements and modifications to the curricular materials.

The unit is based on a unit described in [16] that introduces students to programmable sensors as tools for scientific inquiry. A group of fifteen middle school science and STEM teachers and four university researchers modified the unit for remote instruction during a five-day workshop in the summer of 2020 conducted over Zoom. Each of the teachers had all of the sensor technology physically with them. Teachers were first introduced to the unit as initially designed and then examined each lesson to determine how best to modify the unit for remote implementation.

¹<https://www.makewonder.com/>

²STEM is a required elective class in this school district that explores topics not covered in traditional science and math classes.

3.1 Modified Sensor Unit

The unit is built around students' questions elicited by data displays their teachers built using the programmable sensor technology. Anticipated questions include *what are the different parts?*, *what controls the system?*, and *how does information move?*. The unit consists of four core lessons plus an assessment designed to take place over approximately five to ten one-hour class periods. The large variance in expected time is meant to provide flexibility to teachers who may want more time for students to explore the technology and to go through optional supports in the unit, such as an activity where students learn how to create explanatory models. Table 1 illustrates the goal of each core lesson, along with suggested strategies for remote implementation.

4 STUDY DESIGN

Five of the fifteen teachers participating in the research-practice partnership implemented the sensor unit during Fall 2020. Each teacher worked closely with one researcher during their implementation(s)³ who offered ongoing support, responded to questions as they arose, and collected data. Not all teachers implemented the unit in Fall 2020 due to the circumstances of their remote instruction (e.g., several teachers had remote classes of at least 60 students).

This paper describes Dawn's⁴ third implementation of the unit. Dawn joined the research-practice partnership in Summer 2020 and had not taught the unit before. She had over 20 years of teaching experience and had taught STEM for seven of those years. She had some experience with programming using Scratch and other block-based programming environments, but not MakeCode. Her third implementation of the unit was with a class that consisted of 36 middle school students, about 20% of whom were nonwhite. Dawn's third implementation was chosen to be the focal point of this paper because all students had the technology in hand. She refined her remote instruction over the two previous implementations to streamline many logistical issues, thus allowing the unit's content to be the main focus.

During her first two iterations of the sensor immersion unit, Dawn and the first author met after the majority of her lessons to discuss how the unit was going and brainstorm additional strategies and supports. Based on these discussions, Dawn made two significant changes for her third implementation. These changes aimed to streamline the process so time would remain to complete Lesson 4 and the assessment. First, Dawn added whole-class demonstrations where she assembled the gator:bit and sensors in front of her video camera while her students followed along with their equipment. Second, Dawn and the first author collaborated to create a new student tutorial in MakeCode that was more closely tied to Dawn's data display. The original tutorial focused on students building a display to monitor carbon dioxide, whereas Dawn's data display monitored the temperature and humidity. Her students struggled with the focus on carbon dioxide monitoring since it was unrelated to Dawn's display.

³Due to new schedules designed to accommodate remote learning, some teachers implemented the unit multiple times throughout the semester.

⁴All names are pseudonyms.

4.1 Data Collection

Dawn conducted all of her classes over GoogleMeets, as required by her school due to the COVID-19 Pandemic. The first author attended and recorded 10 of the 16 classes from her third implementation of the sensor immersion unit. Recording classes required Dawn to email the researcher a new Google Meets link every day, and for six of her classes, the link either did not arrive in time or was not sent. Each class lasted approximately 35 minutes, and the first author debriefed with Dawn after the majority of the classes. The debriefs were also recorded. After completing the third implementation of the unit, the first author conducted a 40-minute semi-structured interview with Dawn about her experience teaching the unit over time. Questions cover the implementation experience as a whole (e.g., what do you think your students learned or took away from the unit?), specific questions about each lesson (e.g., in Lesson 4, how well were your students able to communicate what they discovered during the sensor investigation?), and a discussion around the challenges of remote learning. This interview protocol has been refined over four design cycles.

4.2 Data Analysis

The four core CT Practices [37] addressed in the unit (*Collecting Data*, *Visualizing Data*, *Computer Programming*, and *Troubleshooting and Debugging*) provide a framework to conduct a qualitative, deductive analysis of how students engaged in computational thinking during the unit. Table 2 describes how each of these practices is represented in the unit as designed.

All video for the third implementation was timestamped and content logged based on Dawn's interactions with her students. Based on the content log, the first author wrote descriptions of how students engaged in each of the four CT Practices, see Table 2. Two other researchers reviewed the content log and descriptions. The three researchers then met for one hour to refine the descriptions. The descriptions indicated that Lesson 2 and Lesson 3 were the main lessons that students engaged in the CT practices and represented the majority of students' interactions with the sensor technology. The focus would be on Lesson 2 and Lesson 3 for the remainder of the analysis.

All videos from Lesson 2 and Lesson 3 were transcribed using automated transcription software and cleaned by the first author. The descriptions were refined to include only data from the second and third lessons and augmented with quotes from Dawn that illustrated how she supported her students to engage in the CT practices. The researchers then met for another hour to review the descriptions and come to an agreement on their content. In addition, all three researchers reviewed the transcription of Dawn's post-implementation interview and used that to provide context regarding her instruction.

5 RESULTS

This section describes how Dawn's students engaged in each of the four targeted CT Practices during her third implementation of the sensor immersion unit.

Table 1: The lesson goals and strategies to support students to achieve these goals during remote instruction

Lesson	Lesson Goal	Remote Implementation and Supports
Lesson 1	Students generate questions about the data display built by their teacher	Teachers either create a video of the data display or interact with the data display at the beginning of the lesson. Students create a virtual representation of all their questions
Lessons 2 & 3	Students figure out how the sensors collect and display data and create their own data displays. They use the magnetometer on the micro:bit to collect magnetic field data and display it on the micro:bit. They use another sensor to create a display controlled by sensor values.	To support small group work, students are divided into breakout rooms and assigned specific roles such as programmer, wirer, and debugger to go through a series of tutorials to help them learn about the sensors. Students take screenshots and pictures of their sensors and programs to enable teachers to assess progress.
Lesson 4	Students develop a broad conceptual understanding of how ALL sensors work by sharing their displays and seeing the similarities in their programs. Students brainstorm other scientific questions the sensors could help them answer.	Students create videos describing their data displays, including both the hardware and the program. Students use a virtual platform to identify similarities and differences across sensors. Students add questions the sensors could help them answer to a virtual board and go into breakout rooms to discuss the questions.

Table 2: Descriptions of the four CT Practices that students most deeply engage with throughout the unit.

CT Practice	Description
Collecting Data	Students figure out how the sensors can be used to gather data and design systematic ways to collect the data they need for their investigations.
Visualizing Data	Students explore both numerical and interactive displays to communicate the results of the data collection.
Computer Programming	Students modify and create programs to collect and visualize the data from the sensors using MakeCode. Students explore computer science concepts such as conditional logic and variables.
Troubleshooting and Debugging	Students face issues with both the hardware and software not behaving as expected and work to identify and correct the issues.

5.1 Collecting Data

During Lesson 2 of the remote version of Dawn’s sensor immersion unit, students engaged in an activity where they used a magnetometer to explore the magnetic fields of items in their homes independently. One student discovered that her mother’s wedding ring measured 3027 microteslas, which led the class to inquire about the highest possible magnetometer reading. Dawn remarked during her post-implementation interview that

I had a lot of kids running around their houses telling me I got 3295. And they come back and put it up [on our shared chat]... Some of them who are those timid ones even were putting stuff in the chat. Like saying, “Is 75 right? How are they getting those 1000s?” And then I said, “Well, put it next to something magnetic.” And then they go, “Oh, I got it.” You know, you can just see the discovery process.

After collecting data using the magnetometer, students moved on to using a different sensor to collect data about temperature, humidity, or soil moisture. Unlike the magnetometer, these environmental sensors have to be wired onto the micro:bit. Dawn directed students to look at a series of wiring diagrams and demonstrated how to assemble the wires on the environmental sensor by holding the pieces up to her camera. In return, students showed her their setups by holding them up to their cameras. Dawn explained what

the different colored wires meant by telling them, “Green which is data. It’s collecting data. You will use the green [wire] in both of the sensors, but you will not use the yellow [wire], which is like a controller of the data.”

Once they had their sensors wired correctly, students collected data on the temperature, humidity, or soil moisture in their home or neighborhood environments. Students moved at varying paces through the activity and did not share their data as much as they did during the magnetometer activity. Students spent more time exploring with the soil moisture sensor than with the temperature and humidity sensors, which Dawn attributed to them asking more questions about and seeing more variation in the data. She shared, “Some of them did put it in a plant. And some of them did say ‘What’s the difference between this succulent plant and this ivy?’ or whatever it was.”

5.2 Visualizing Data

About half of Dawn’s students completed an extension activity to create a metal detector using the magnetometer on the micro:bit. The metal detector displays a picture on the micro:bit that gets brighter when it gets closer to magnetic objects. Students use the battery pack to make their metal detector mobile and go search for metal. Dawn instructed her students to go outside and use the micro:bit to search for metal objects.

When students collected data using the environmental sensor or the soil moisture sensor, they displayed their data using single input events at first (e.g., when you press button A, the temperature value scrolls across the micro:bit LEDs). Next, Dawn encouraged the students to explore how they could display all the values collected using just one input event, “So that when you push a button, you get all the data. One button. Boom. Bam. Done.” She had them think about how to create labels using the LEDs on the micro:bit to tell the viewer what type of data they see, “so once it [your program] has found the number, how would you get to show it in a string, like T equals and then that number?”

5.3 Programming

Students’ first experience with programming in the unit came when they followed a tutorial that walked them through how to program a magnetometer in MakeCode to collect data on magnetic field strength. Dawn went through the tutorial with the students by sharing her screen and showing the students select aspects of MakeCode. For example, Dawn explained how different blocks could control when data is displayed using the simulator.

I’m just going to show number 10. If it’s on start, do you notice what it did over on my micro:bit? As soon as I plug it in, as soon as it’s turned on, it’s going to do that? Right? If I change that, and I put it in forever. Notice, it’s going to just keep going and going, it’s going to show that number for as long as it’s on.

After the students programmed the magnetometer, they worked on a tutorial to replicate the data display that anchored the unit. They followed a tutorial that walked them through how to collect temperature, humidity, and soil moisture data using different buttons to collect different data. Dawn introduced the students to variables by showing a short video. She presented a challenge where students had to modify their program to collect all the data using only one button instead of three different buttons like the tutorial, “then I want you to go back when you’re finished. And think about this. If I wanted all that data on button A, what would I do?”

5.4 Troubleshooting and Debugging

Students engaged in troubleshooting and debugging throughout the programming and wiring process. The alligator clips that held the various components together were not particularly sturdy, especially when students carried the equipment around in their homes to make measurements. To prevent the alligator clips from coming unclipped, Dawn demonstrated, by holding the hardware up to her camera, how to connect the alligator clips using both holes on the gator:bit to ensure a secure connection. Dawn cautioned, “When you put it [the alligator clip] in, you want to make sure that it goes through the two holes.”

When students reported data values that did not make sense, Dawn encouraged them to examine both their program and their hardware; however, she emphasized the software portion as the first place to look. For example, in the case of the unusual temperature value, Dawn told the student, “Go back to your code and see what possibly could be up. And then also check where your sensor is and your wiring for sure.” She would follow up with the student later

in the class to confirm they had identified and fixed the bug. If the student still struggled, she encouraged them attend office hours.

When a student had an issue with their program, Dawn asked them to share their screen so she and other students could help them debug the program. First, she asked students if they could spot the problem. Dawn then helped the whole class understand the problem by illustrating the consequences of the wrong code.

In one case, a student encountered problems displaying the values from the temperature sensor. The whole class looked at the student’s code together, and a classmate responded that the code is missing the *on start* block with the initialization. Dawn added that “Yep, you definitely always have to have that *on start*. If you don’t tell it to start that gator:environment, it will never do all those buttons and code that you’ve started to put on there.” Dawn encouraged the class to read over the student’s code, saying, “So let’s just read what you have there. If I press A, then it’s going to go get the temperature.” Then she illustrated what will happen if the student tried to display all sensor values when button A is pressed “If I press A and it gives me like just four numbers. I won’t know what any of those are, it might look like it’s 1000s of degrees or something.”

6 DISCUSSION

This section explores the significant themes that emerged from Dawn’s implementation and situates the challenges and affordances in the larger context outside of just remote learning and into the physical computing activities in general.

RQ1) How can students engage in computational thinking practices around data and programming in a remote physical computing unit? Dawn’s students successfully engaged in the four CT Practices most relevant to the sensor immersion unit. The *Data Collecting Practice* emerges in two main ways in the unit. First, the process of assembling the hardware is critical to students being able to collect data from the additional sensors not built into the micro:bit. Second, students explored different objects and spaces in their homes and shared that data to develop a collective understanding of their available information. For the *Visualizing Data Practice*, students used the LEDs on the micro:bit to display the numerical value of the sensor data, added labels to that numerical data so that the visualization had just one input event, and some students created an interactive visualization in the form of a metal detector that got brighter the closer the micro:bit was to a metal object. Students engaged with the *Programming Practice* by creating several different programs in MakeCode throughout the unit. *Troubleshooting and Debugging* encompassed not only code that is not performing as expected but also issues around hardware assembly and sensor accuracy. In addition, students used strategies such as reading through the program, screen sharing to solve a problem collaboratively, and the process of elimination to determine if the problem is hardware or software related.

However, the extent to which all students engaged in the CT Practices appears to lack great depth, likely due to the relatively short time frame of the unit. The examples provided in the analysis suggest a certain level of sophistication in terms of the *Collecting Data Practice* and *Troubleshooting and Debugging Practice*, but to truly engage with CT Practices of *Visualizing Data* and *Computer*

Programming as described by Weintrop and colleagues [37], more in-depth interactions need to be present such as programming interactive visualizations that involve data analysis.

RQ2) What challenges and affordances emerge from engaging in physical computing activities remotely?

Implementing physical computing activities remotely is challenging and not an ideal situation. Four major challenges emerged from Dawn's implementation: (1) an increase in the amount of direct instruction, (2) less opportunity for exploration and student choice, (3) debugging physical devices remotely, and (4) supporting small group work.

She had less instructional time than in her regular classes because of additional hurdles such as the extra time needed for all students to join the GoogleMeet and technical issues that inevitably arose. Given the decreased class time, she went through tutorials with students instead of having them explore them on their own first. Thus the unit became a more blended instructional unit [17, 19] instead of being as grounded in student investigations. Debugging hardware issues required students to hold up the sensors to the camera, making it challenging to view the system thoroughly. Simple errors like a flipped power switch that are quickly discovered when students are in person took minutes to figure out using the camera. GoogleMeets did not have easily accessible breakout rooms when Dawn implemented the unit. She had to create individual GoogleMeets to support small group work. Getting students into these GoogleMeets and ensuring that all students were participating presented an insurmountable challenge for Dawn. She felt her time was better spent creating that whole class collaborative feel by focusing on full group discussion.

While Dawn encountered various challenges, there were also some unexpected affordances of the remote environment in her implementation. All students were provided with the sensor technology and had to engage in the building and programming themselves. They could not depend on other students to do it for them. This circumstance offered all students a degree of agency in the process they might not have otherwise had by removing some of the issues that can arise from collaborative computing projects, such as one or two students taking control of the entire group. Students were also able to use the sensors to investigate their home environment, making the data more meaningful and more varied compared to data collected within a single classroom. Lastly, during the troubleshooting debugging process, Dawn or other students could not immediately step in and physically fix the problem; they could only guide them to fix their problem through screen sharing: one piece of remote learning technology that added to the learning experience.

7 CONCLUSION

This paper describes how students can engage in data and computational problem solving practices remotely with physical computing devices. Challenges existed around opportunities for independent exploration, small group work, and debugging physical devices. However, students successfully built and used the sensor technology at home and collected and visualized data from their local environment. Teachers and school administrators are often unwilling to send equipment home with students, but Dawn's experience

suggests that doing so can afford a positive and productive learning opportunity. All of Dawn's technology was returned, and there have been no issues with any of it breaking. Students were often compelled to work through complex computational problems themselves because no teacher or other student could fix it for them. While this study only examined one teacher's experience, these affordances provide an opportunity to reimagine physical computing activities for both remote and in-person learning.

ACKNOWLEDGMENTS

This work is supported by the National Science Foundation Award No. 1742053 and No. 1742046 and the James S. McDonnell Foundation. We want to thank the anonymous reviewers for their comments and suggestions. Our work would not be possible without the teachers and students who allow us into their classrooms.

8 SELECTION AND PARTICIPATION OF CHILDREN

Before implementation, Dawn provided parents with a parental consent form that briefly described the larger project and the kind of information the research team would be collecting. The consent form stated that students should still participate as usual in the class even if they did not consent to have their data used in the study. Students received a similar form written in student-friendly language where they could provide their assent to participate. With this in mind, for parents or students who do not consent, every effort was made to ensure that the student was not recorded and no data was collected from them. All student data is anonymized, names are changed, and videos and images are blurred to protect student identity.

REFERENCES

- [1] Stamatina Anastopoulou, Mike Sharples, Shaaron Ainsworth, Charles Crook, Claire O'Malley, and Michael Wright. 2012. Creating personal meaning through technology-supported science inquiry learning across formal and informal settings. *International Journal of Science Education* 34, 2 (2012), 251–273.
- [2] Norman R Augustine et al. 2005. Rising above the gathering storm: Energizing and employing America for a brighter economic future. Retrieved March 19 (2005), 2008.
- [3] Jonny Austin, Howard Baker, Thomas Ball, James Devine, Joe Finney, Peli De Halleux, Steve Hodges, Michał Moskal, and Gareth Stockdale. 2020. The BBC micro: bit: from the UK to the world. *Commun. ACM* 63, 3 (2020), 62–69.
- [4] Quentin Biddy, Alexandra Gendreau Chakarov, Jeffrey Bush, Colin Hennessy Elliott, Jennifer Jacobs, Mimi Recker, Tamara Sumner, and William R Penuel. 2021. A Professional Development Model to Integrate Computational Thinking into Middle School Science through Co-designed Storylines. *Contemporary Issues in Technology and Teacher Education* 21, 1 (Mar 2021).
- [5] Paulo Blikstein. 2012. Bifocal modeling: a study on the learning outcomes of comparing physical and computational models linked in real time. In *Proceedings of the 14th ACM international conference on Multimodal interaction*. ACM, 257–264.
- [6] Paulo Blikstein. 2013. Gears of our childhood: constructionist toolkits, robotics, and physical computing, past and future. In *Proceedings of the 12th international conference on interaction design and children*. 173–182.
- [7] Leah Buechley, Mike Eisenberg, Jaime Catchen, and Ali Crockett. 2008. The LilyPad Arduino: using computational textiles to investigate engagement, aesthetics, and diversity in computer science education. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. 423–432.
- [8] Cory A Buxton. 2010. Social problem solving through science: An approach to critical, place-based, science teaching and learning. *Equity & excellence in education* 43, 1 (2010), 120–135.
- [9] Cynthia E Coburn and William R Penuel. 2016. Research–practice partnerships in education: Outcomes, dynamics, and open questions. *Educational Researcher* 45, 1 (2016), 48–54.

- [10] Cynthia E Coburn, William R Penuel, and Kimberly E Geil. 2013. Research-practice partnerships: A strategy for leveraging research for educational improvement in school districts. *New York, NY: William T. Grant Foundation* (2013).
- [11] Lila Finch, Celeste Moreno, and R Benjamin Shapiro. 2020. Teacher and student enactments of a transdisciplinary art-science-computing unit. *Instructional Science* 48, 5 (2020), 525–568.
- [12] Barry J Fishman, William R Penuel, Anna-Ruth Allen, Britte Haugan Cheng, and NORA Sabelli. 2013. Design-based implementation research: An emerging model for transforming the relationship of research and practice. *National society for the study of education* 112, 2 (2013), 136–156.
- [13] Bjørn Fjukstad, Nina Angelvik, Maria Wulff Hauglann, Joachim Sveia Knutsen, Morten Grønnesby, Hedinn Gunhildrud, and Lars Ailo Bongo. 2018. Low-Cost Programmable Air Quality Sensor Kits in Science Education. In *Proceedings of the 49th ACM Technical Symposium on Computer Science Education*. ACM, 227–232.
- [14] Ian Foster. 2006. A two-way street to science's future. *Nature* 440, 7083 (2006), 419–419.
- [15] Alexandra Gendreau Chakarov, Quentin Bidy, Jennifer Jacobs, William Penuel, Mimi Recker, and Tamara Sumner. [n.d.]. Professional development supporting middle school teachers to integrate computational thinking into their science classes. In *Preparing Teachers to Teach Computer Science: Models, Practices and Policies*, C. Mouza, A. Yadav, and A. Leftwich (Eds.). Information Age Publishing, Charlotte, NC. forthcoming.
- [16] Alexandra Gendreau Chakarov, Quentin Bidy, Jennifer Jacobs, Mimi Recker, and Tamara Sumner. 2020. Opening the Black Box: Investigating Student Understanding of Data Displays Using Programmable Sensor Technology. In *Proceedings of the 2020 ACM Conference on International Computing Education Research*. 291–301.
- [17] Shuchi Grover and Roy Pea. 2016. Designing a blended, middle school computer science course for deeper learning: A design-based research approach. Singapore: International Society of the Learning Sciences.
- [18] Heather J Hether, Joe C Martin, and Andrew W Cole. 2017. The Internet of Things and wearable technology as a classroom resource. *New media and digital pedagogy: Enhancing the twenty-first-century classroom* (2017), 129–146.
- [19] Yasmin B Kafai, Eunyoung Lee, Kristin Searle, Deborah Fields, Eliot Kaplan, and Debora Lui. 2014. A crafts-oriented approach to computing in high school: Introducing computational concepts, practices, and perspectives with electronic textiles. *ACM Transactions on Computing Education (TOCE)* 14, 1 (2014), 1–20.
- [20] J Kahn and R Hall. 2016. Getting personal with big data: Stories with multivariable models about global health and wealth. In *annual meetings of the American Educational Research Association, Washington, DC Best Paper award, SIG for Learning Sciences and Advanced Technologies*.
- [21] NGSS Lead States. 2013. Next generation science standards: For states, by states.
- [22] Susan Lechelt, Yvonne Rogers, and Nicolai Marquardt. 2020. Coming to your senses: promoting critical thinking about sensors through playful interaction in classrooms. In *Proceedings of the Interaction Design and Children Conference*. 11–22.
- [23] Orni Meerbaum-Salant, Michal Armoni, and Mordechai Ben-Ari. 2013. Learning computer science concepts with scratch. *Computer Science Education* 23, 3 (2013), 239–264.
- [24] Leyla Norooz, Matthew Louis Mauriello, Anita Jorgensen, Brenna McNally, and Jon E Froehlich. 2015. BodyVis: A New Approach to Body Learning Through Wearable Sensing and Visualization. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. ACM, 1025–1034.
- [25] William R Penuel. 2016. Studying science and engineering learning in practice. *Cultural Studies of Science Education* 11, 1 (2016), 89–104.
- [26] Katherine Perkins, Wendy Adams, Michael Dubson, Noah Finkelstein, Sam Reid, Carl Wieman, and Ron LeMaster. 2006. PhET: Interactive simulations for teaching and learning physics. *The physics teacher* 44, 1 (2006), 18–23.
- [27] Mareen Przybylla and Ralf Romeike. 2012. My Interactive Garden—A Constructionist Approach to Creative Learning with Interactive Installations in Computing Education. *Constructionism: Theory, Practice and Impact. Proceedings of Constructionism 2012* (2012), 395–404.
- [28] BJ Reiser, M Novak, and M Fumagalli. 2015. NGSS storylines: How to construct coherent instructional sequences driven by phenomena and motivated by student questions. In *Illinois Science Education Conference, Tinley Park, IL*. <http://bit.ly/1sS1y26>.
- [29] Brian J Reiser. [n.d.]. Designing coherent storylines aligned with NGSS for the K-12 classroom.
- [30] Brian J Reiser, Michael Novak, and Tara AW McGill. 2017. Coherence from the students' perspective: Why the vision of the framework for K-12 science requires more than simply “combining” three dimensions of science learning. In *Board on Science Education Workshop “Instructional Materials for the Next Generation Science Standards*.
- [31] Mitchel Resnick, Fred Martin, Randy Sargent, and Brian Silverman. 1996. Programmable bricks: Toys to think with. *IBM Systems journal* 35, 3.4 (1996), 443–452.
- [32] Mitchel Resnick, Stephen Ocko, et al. 1990. *LEGO/logo-learning through and about design*.
- [33] Tricia Shelton. 2015. Climbing the NGSS mountain. *The Science Teacher* 82, 9 (2015), 65.
- [34] NGSS Lead States. 2013. Appendix F: Science and Engineering Practices in the NGSS. *Next Generation Science Standards: For States, By States*. (2013).
- [35] Katie Headrick Taylor and Rogers Hall. 2013. Counter-mapping the neighborhood on bicycles: Mobilizing youth to reimagine the city. *Technology, Knowledge and Learning* 18, 1-2 (2013), 65–93.
- [36] Seth Tisue and Uri Wilensky. 2004. Netlogo: A simple environment for modeling complexity. In *International conference on complex systems*, Vol. 21. Boston, MA, 16–21.
- [37] David Weintrop, Elham Beheshti, Michael Horn, Kai Orton, Kemi Jona, Laura Trouille, and Uri Wilensky. 2016. Defining computational thinking for mathematics and science classrooms. *Journal of Science Education and Technology* 25, 1 (2016), 127–147.