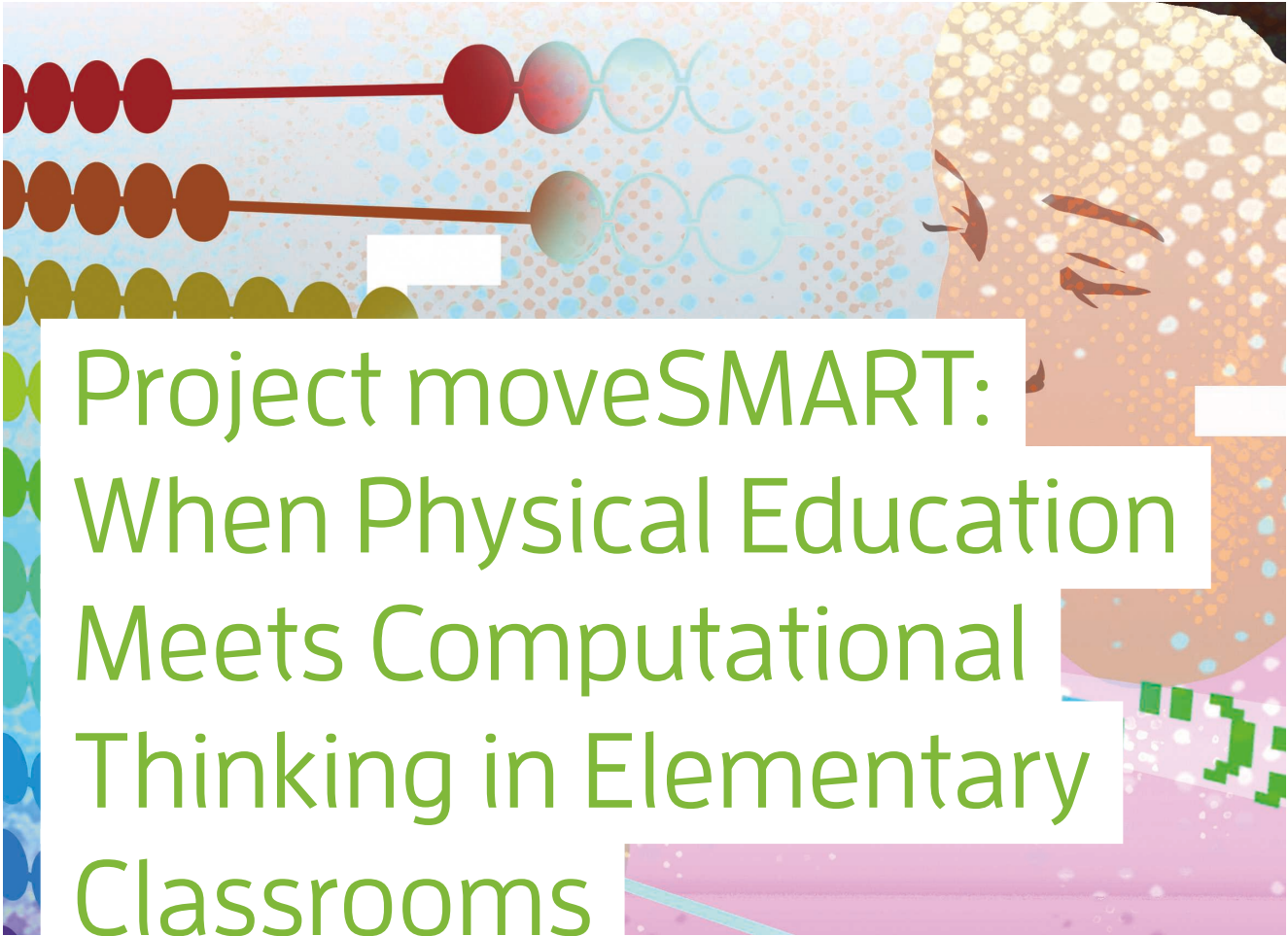




Project moveSMART: When Physical Education Meets Computational Thinking in Elementary Classrooms

Connor Fritz, Dylan Bray, Grace Lee, Christine Julien ,
Sheri Burson, and Darla Castelli, University of Texas at Austin

Carol Ramsey, CS Ed Group and University of Texas at Austin

Jamie Payton, Temple University

Project moveSMART uses a web-based platform to integrate physical education with computer science (CS) and computational thinking (CT). This article describes a series of tutorials to introduce elementary students to CS/CT by making connections to physical activity and grade-level curricula in other subjects.

The ubiquitous nature of digital technology has made computing critical in K-12 education, joining science, technology, engineering, and mathematics (STEM) skills as fundamental.¹ Yet,

formal expectations to integrate computer science (CS) and computational thinking (CT) into K-12 curricula have only recently been established, and many teachers have had little to no training in computing education.² Despite the great need and demand for such competencies, the inclusion of CS/CT curricula is spotty at best and non-existent at worst.³

Digital Object Identifier 10.1109/MC.2022.3167600
Date of current version: 24 October 2022

The need to address CS/CT in K-12 education is even more urgent when one considers racial and ethnic inequities. Disparities in STEM skills among Hispanic and Black students relative to White students are long standing. Access to quality CS/CT education is disproportionately lacking for students of color, students from low-income families, and female students.⁴ While students across demographic groups express interest in learning computing, Black and Hispanic students often encounter social barriers to participating in CS/CT, including stereotypes of who belongs in CS and parents' and educators' beliefs that underrepresented groups are not as interested in pursuing the subject.⁵ Students from low-income families face structural barriers (for example, a lack of home computers, CS courses in their schools, and extracurricular CS/CT opportunities) that limit access and exposure to CS. Despite the modern relevance of computing, state learning standards for elementary students rarely include CS/CT topics. While many teachers are enthusiastically supportive of teaching CS/CT, their ability to add to the curriculum is constrained by the need to improve with respect to state accountability standards and adhere to a provided curriculum.

Physical education is also increasingly neglected in elementary school despite its many demonstrated benefits. For children, physical activity (PA) is a predictor of adolescent health⁶ and success in school.⁷ Despite the benefits of PA, 80% of adolescents fail to meet the recommended hour of daily exercise.⁸ Racial and ethnic minority and economically disadvantaged youth show even lower PA rates than White and economically advantaged peers. Hispanic youth are significantly less

likely to participate in 60 min/day of PA than non-Hispanic youth,⁹ and only 24% of children from low-income families report participating in organized PA, compared to 49% of children from high-income families.¹⁰ Longitudinal studies reveal that childhood PA decreases with age, and recent findings suggest that PA begins to decline around nine.¹¹ This makes elementary school a prime candidate for interventions to increase student PA.

Although teachers may recognize the importance of CS and physical education, they also need to focus on delivering content aligned with state learning standards, which often do not involve PA and CS/CT. We address these challenges with Project moveSMART. Project moveSMART is a collaborative educational game built around a researcher-practitioner partnership (RPP) that includes teachers from multiple schools and school districts. It promotes increased PA and CS/CT while delivering content that aligns with state learning standards. In many cases, these three facets are integrated into the same content. For instance, in one learning activity, students program their own step counter, measure their steps as they complete a PA, then finish an assessment that includes questions involving inequalities (a topic covered in state learning standards). Project moveSMART also promotes PA through the online platform used to deliver educational content, as students increase their class score by logging higher rates of exercise.

This article details a Project moveSMART pilot study in which elementary school students completed a set of tutorials that combined PA, CS/CT concepts, and content aligning with Texas state learning standards. We found that fourth-grade students who participated in these tutorials showed

significant increases in their coding confidence and perception of coders. This study also brought insights concerning the benefit of incremental introduction of platform features and the importance of student engagement to success in CS/CT content delivery.

RELATED WORK

Various other projects have used an online platform and gamification to promote PA in students. In particular, Project moveSMART builds on KidsGoGreen,¹² a game that promotes sustainable transit and independent mobility for elementary aged students in Italy. Like Project moveSMART, KidsGoGreen takes students through a virtual journey during which they unlock educational content. However, KidsGoGreen does not include a focus on CS/CT concepts, and it does not directly integrate PA with the learning activities delivered through its online platform.

A relatively new area in CS education is physical computing, which involves using software and hardware to build physical systems and teach CS/CT concepts.¹³ Approaches that utilize physical computing often use embedded microcomputers, such as the BBC micro:bit,¹⁴ that are meant to be applied in educational contexts. While Project moveSMART employs physical computing in the CS/CT learning activities discussed in the following section, these learning activities also involve PA to further increase student engagement and encourage healthy behaviors. Another novel aspect of Project moveSMART is that content aligning with state learning standards is integrated throughout the learning activities delivered through the online platform. This enables teachers to justify devoting class time to activities that also cover CS/CT and encourage PA. To the best of our knowledge, no other

project has simultaneously addressed these issues.

THE PILOT STUDY

We piloted moveSMART in partnership with Hornsby–Dunlap Elementary School (HDES) in the Del Valle Independent School District. At HDES, 69% of the students are Hispanic, and 18.9% are African American. In 2018–2019, 27% of students met grade-level expectations in science, and 42% met the expectations in math.¹⁵ HDES is a Title I school; 86.5% of students qualify for free or reduced lunch. The fourth- and fifth-grade teachers, as well as the school's physical education teacher and principal, are part of the RPP and have worked as collaborators in developing moveSMART.

Figure 1 shows an overview of the moveSMART platform integrated with PA and the regular school curriculum. The platform hinges on an educational “game” played cooperatively by a class. In moveSMART, a class progresses through a virtual journey (for example, the fourth-grade route crosses Texas, while the fifth-grade route crosses the United States) when students participate in PA opportunities offered within the school day (for instance, in physical education class and at recess). Students log their PA by choosing one of “red,” “yellow,” or “green” levels, with red indicating little activity level and green indicating high activity. Students can log PA through a web-based check-in system or through a physical check-in

box. The box consists of a Raspberry Pi connected to a radio-frequency identification (RFID) card reader. Students scan an assigned RFID card, then choose their activity level by pushing a colored button. Pushing the button triggers an application programming interface request to the moveSMART cloud platform to store the student's activity level. When students check in, the score for their class increases, which moves the class farther along its virtual journey.

A moveSMART journey passes through “waypoints” with learning modules that incorporate curricular material from across disciplines, placed in the geographical or cultural context of the waypoint. The waypoints contain

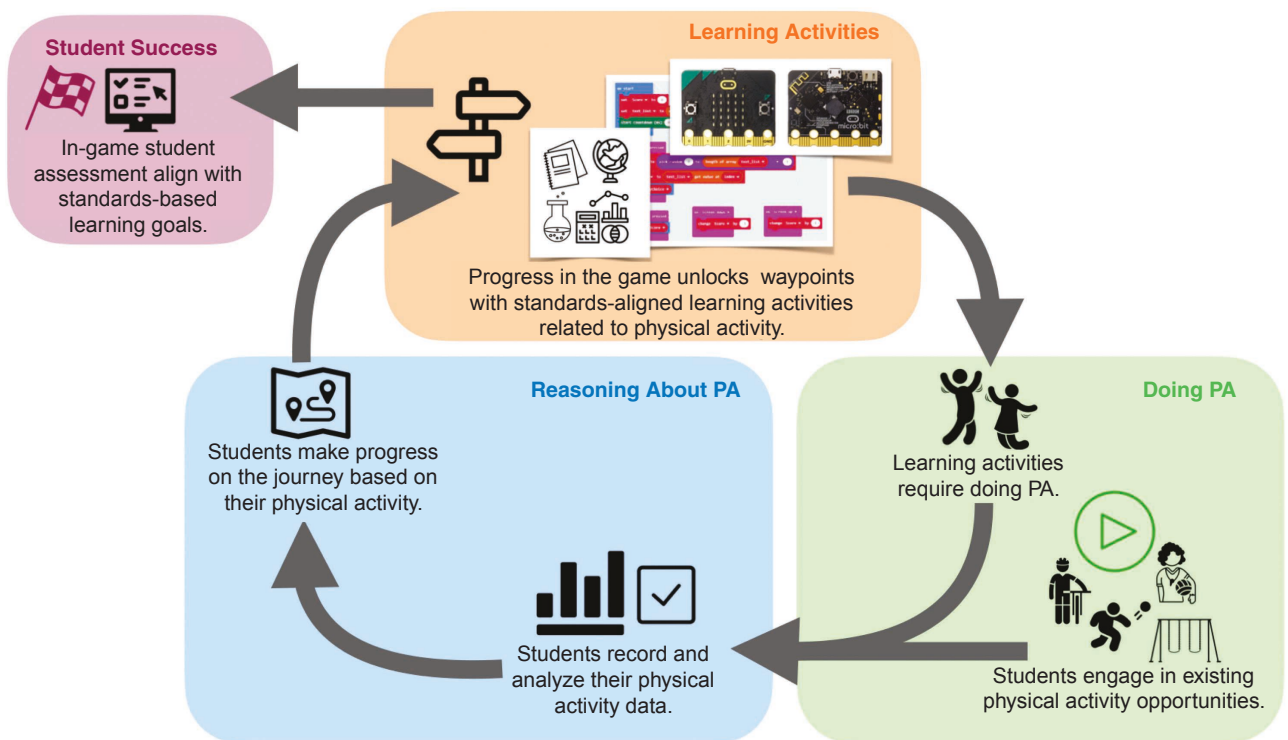


FIGURE 1. Project moveSMART. Starting from the bottom right, students engage in in-school physical activity (PA). They record their data in moveSMART, which moves the class through the journey. Progress unlocks waypoints, which contain learning activities across the curriculum. Learning activities 1) generate additional PA opportunities and 2) are tied to learning standards that are measured through in-game assessments.

embedded content, assessments, and CS/CT learning activities. This article focuses on the CS/CT activities in the waypoints; in the following, we describe series of activities through which students create their own wearable activity monitor and integrate its reports of sensed activity into the moveSMART game. These learning activities rely on the BBC micro:bit,¹⁴ a small computer built for educational purposes. The micro:bit is a physical computing device—a programmable computing system that can interact with its physical environment. By enabling students to program real-world devices, physical computing platforms concretely demonstrate the value of programming. Additionally, students from groups that are traditionally underrepresented in CS respond positively to educational interventions involving physical computing.¹⁶ The CS/CT learning activities we designed for moveSMART are meant to be completed in succession, as each

one builds on concepts introduced in earlier activities.

Each moveSMART CS/CT learning activity is also tied to grade-level components of the K–12 CS framework,¹⁷ a set of guidelines to develop CS educational standards and curricula. The framework consists of concepts and practices. Practices describe behaviors and ways of thinking that are expected of computationally literate students. Concepts are the major CS content areas that are relevant for computationally literate students. Concepts are divided into core areas: computing systems, networks and the internet, data and analysis, algorithms and programming, and impacts of computing, which are further delineated by subconcepts. Throughout the descriptions of the learning activities in the following, we tie each activity to the K–12 CS concept(s) it addresses.

In general, a learning activity starts by introducing students to relevant CS/CT content using embedded videos,

text, and examples. Students then complete a walk-through in Microsoft MakeCode,¹⁸ a coding environment in which students use blocks to create programs to run on a real or emulated micro:bit. Figure 2 presents an intermediate step of the second activity, which students undertake after learning about accelerometers. MakeCode provides a playground where the students can experiment. We developed a set of tutorials for MakeCode along with some moveSMART programming abstractions that enable us to hide some of the complexities of programming, which the learning activities incrementally remove as the students’ programming competence grow. In Figure 2, the students use a “show number of steps” block and an “increase step count” block from the “MoveSMART” tray in MakeCode. At this point in the curriculum, students have not yet been introduced to variables, so we hide them under an

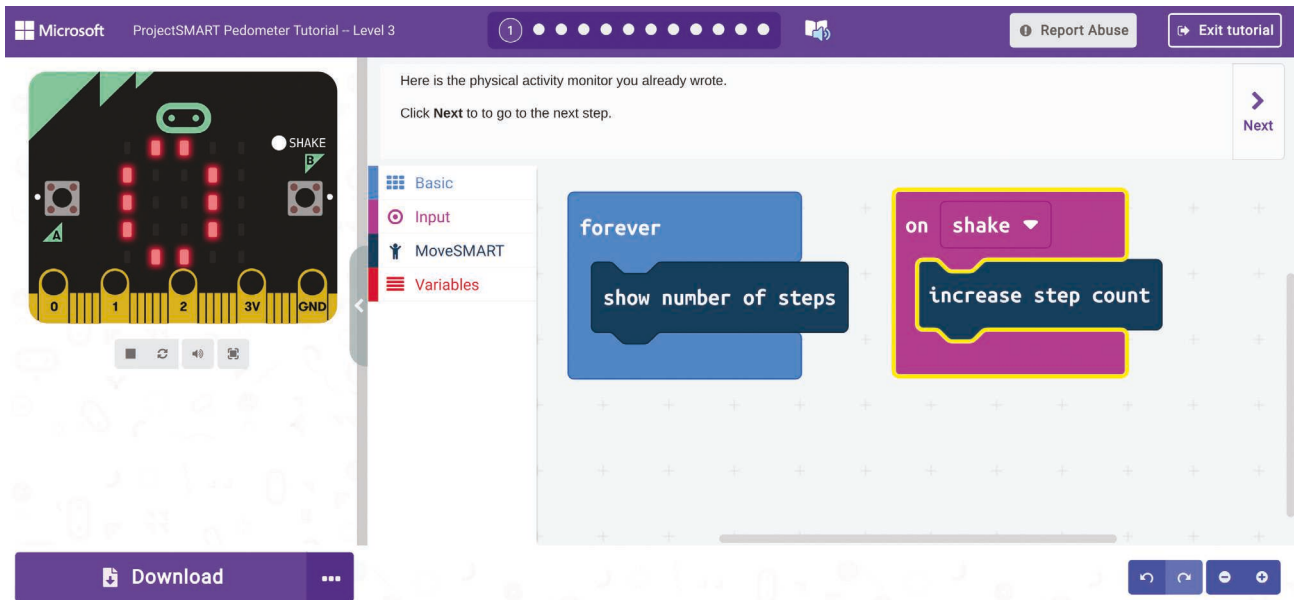


FIGURE 2. The second CS/CT learning activity in Project moveSMART, delivered through the MakeCode tutorial platform.

abstraction. At the end of each walk-through, students download their completed programs onto physical micro:bits to see them in action, use them for other classroom activities, or complete PA-related tasks.

To integrate CS/CT learning with moveSMART, we also developed in-app assessments. These were requested by teachers for all learning activities in the game, but they were essential for CS/CT because no other forms of assessment exist for these in the curriculum. As an example, Figure 3 describes the assessment that follows the fourth learning activity, which introduces control flow. We next walk through the seven CS/CT learning activities we

designed. To date, we have integrated the first five into the moveSMART learning platform. We piloted the first two at HDES during the 2020–2021 academic year. (Because of significant changes to elementary instruction in 2020–2021 due to COVID-19, most of our interactions were via virtual channels. However, in the final week of school, we did have one class period each with the fourth and fifth grades, where we piloted the CS/CT learning activities.)

Learning activity 1: Introduction

The first activity acclimates students to the micro:bit and MakeCode and guides them through creating a timer.

We use two short videos to introduce the micro:bit and the concept of a microprocessor. Students then follow a guided tutorial to construct a micro:bit timer. When the timer is complete, students work in pairs to time how long it takes each of them to complete a trail making test,¹⁹ a measure of cognitive flexibility. Upon completing this activity using pencils and paper, students return to moveSMART to complete an assessment. The assessment for this activity focuses on unit conversions between seconds and microseconds: “(1) Your timer counts seconds, but the micro:bit can also measure time in milliseconds: $1\text{ s} = 1,000\text{ ms}$. If the trail making task took your friend Robert 23 s, how many

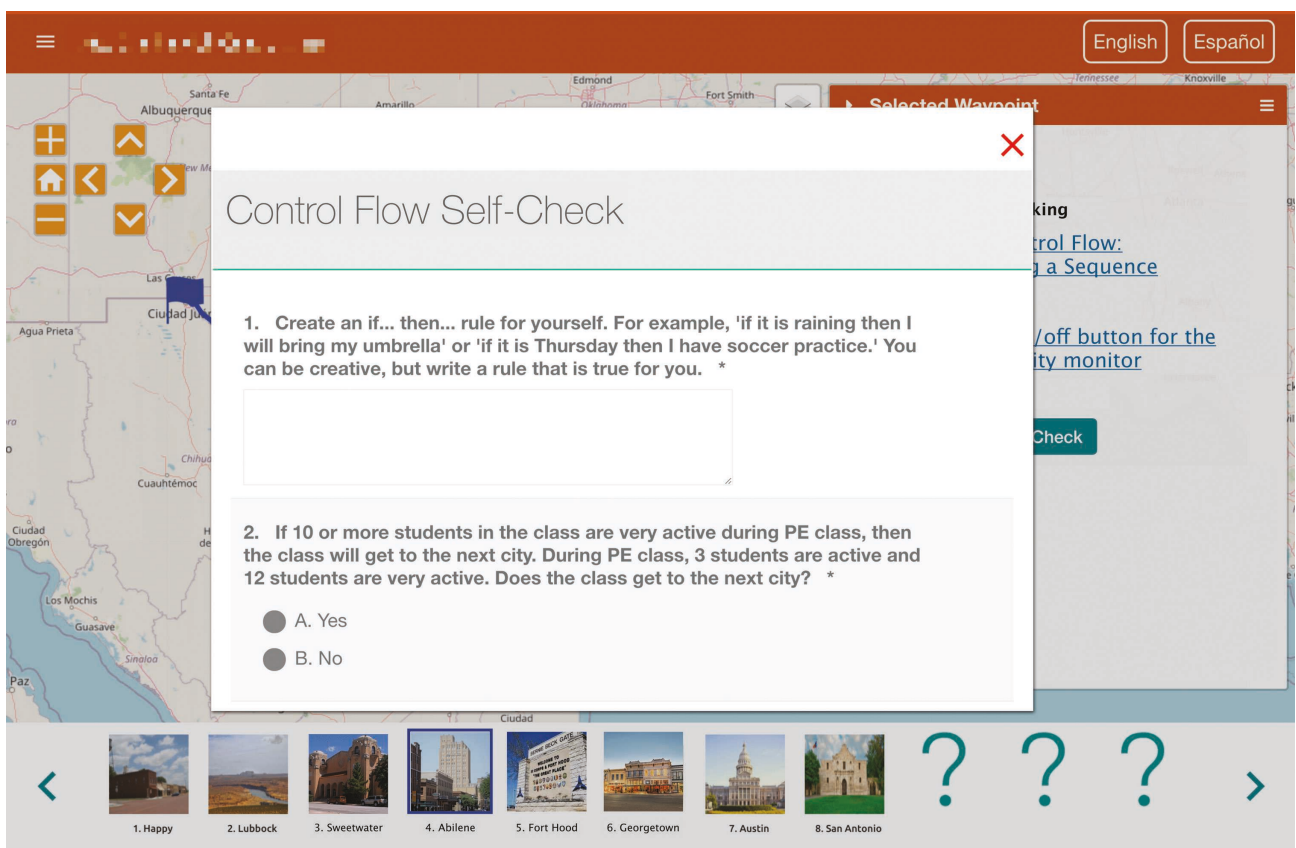


FIGURE 3. An assessment embedded into the Project moveSMART platform.

milliseconds did it take?” and “(2) If you took 22,923 ms to complete the trail making task and your friend Robert took 23 s, which one of you completed it faster?” These questions are aligned with the state-level mathematics standards for fourth and fifth grades in Texas. In addition, they prepare students to work with the native timers in MakeCode, which count time in milliseconds, rather than the moveSMART abstraction, for which we use seconds. The activity is connected to the hardware and software subconcept of the computing systems concept in the K-12 CS framework.

Learning activity 2: Sensing

We next introduce students to the concept of sensing, which is aligned with the devices subconcept of computing systems in the K-12 CS framework as well as the collection subconcept of data and analysis. We start with a PA that has a student intentionally move along three axes of acceleration [for example, “Step left, then right. That’s the first axis. Step forward, then backward. That’s the second axis. Where’s the third axis? (Hint: JUMP!)”]. We then show a video to introduce these axes within the micro:bit and explore, physically, how this relates to their real device. The students then use a MakeCode walk-through to create a step counter that uses the micro:bit accelerometer. Because students have not yet been introduced to variables to store data, this activity relies on abstractions.

When their step counters are complete, the students “wear” them (for example, by sticking them in their pocket or sock) and are guided through a PA with a partner. The students take turns playing charades, acting out the movements of an animal, and measure which movements generate more “steps” on their step counters. This activity provides an introduction to the

physical education concept of intensity. At the end of this activity, students are asked to express their results from the PA in terms of an inequality (for example, “Write an inequality that expresses how your animal activity compared to your partner’s. For instance, if I had 16 steps for acting out a snake, but my partner had 29 steps while acting out a bear, I would write $16 < 29$.”).

Learning activity 3: Variables

The third learning activity introduces students to variables. It explains variables through accessible language, pictures of MakeCode blocks, and animations of a virtual micro:bit. After reading through this content, students are routed to the MakeCode platform, which displays the program they wrote in the previous learning activity. Students are guided through refactoring their code to use variables. This activity is directly connected to the variables subconcept of algorithms and programming in the K-12 CS framework. By introducing students to refactoring and iterative development, this tutorial also aligns with the program development subconcept. To solidify students’ understanding of this essential CS/CT concept, the in-app assessment asks questions about the definition of variables and the use of variables in sample code.

Learning activity 4: Control flow

The fourth learning activity introduces students to the importance of sequence and control flow in computing and connects this concept to sequence and logical flow in reading and writing. Again, the activity introduces basics through simple videos and text, then provides a MakeCode walk-through to develop a step counter that students can control with an on-off button. By

introducing the *if* programming construct, this learning activity covers the control subconcept of the algorithms and programming concept in the K-12 CS framework. After building this new step counter, students are engaged in a combined experimentation and PA lesson in which they collect data to compare their micro:bit’s step count to a ground truth they count themselves. They collect these data when the micro:bit step counter is in their hand, in their pocket, and in their sock or shoe. We then define accuracy for the students (as “how well a measurement matches the real value”) and ask them to determine which placement results in the most accurate count. This experimentation connects to state learning standards in both science and math. Finally, we close the activity with the assessment in Figure 3, which focuses on fundamentals of control flow, with a direct connection to sequence in reading and writing.

Learning activity 5: Rate

The fifth learning activity introduces rate as a measurement of something per unit of something else. This activity focuses on step rate, or the number of steps per unit of time. We start with the concept of rate, independent of CS/CT. We walk students through some math problems to compute step rates and practice comparing them (for example, “You walked 120 steps in a minute. Your friend also walked 120 steps, but took an hour. Who has the higher step rate? Who was more active?”). After these examples, students visit MakeCode to create the most complex program yet: one that calculates and displays their step rate by dividing the number of steps by the time elapsed since a button press. A snapshot of a midway point in this

tutorial is presented in Figure 4; from the figure, it is easy to see the growing sophistication of the students' programming skills relative to the early program in Figure 2. This activity focuses on the K-12 CS subconcepts of visualization and transformation (a subconcept of data and analysis) and program development (from algorithms and programming).

Learning activity 6: Complex conditionals

The sixth learning activity focuses on complex conditionals (for example, adding *else* to the *if* from the fourth activity). The activity starts with a physical education lesson about rate and exercise intensity. Students are reminded how their

bodies provide indications of PA intensity (for instance, how hard they breathe, how fast their heart beats, and so on) and that their step rate is yet another measure of intensity. They are guided through some PA that uses their micro:bit step rate counter to connect their step rate to these other feelings of intensity. With this knowledge, students undertake a MakeCode tutorial in which they calibrate their feelings of step rate and intensity to moveSMART activity levels. At the end of this lesson, rather than displaying their step rate, the micro:bit prints out *red*, *yellow*, or *green*. Within the K-12 CS framework, the focus is primarily on the control subconcept of algorithms and programming.

Learning activity 7: Communication

In the final learning activity, students get to change the moveSMART game itself. Rather than logging their activity with an RFID card or using the web-based check-in, the students use a communication link to send their activity level from the micro:bit to the Raspberry Pi in their class's physical check-in box. The learning activity starts via a simple lesson about networks and packets and how devices communicate information. A MakeCode tutorial walks students through creating a simple "packet" that contains their activity level (red, yellow, or green) and some information that identifies them (such as their student number). The students use the MakeCode radio to send the packet to another

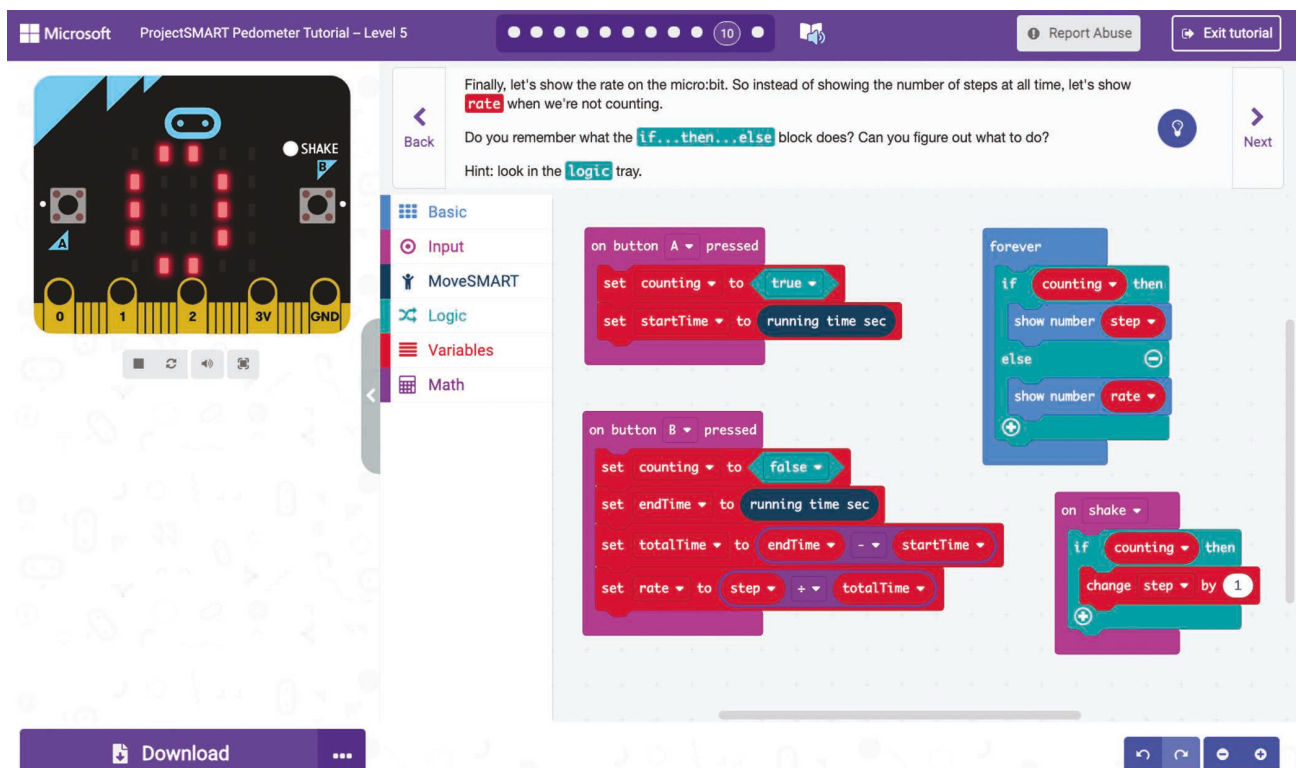


FIGURE 4. The fifth CS/CT learning activity in Project moveSMART.

micro:bit that is connected to the Raspberry Pi inside the checkin box. This activity is connected to the network communication and organization sub-concept of the networks and the internet in the K-12 CS framework.

PA AND CS INTEREST

In the final week of the 2020-2021 academic year, we added the first five micro:bit tutorials to our active moveSMART deployment at HDES and made them available to two fourth-grade classes and the entire fifth grade. We joined the classes in person and guided them through the learning activities. Students worked on the micro:bit tutorials in pairs during a 50-min class period. While progressing through the

tutorials, they could ask teachers and the other RPP members in attendance for assistance. We worked with the two fourth-grade classes on the first day. Because of COVID-19, only nine fourth-grade students were in physical attendance. One member of the research team engaged the virtually connected students via the remote learning platform, but those students did not complete the activities with the micro:bit.

After the visit to the fourth-grade generated excitement in the school, we worked with the entire fifth grade on the second day. The fourth graders had been engaging with the moveSMART platform throughout the school year, so they could easily log in and navigate the website. The fifth-grade

students had no previous exposure to the moveSMART platform. As a result, most of the fourth-grade students completed the first two CS/CT learning activities. In contrast, most, but not all, of the fifth-grade students completed the first CS/CT learning activity. None of them completed the second one.

Based on these interactions and our experiences engaging these students with moveSMART throughout the school year, we made the following observations: 1) even a short intervention using the micro:bit-based learning activities has the potential to improve students' coding attitudes and 2) incremental deployment of features helped maintain engagement. With respect to the first observation, we delivered "Elementary Students Attitudes Towards Coding"²⁰ as a pre-test and posttest. Students completed the measure the day before the CS/CT learning activities and again at the end of the 50-min class period. The attitude measures were delivered through waypoints in the Project moveSMART map. The measure has five constructs: coding confidence, interest, utility, social value, and perception of coders. The results for both grades are given in Figure 5. After engaging with the micro:bit tutorials, fourth-grade students showed significant increases in coding confidence (p value = 0.0023; n = 7) and perception of coders (p value = 0.0231; n = 7).

There were also improvements in fourth-grade students' coding interest, attitudes toward coding utility, and perceptions of the social value of coding. However, these improvements were not statistically significant. Because the micro:bit tutorials also include PA components and concepts that align with state learning standards, they could be easily integrated into teachers'

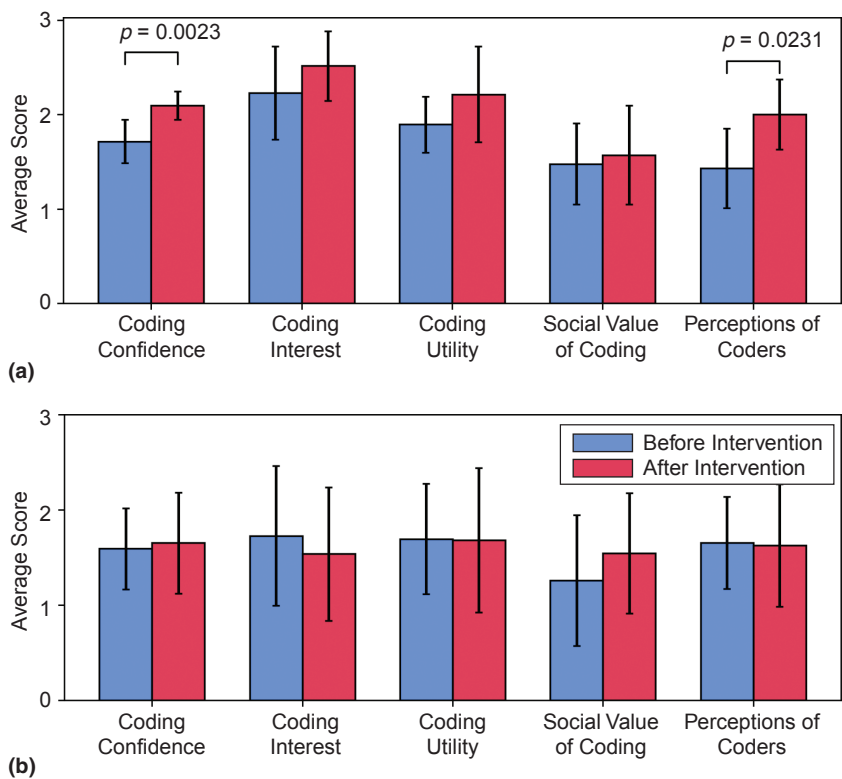


FIGURE 5. The average coding attitude survey responses for (a) fourth-grade and (b) fifth-grade students before and after completing the first five micro:bit tutorials.

ABOUT THE AUTHORS

CONNOR FRITZ is with the Mobile and Pervasive Computing Laboratory, University of Texas at Austin, Austin, Texas, 78712, USA. His research interests include digital health and the use of software for behavior change. Fritz received an M.S. in electrical and computer engineering from the University of Texas at Austin. Contact him at cdfritz7@gmail.com.

DYLAN BRAY is a former member of the Mobile and Pervasive Computing Laboratory, University of Texas at Austin, Austin, Texas, 78712, USA. His research interests include human–software interaction and applying software for digital health in traditionally underserved communities. Bray received an M.S. in electrical and computer engineering from the University of Texas at Austin. Contact him at dbray@utexas.edu.

GRACE LEE is with the Mobile and Pervasive Computing Laboratory, University of Texas at Austin, Austin, Texas, 78712, USA. Her research interests include software for serious games that aim at behavioral health change. Lee received an M.S. in communications engineering from National Chiao Tung University. Contact her at gracewlee@utexas.edu.

CHRISTINE JULIEN is a full professor at the University of Texas at Austin, Austin, Texas, 78712, USA. Her research interests include mobile and pervasive computing, middleware, and context-aware computing. Julien received a D.Sc. in computer science from Washington University, Saint Louis. She is a Senior Member of IEEE. Contact her at c.julien@utexas.edu.

SHERI BURSON is a health education and health behavior doctoral candidate at the University of Texas at Austin,

Austin, Texas, 78712, USA. Her research interests include PA and play in elementary school and how they impact academic achievement and social–emotional well-being. Burson received a Ph.D. in health education from the University of Texas at Austin. Contact her at slb4975@utexas.edu.

DARLA CASTELLI is a full professor in the Department of Kinesiology and Health Education, University of Texas at Austin, Austin, Texas, 78712, USA. Her research interests include the effects of PA on cognitive function among children in schools and natural settings. Castelli received a Ph.D. in health education from the University of Texas at Austin. Contact her at dcastelli@utexas.edu.

CAROL RAMSEY is a high-school computer science teacher and the owner of CS ED Group. She is also an assistant professor of instruction at the University of Texas at Austin, Austin, Texas, 78712, USA. Her research interests include equity-focused K–12 computer science curriculum and pathways. Ramsey received an M.S. in science, technology, engineering, and mathematics from the University of Texas at Austin. Contact her at ramsey@csedgroup.com.

JAMIE PAYTON is a professor and the chair of computer and information sciences at Temple University, Philadelphia, Pennsylvania, 19122, USA. Her research interests include evidence-based approaches for equitable and inclusive computer science education, broadening participation of historically excluded groups in computing, and pervasive computing for smart health, safety, and well-being. Payton received a Ph.D. in computer science and engineering from Washington University, Saint Louis. She is a Member of IEEE. Contact her at payton@temple.edu.

curricula. There were no statistically significant changes for the fifth-grade students' coding attitudes, but a large portion of the fifth-grade class period was spent introducing moveSMART, so

many students did not make significant progress through the learning activities. Because the fourth-grade students had been more engaged with the platform throughout the year, they were

able to make greater progress because they had less trouble logging into and navigating the platform. This highlights the importance of incrementally introducing platform features.

Importantly, we also received feedback from the teachers. One teacher (a physical education teacher) told us, “Initially, I thought, computer science in elementary school, it doesn’t matter. After watching [the students] doing it, I was fascinated with how much they loved this activity. They initially didn’t think they were capable of doing it. They had so much fun. This opened their minds to doing computer science, and they really believed in themselves.”

We have described a set of CS/CT learning activities centered around the micro:bit and deployed through the Project moveSMART platform. These activities teach students CS/CT concepts as they build a device to measure their PA. By creating and using a physical computing solution, students gain a better understanding of how CS/CT can be applied in the real world. In a pilot study, we found that fourth-grade students at our partner school had an improved confidence in their ability to code and in their perception of coders after a 50-min intervention. This pilot study suggests that micro:bit learning activities that integrate PA may be useful for engaging students from populations that have been historically excluded in computing. We are now performing a study with two fourth-grade and two fifth-grade classes (consisting of more than 100 students) that are engaging with moveSMART throughout the school year. 

ACKNOWLEDGMENT

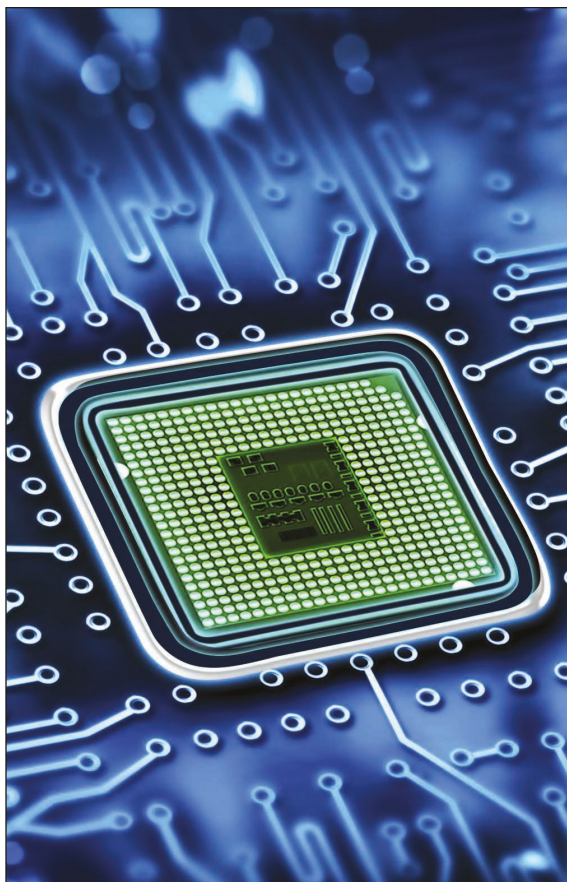
The authors would like to thank the teachers and administrators who are part of the RPP. Without their patience and dedication, none of the work in this article would be possible. This work was funded, in part, by the National Science

Foundation, under grants CNS-2031498 and CNS-2031324, and by Whole Communities-Whole Health, a research grand challenge at the University of Texas at Austin. Any opinions, findings, conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the sponsors. This work involved human subjects or animals in its research. Approval of all ethical and experimental procedures and protocols was granted by the University of Texas at Austin Institutional Review Board under Application No. STUDY00000605.

REFERENCES

1. C. Cadenas-Sanchez *et al.*, “Fitness, physical activity and academic achievement in overweight/obese children,” *J. Sports Sci.*, vol. 38, no. 7, pp. 731–740, 2020, doi: 10.1080/02640414.2020.1729516.
2. A. Yadav, S. Gretter, J. Good, and T. McLean, “Computational thinking in teacher education,” in *Emerging Research, Practice, and Policy on Computational Thinking*, P. Rich, and C. Hodges, Eds. Cham: Springer-Verlag, 2017, pp. 205–220.
3. E. Hunsaker, “Computational thinking,” in *The K-12 Educational Technology Handbook*, A. Ottenbreit-Leftwich and R. Kimmons, Eds. EdTech Books, 2020, pp. 1–413.
4. C. L. Fletcher and J. R. Warner, “CAPE: A framework for assessing equity throughout the computer science education ecosystem,” *Commun. ACM*, vol. 64, no. 2, pp. 23–25, 2021, doi: 10.1145/3442373.
5. J. Wang and S.H. Moghadam, “Diversity barriers in K-12 computer science education: Structural and social,” in *Proc. 2017 ACM SIGCSE Tech. Symp. Comput. Sci. Educ.*, pp. 615–620, doi: 10.1145/3017680.3017734.
6. A. Lukács, P. Sasvári, and E. Kiss-Tóth, “Physical activity and physical fitness as protective factors of adolescent health,” *Int. J. Adolescent Med. Health*, vol. 32, no. 6, Aug. 2018, doi: 10.1515/ijamh-2018-0017.
7. D. P. Coe, T. Peterson, C. Blair, M. Schutten, and H. Peddie, “Physical fitness, academic achievement, and socioeconomic status in school-aged youth,” *J. School Health*, vol. 83, no. 7, pp. 500–507, Jul. 2013, doi: 10.1111/josh.12058.
8. R. Guthold, G. A. Stevens, L. M. Riley, and F. C. Bull, “Global trends in insufficient physical activity among adolescents: A pooled analysis of 298 population-based surveys with 1·6 million participants,” *Lancet Child Adolescent Health*, vol. 4, no. 1, pp. 23–25, Jan. 2020, doi: 10.1016/S2352-4642(19)30323-2.
9. T. H. Fakhouri, J. P. Hughes, D. J. Brody, B. K. Kit, and C. L. Ogden, “Physical activity and screen-time viewing among elementary school-aged children in the United States from 2009 to 2010,” *JAMA Pediatrics*, vol. 167, no. 3, pp. 223–229, 2013, doi: 10.1001/2013.jamapediatrics.122.
10. H. W. Kohl and H. D. Cook, *Educating the Student Body: Taking Physical Activity and Physical Education to School*. Washington, DC, USA: The National Academies Press, 2013.
11. M. A. Farooq *et al.*, “Timing of the decline in physical activity in childhood and adolescence: Gateshead Millennium Cohort Study,” *Brit. J. Sports Med.*, vol. 52, no. 15, pp. 1002–1006, Aug. 2018, doi: 10.1136/bjsports-2016-096933.
12. E. Farella *et al.*, “CLIMB: A pervasive gameful platform promoting child independent mobility,” *IEEE Pervasive Comput.*, vol. 19, no.

- 1, pp. 32–42, 2020, doi: 10.1109/MPRV.2019.2939730.
13. S. Hodges, S. Sentance, J. Finney, and T. Ball, “Physical computing: A key element of modern computer science education,” *Computer*, vol. 53, no. 4, pp. 20–30, Apr. 2020, doi: 10.1109/MC.2019.2935058.
14. J. Austin *et al.*, “The BBC micro:bit from the U.K. to the world,” *Commun. ACM*, vol. 63, no. 3, pp. 62–69, 2020, doi: 10.1145/3368856.
15. “Accountability rating systems,” Texas Education Agency, Austin, TX, USA. Accessed: Oct. 26, 2021. [Online]. Available: <https://rptsvr1.tea.texas.gov/perfreport/account/index.html>
16. D. G. Kelley and P. Seeling, “Introducing underrepresented high school students to software engineering: Using the micro:bit micro-controller to program connected autonomous cars,” *Comput. Appl. Eng. Educ.*, vol. 28, no. 3, pp. 737–747, Apr. 2020, doi: 10.1002/cae.22244.
17. M.C. Parker and L.A. DeLyser, “Concepts and practices: Designing and developing a modern K-12 CS framework,” in *Proc. ACM SIGCSE Tech. Symp. Comput. Sci. Educ.*, Mar. 2017, pp. 453–458, doi: 10.1145/3017680.3017778.
18. T. Ball, A. Chatra, P. de Halleux, S. Hodges, M. Moskal, and J. Russell, “Microsoft MakeCode: Embedded programming for education, in blocks and TypeScript,” in *Proc. 2019 ACM SIGPLAN Symp. SPLASH-E*, pp. 7–12, doi: 10.1145/3358711.3361630.
19. R. M. Reitan, “Trail making test: Manual for administration and scoring,” Reitan Neuropsychology Lab., Tuscon, AZ, USA, 1992.
20. S. L. Mason and P. J. Rich, “Development and analysis of the elementary student coding attitudes survey,” *Comput. Educ.*, vol. 153, p. 103,898, Aug. 2020, doi: 10.1016/j.compedu.2020.103898.



IEEE TRANSACTIONS ON

COMPUTERS

Call for Papers: *IEEE Transactions on Computers*

Publish your work in the IEEE Computer Society's flagship journal, *IEEE Transactions on Computers*. The journal seeks papers on everything from computer architecture and software systems to machine learning and quantum computing.

Learn about calls for papers
and submission details at
www.computer.org/tc.

