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

Connor Fritz, Dylan Bray, Grace Lee, and Christine Julien
Electrical and Computer Engineering, The University of Texas at Austin

Sheri Burson and Darla Castelli
Health Education and Kinesiology, The University of Texas at Austin

Carol Ramsey
CS Ed Group

Jamie Payton
Computer Science, Temple University

***Abstract*—Although computing skills are increasingly required for success in high school, college, and beyond, there is little emphasis on improving computational thinking in elementary curricula. Computer science pathways that do exist often fail to engage student populations that are traditionally underserved. Project moveSMART uses a web-based platform to integrate opportunities for physical education with computer science and computational thinking (CS/CT) learning activities. Project moveSMART was developed through a researcher-practicioner partnership involving computer scientists, educational researchers, and teachers. This article describes a series of tutorials from Project moveSMART designed to introduce elementary students to CS/CT by making connections to physical activity and grade-level curricula in other subjects. Through these tutorials, students create a physical activity monitor using the BBC micro:bit. Fourth grade students that underwent a single day intervention experienced a significant improvement in their interest in coding and in their perceptions of coders.**

Index Terms: computer science education, serious games, pervasive computing

■ **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 and computational thinking (CS/CT) into K-12 curricula have only recently been established, and many current 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].

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 of 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 computer science and parents' and educators' beliefs that underrepresented groups are not as interested in pursuing computing [5]. Students from low-income families face structural barriers (e.g., lack of home computers, lack of computer science courses in their schools, and lack of extracurricular CS/CT opportunities) that limit access and exposure to computer science learning. Despite the modern relevance of computing, state learning standards for elementary students rarely include CS/CT topics. While teachers are often 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 to adhere to a provided curriculum.

Physical education is also increasingly neglected in elementary school despite its many demonstrated benefits. For children, physical activity is a predictor of adolescent health [6] and success in school [7]. Despite the benefits of physical activity (PA), 80% of adolescents fail to meet the recommended hour of daily PA [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 minutes per 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 age 9 [11]. This makes elementary school a prime candidate for interventions to increase student PA.

Although teachers may recognize the importance of computer science and physical education, they also need to focus on delivering content aligned with state learning standards, which often do not involve PA or 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. Project moveSMART promotes increased PA and CS/CT while also 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 physical activity, 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's score by logging higher rates of physical activity.

This paper details a Project moveSMART pilot study in which elementary school students completed a set of tutorials that combine PA, CS/CT concepts, and content aligning with Texas state learning standards. We found that 4^{th} 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 physical activity 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 does not directly integrate physical activity with the learning activities delivered through its online platform.

A relatively new area in computer science education is physical computing, which involves using software and hardware to build physical systems and to teach CS/CT concepts [13]. Approaches that utilize physical computing often

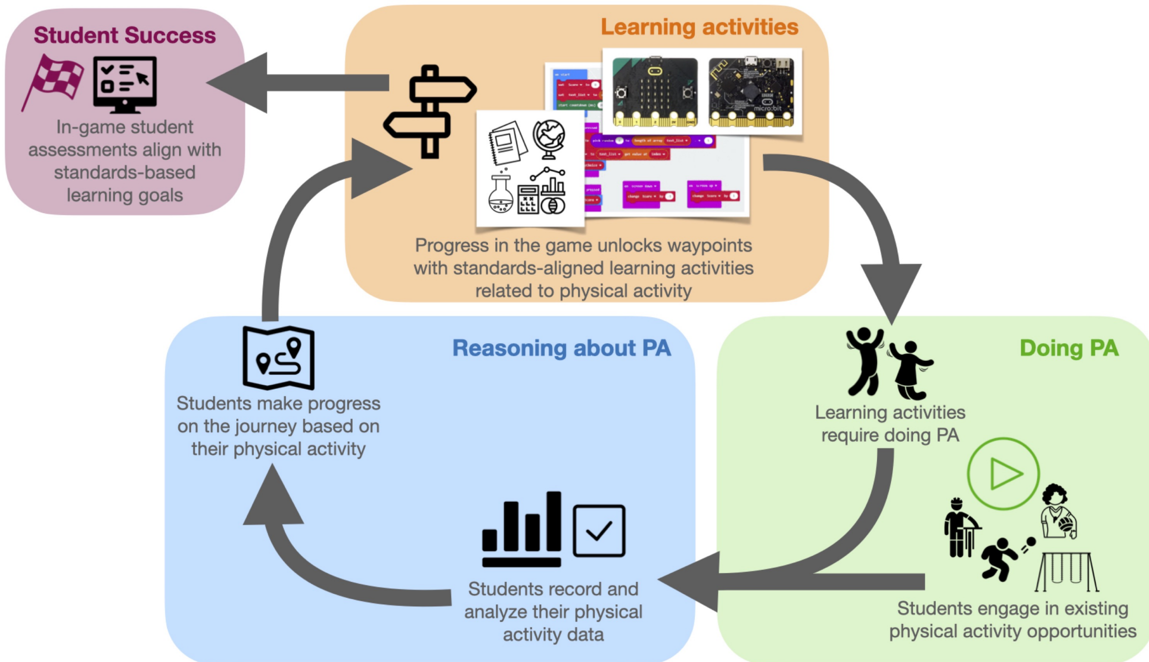

Figure 1. moveSMART. Starting from the bottom right, students engage in in-school physical activity. They record their data in moveSMART, which progresses the class along the journey. Progress unlocks waypoints, which contain learning activities across the curriculum. Learning activities (1) generate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

use embedded microcomputers such as the BBC micro:bit [14] that are meant to be applied in educational contexts. While Project moveSMART utilizes physical computing in the CS/CT learning activities discussed in the next 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 moveSMART online platform. This integration allows 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 addressed these issues simultaneously.

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 1 school; 86.5% of students qualify for free or reduced lunch. The 4^{th} and 5^{th} grade teachers, as well as the school's physical education teacher and principal are part of the moveSMART researcher practitioner partnership (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 (e.g., the 4^{th} grade route crosses Texas, while the 5^{th} grade crosses the U.S.) when students participate in PA opportunities offered within the school day (e.g., in physical education class or at recess). Students log their PA by choosing one of "Red", "Yellow", or "Green" levels with "Red" indicating a low activity level and "Green" indicating a high activity level. Students can log PA through a web-based check-in system, or through a physical check-in box. The box

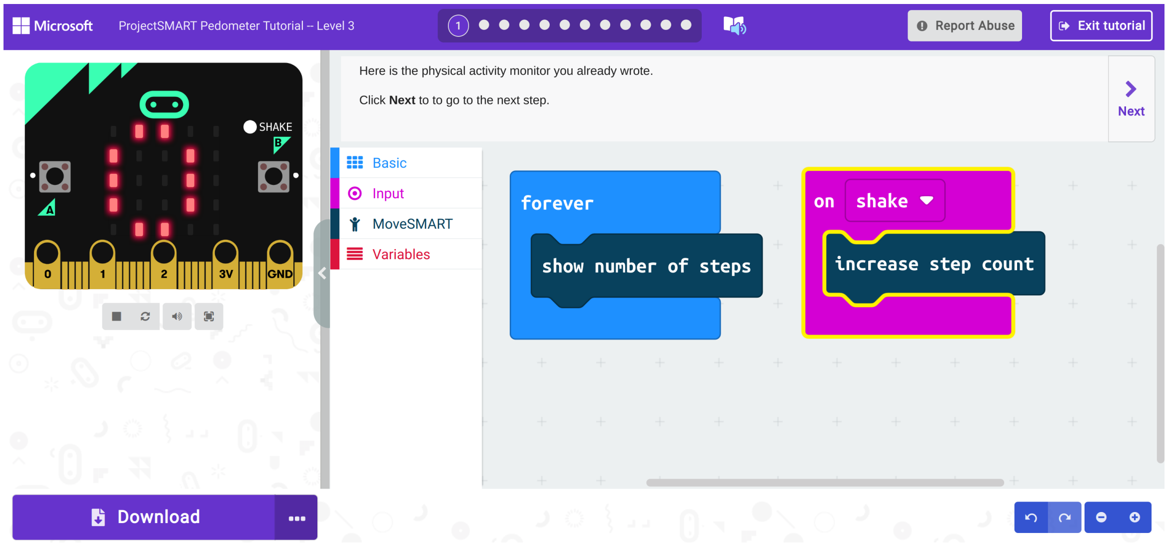

Figure 2. The second CS/CT learning activity in moveSMART, delivered through the MakeCode tutorial platform

consists of a Raspberry Pi connected to an RFID card reader. Students scan an assigned RFID card, then choose their activity level by pushing a colored button. The button push triggers an API 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 further 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 embedded content, assessments, and CS/CT learning activities. This paper focuses on the CS/CT activities in the waypoints; below, 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 allowing students to program real-world devices, physical computing platforms concretely demonstrate the value of programming to students. Additionally, students from groups that are traditionally underrepresented in computer science 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 Computer Science Framework [17], a set of guidelines used to develop computer science educational standards and curricula. The K-12 CS Framework consists of both *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 concepts: *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 below, 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 shows an in-

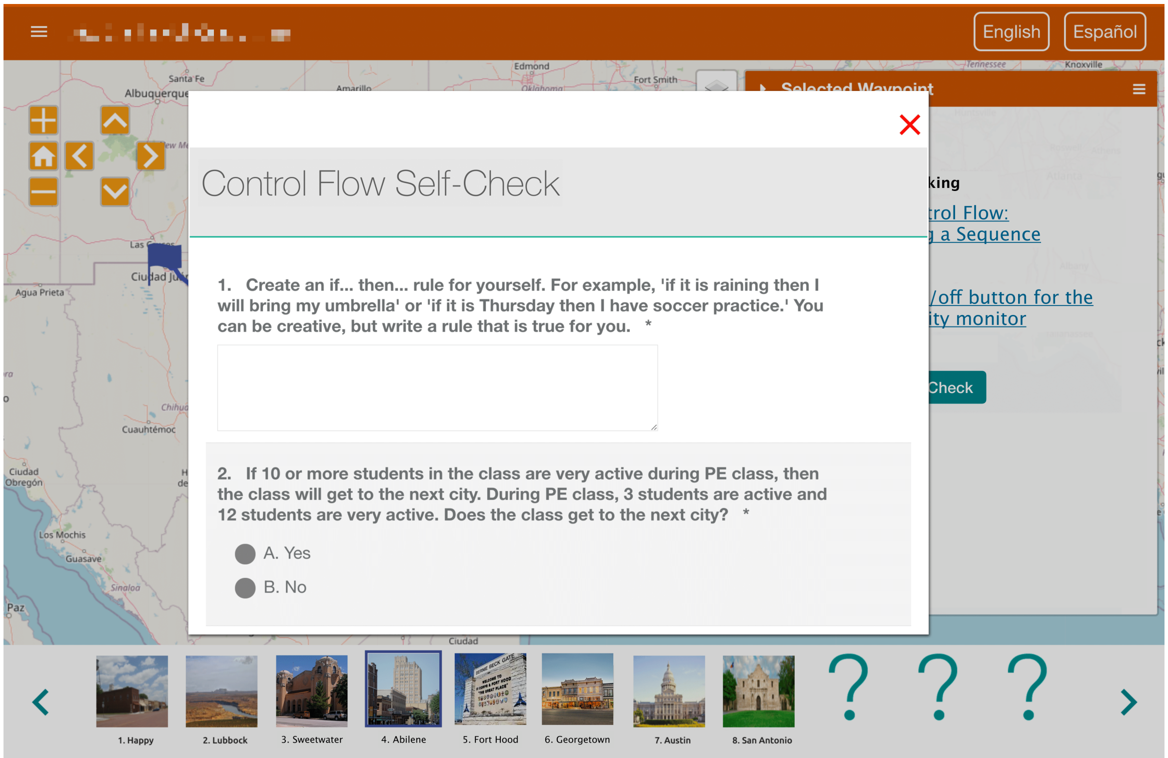

Figure 3. An assessment embedded into the moveSMART platform

termediate step of the second activity, which students undertake after learning about accelerometers. MakeCode provides a playground in which the students can experiment. We developed a set of tutorials for MakeCode, along with some moveSMART programming abstractions that allow 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 abstraction. At the end of each walk-through, students download their completed programs onto physical micro:bits to see them in action, to use them for other classroom activities, or to complete physical activity 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 shows 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 learning activities in HDES during the 2020-2021 academic year.[1]

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

[1] Because of significant changes to elementary instruction in 2020-2021 due to the COVID-19 pandemic, most of our interactions with the school were via virtual channels. However, in the last week of school, we did have one class period each with the 4^{th} and 5^{th} grade, where we piloted the CS/CT learning activities.

long it takes each of them to complete a *Trail Making Test* [19], a measure of cognitive flexibility. Upon completing this activity using pencil and paper, students return to moveSMART to complete an assessment. The assessment for this first 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 second = 1000 milliseconds. If the trail making task took your friend Robert 23 seconds, how many milliseconds did it take?* and (2) *If you took 22,923 milliseconds to complete the trail making task and your friend Robert took 23 seconds, which one of you completed it faster?*. These questions are aligned with the state-level mathematics standards for 4^{th} and 5^{th} grade 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 with the *Collection* subconcept of *Data and Analysis*. We start with a physical activity that has the student intentionally move along the three axes of acceleration (i.e., *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 three axes within the micro:bit and explore, physically, how this relates to their real device. The students then use a MakeCode walkthrough 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 (e.g., by sticking them in their pocket or sock) and are guided through a physical activity 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 physical activity in terms of an inequality (e.g., *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. The activity explains variables using 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 walkthrough 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 physical activity lesson in which they collect data to compare their micro:bit's step count to a ground truth they count themselves. They collect this data as the micro:bit step counter is held in their hand, placed in their pocket and when in their sock or shoe. We then define accuracy for the students (as "how well a measurement matches the real value") and ask the students to determine which placement results in the most

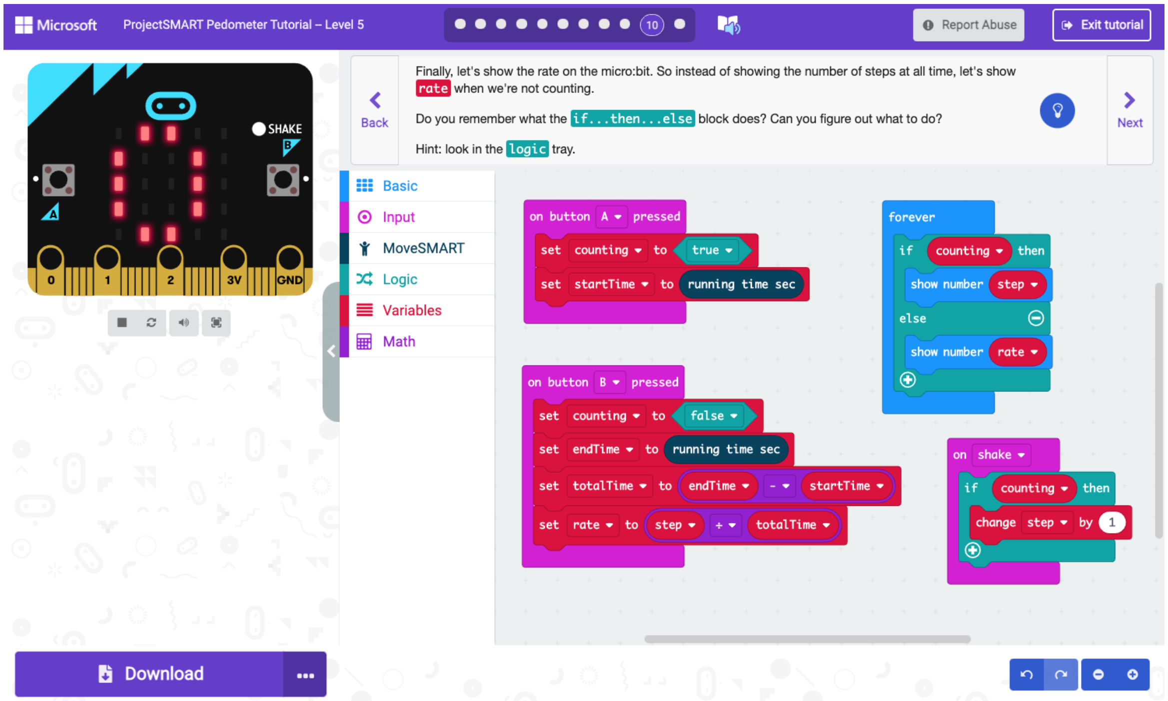

Figure 4. The fifth CS/CT learning activity in Project moveSMART.

accurate count. This experimentation connects to state learning standards in both science and math. Finally, we close the activity with the assessment shown 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 to practice comparing them (e.g., *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 shown in Figure 4; from the figure it is easy to see the growing sophistication of the students' programming skills relative to the early program shown 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 (e.g., adding **else** to the *if* from the fourth activity). The activity starts with a P.E. lesson about rate and physical activity intensity. Students are reminded how their bodies provide indications of physical activity intensity (e.g. how hard they breathe, how fast their heart beats, etc.) and that their step rate is yet another measure of intensity. They are guided through some physical activity 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 using 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 (e.g., their student number). The students use the MakeCode `radio` to send the packet to another micro:bit that is connected to the Raspberry Pi inside of the checkin box. This activity is connected to the *Network Communication and Organization* subconcept of the *Networks and the Internet* in the K-12 CS framework.

PHYSICAL ACTIVITY 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 Hornsby Dunlap Elementary School and made them available to two 4^{th} grade classes and the entire 5^{th} 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 minute class period. While progressing through the tutorials, students could ask teachers and the other RPP members in attendance for assistance. We worked with the two 4^{th} grade classes in person on the first day. Because of the COVID-19 pandemic, only 9 4^{th} grade students were in attendance in person. One member of the research team engaged the virtually connected students via the remote learning platform, but they did not complete the activities with the micro:bit. After the visit to the 4^{th} grade generated excitement in the school, we worked with the entire 5^{th} grade on the second day. The 4^{th} graders had been engaging with the moveSMART platform throughout the school year, so they could easily login and navigate through the website. The 5^{th} grade students had no previous exposure to the moveSMART platform. As a result, most of the 4^{th} grade students completed the first two CS/CT learning activities. In contrast, most, but not all, of the 5^{th} grade students completed the first CS/CT learning activity. None of them completed the second one.

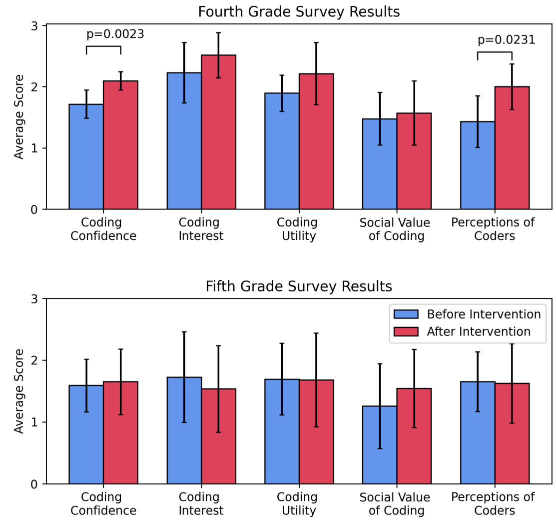

Figure 5. Average Coding Attitude Survey responses for 4^{th} and 5^{th} grade students before and after completing the first five micro:bit tutorials.

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 the Elementary Students Attitudes Towards Coding [20] measure as a pre-test and as a post-test. Students completed the measure the day before the CS/CT learning activities and then again at the end of the 50 minute 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 shown in Figure 5. After engaging with the micro:bit tutorials, 4^{th} 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 4^{th} grade students' coding interest, attitudes towards coding utility, and perceptions of the social value of coding. However, these improvements were not statistically significant. Because the micro:bit tutorials also include physical activity components and concepts that align with state learning standards, they could be easily integrated into teachers' curricula. There were no statistically significant changes for the 5^{th} grade students' coding attitudes, but a large portion of the 5^{th} grade class period was spent introducing moveSMART, so many students did not make significant progress through the learning activities. Because the 4^{th} 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 through 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."

CONCLUSION

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 physical activity. 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 4^{th} grade students at our partner school had an improved confidence in their ability to code and in their perception of coders after a 50 minute intervention. This pilot study suggests that micro:bit learning activities that integrate physical activity may be useful for engaging students from populations that have been historically excluded in computing. We are now performing a study with two 4^{th} grade and two 5^{th} grade classes (consisting of over 100 students, total) engaging with moveSMART across 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 paper 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.

REFERENCES

1. C. Cadenas-Sanchez, J.H. Migueles, I. Esteban-Cornejo, J. Mora-Gonzalez, P. Henriksson, M. Rodriguez-Ayllon, P. Molina-García, M. Löf, I. Labayen, C.H. Hillman, A. Catena, and F.B. Ortega, "Fitness, physical activity and academic achievement in overweight/obese children," *Journal of Sports Sciences*, 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*, pp. 205–220, 2017, doi: 10.1007/978-3-319-52691-1_13.
3. E. Hunsaker, "Computational thinking," *The K-12 educational technology handbook* 2020.
4. C.L. Fletcher, and J.R. Warner, "CAPE: A framework for assessing equity throughout the computer science education ecosystem," *Communications of the 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 *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education*," 2017, pp. 615–620.
6. A. Lukács, P. Sasvári, E. Kiss-Tóth, "Physical activity and physical fitness as protective factors of adolescent health," *International Journal of Adolescent Medicine and Health*, vol. 32, no. 6, August 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," *Journal of School Health*, vol. 83, no. 7, pp. 500-507, July 2013, doi: 10.1111/josh.12058.
8. R. Guthold, G.A. Stevens, L.M. Riley, F.C. Bull, "Global trends in insufficient physical activity among adolescents: a pooled analysis of 298 population-based surveys with

1·6 million participants," *The Lancet Child & Adolescent Health*, vol. 4, no. 1, pp. 23-25, January 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 III and H.D. Cook, "Educating the student body: Taking physical activity and physical education to school," 2013.
11. M.A. Farooq, K.N. Parkinson, A.J. Adamson, M.S. Pearce, J.K. Reilly, A.R. Hughes, X. Janssen, L. Basterfield, J.J. Reilly, "Timing of the decline in physical activity in childhood and adolescence: Gateshead Millennium Cohort Study," *British Journal of Sports Medicine*, vol. 52, no. 15, pp. 1002-1006, August 2018, doi: 10.1136/bjsports-2016-096933.
12. E. Farella, M. Ferron, D. Giovanelli, C. Leonardi, A. Marconi, P. Massa, A.L. Murphy, M. Nori, M. Pistore, G. nd Schiavo, "CLIMB: A pervasive gameful platform promoting child independent mobility," *IEEE Pervasive Computing*, 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, April 2020, doi: 10.1109/MC.2019.2935058.
14. J. Austin, H. Baker, T. Ball, J. Devine, J. Finney, P. De Halleux, S. Hodges, M. Moskal, and G. Stockdale, "The BBC micro:bit: from the U.K. to the world" *Communications of the ACM.*, vol. 63, no. 3, pp. 62–69, 2020, doi: 10.1145/3368856.
15. Texas Education Agency. "Accountability Rating Systems." Texas Education Agency. https://rptsvr1.tea.texas.gov/perfreport/account/index.html (accessed Oct. 26, 2021)
16. D. G. Kelley and P. Seeling, "Introducing underrepresented high school students to software engineering: using the micro:bit microcontroller to program connected autonomous cars," *Computer Applications in Engineering Education*, vol. 28, no. 3, pp. 737-747, April 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 *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education*, pp. 453-458, March 2017, 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 *Proceedings of the 2019 ACM SIGPLAN Symposium on SPLASH-E*, pp. 7–12, 2019, doi: 10.1145/3358711.3361630.
19. R. M. Reitan, *Trail Making Test: Manual for administration and scoring.* Tuscon, AZ, USA: Reitan Neuropsychology Laboratory, 1992.
20. S.L. Mason and P.J. Rich, "Development and analysis of the elementary student coding attitudes survey," *Computers & Education*, vol. 153, August 2020, doi: 10.1016/j.compedu.2020.103898.

Connor Fritz is with the Mobile and Pervasive Computing Laboratory at the University of Texas at Austin, USA. He is currently pursuing the M.S.E degree at the University of Texas at Austin. Connor is interested in digital health and the use of software for behavior change. Contact him at cdfritz7@gmail.com.

Dylan Bray is a former member of the Mobile and Pervasive Computing Laboratory. He graduated with an M.S.E degree from the University of Texas at Austin in 2020. Dylan is interested in human-software interaction and applying software for digital health in traditionally underserved communities. Contact him at dfbray@utexas.edu.

Grace Lee is a member of the Mobile and Pervasive Computing Laboratory. She is pursuing a PhD degree at the University of Texas at Austin, USA. She is developing a software framework for serious games that aim at health behavioral change. Contact her at gracewlee@utexas.edu.

Christine Julien (IEEE Senior Member) is a full professor at the University of Texas at Austin, USA. She received the D.Sc. in Computer Science from Washington University in Saint Louis in 2004. Her research interests include mobile and pervasive computing, middleware, and context-aware computing. 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 and a member of the Kinetic Kidz Lab. Sheri is interested in physical activity and play in elementary school and how they impact academic achievement and social-emotional well-being. Contact her at slb4975@utexas.edu

Darla Castelli is a full professor in Department of Kinesiology and Health Education at The University

of Texas at Austin. Her research examines the effects of physical activity on cognitive function among children in schools and natural settings. Her work is predicated on her experiences as a former public school teacher and administrator. Contact her at dcastelli@utexas.edu.

Carol Ramsey is a high school computer science teacher with IDEA Public Schools and owner of CS ED Group, a computer science education consulting company. She earned her STEM Education master's and Computer Science bachelor's degrees from The University of Texas at Austin. Her work focuses on equity-focused K12 computer science curriculum and pathways. Contact her at ramsey@csedgroup.com.

Jamie Payton (IEEE Member) is a Professor and Chairperson of Computer and Information Sciences at Temple University. She received her doctoral degree in Computer Science and Engineering from Washington University in Saint Louis in 2006. 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. Contact her at payton@temple.edu.