Engineering Educators Bringing the World Together
2025 ASEE Annual Conference & Exposition
Palais des congrès de Montréal, Montréal, QC • June 22–25, 2025
Paper ID #49341

Co-Designing Integrated CS Curriculum Artifacts with K-5 Classroom Teachers

Maverick Berner, Marquette University
Dr. Darcy Ronan, Sacred Heart University
Dr. Dennis Brylow, Marquette University
Maximus Berner, Marquette University

Co-Designing Integrated CS Curriculum Artifacts with K-5 Classroom Teachers

1 Abstract

Recognizing the critical role of early education in developing students' technological literacy, this study investigates the effectiveness of a multifaceted support program designed to empower teachers with the skills, resources, and pedagogical strategies necessary to successfully integrate computer science (CS) into other core subject areas. In this experience report, we detail our observations while working with K-5 teachers across Wisconsin and Connecticut to design novel digital tools and software to support CS integration in their classrooms. The initiative aims to address major barriers to CS education access in high-need school districts while simultaneously enriching learning outcomes in the subject area targeted for CS integration. To achieve this we center teachers at the heart of the development process, driving design decisions to build a custom solution that will most effectively engage their particular student populations.

2 Introduction and Motivation

As technological innovations continue to advance, computer science is increasingly considered an essential discipline for both our future workforce and engaged citizenry. Consequently, there is a growing national recognition of the imperative to improve CS education in K-12 to meet public demand, embraced by the US Department of Education, researchers and professionals alike [1]. Although substantial progress remains to be made, concentrated efforts to improve CS education in the past decade have shown promising results. The percentage of public high schools in Wisconsin that offer foundational CS has increased from merely 34% during AY18 (the academic year 2017-2018) to 56% in AY22 [2]. Likewise, Connecticut has also seen an increase from 67% in AY19 to 84% in AY22 [2]. Although it is encouraging to see an increase in the adoption of CS, this binary (offers / does not offer) metric does not capture whether the quantity and quality of the CS courses offered are satisfactory. Neither does it address the large disparities in gender, race, and socioeconomic status among students who enroll in CS.

In this paper, we will detail our experience working with K-5 teachers to co-develop novel CS-embedded curriculum and accompanying digital artifacts. These curriculum modules integrate CS content with other core subject areas taught in their classrooms and may be supplemented by a mix of both plugged (on the computer) and unplugged (no digital devices required) lesson components. Although many of the teacher participants in the project were able to realize their new curriculum modules using existing software or classroom technology, here we focus on those teachers whose designs required a *novel digital artifact*, such as a new web application with specific functionality, customized content to work with an existing tool, or a new piece of standalone software. By providing the developer support to produce these digital artifacts, we empower teacher participants to embed CS content into potentially challenging classroom contexts that are resistant to drop-in, ”one-size-fits-all” integration solutions. Using this approach, teachers can create more thoughtful and robust CS curriculum modules that better complement their particular needs.

This initiative promotes equitable access to CS education by addressing three major barriers: the digital divide between students in higher-resource and lower-resource districts, the lack of CS education materials featuring consciously inclusive design, and the low confidence or preparedness among K-5 teachers regarding CS education. We address aspects of the digital divide through attentive co-design, ensuring that the novel digital artifacts work with readily available classroom resources. Some districts have access to various robotics kits and supply all students with personal computers, while others may have limited computer access or may not grant access to special-purpose software. Costs, internet access, and overarching district technology or privacy policies also sharply limit teachers’ options. Inclusive design requires attention to accessibility and adaptable features, such as alternative language support, from the onset of the design process: ensuring that students’ diverse needs are addressed as a core feature rather than being added as an afterthought at the end. Finally, attention to teacher confidence and self-efficacy are at the heart of our mission, as we work to raise teachers from passive consumers of CS curricula to engaged co-producers of integrated CS experiences; our participants return to their classrooms with first-hand experiences in using CS knowledge to build solutions to problems they care about, precisely the kind of engagement we hope they are able to share with their own students when learning about CS.

3 Project Context

The design and implementation of novel CS-embedded curriculum units is the culmination of teachers’ participation in Project FUTURE – a multi-year, multi-state engagement between CS education researchers and elementary teachers. The project provided introductory CS professional learning to all elementary teachers (N = 791) at partner schools (N = 28) via Code.org CS Fundamentals [3] workshops (“CSF Intro”). Subsequent supports included the CSF “Deeper Dive” [4] as well as an instructional coaching program. In 2022, a subset of teachers from the original pool of participants were then recruited for a two-year unit development and testing process. This smaller, more intensive experience would gather teachers each year for a five-day Summer Intensive followed by two academic year sessions. The collaborative was launched in Wisconsin (N = 12) and Connecticut (N = 26).

The initial 2022 summer institute supported small groups of teachers in devising novel CS-embedded units, with an emphasis on unplugged instructional resources (e.g. do not require a computer) and existing digital artifacts. The goal of the culminating institute in the summer of 2023, and the focus of this experience report, was to realize the co-design of novel digital artifacts to support CS-embedded units. To achieve this goal, the summer institute featured an intensive collaboration between the teachers, with their expertise in curriculum design, and a team of CS content experts and software developers, capable of building a digital curriculum support tool to the teachers' specifications.

3.1 Digital Divide

One of the fundamental issues contributing to inequity in CS education is the digital divide, where disparities in access to technology and internet connectivity disproportionately affect students from under-served communities [5]. Students in economically disadvantaged schools often lack the necessary hardware, software, and internet access to meaningfully engage with CS curricula. This disparity is abundantly evident looking at our own data as well. For example, in Wisconsin we observed that districts with teachers who have participated in a CS education professional development (PD) program have an implementation rate between 40% and 50% from 2014 to 2022 – nearly half of the more than 2,500 participants went on to implement CS curriculum in their classrooms.

However, when we look at our largest and most diverse metropolitan school district, out of the hundreds of teachers who attended PD, less than 5% were able to implement in their classrooms. In our highest-need school districts, the implementation rate for high quality CS curriculum is lower by a factor of ten. Subsequent study has revealed that this alarming difference can not only be attributed to aforementioned barriers such as technology access, but also to other logistical barriers, including classroom time constraints. For example, in classrooms that contain students with greater needs of core-subject intervention, teachers have far less flexibility to introduce standalone CS curriculum.

Through Project FUTURE, teachers design custom CS lessons that integrate seamlessly into their existing curricula. This approach addresses barriers to CS education posed by time constraints in the standard school day. Alternatives such as after-school programming may be inaccessible in low-income school districts due to the additional cost of transportation and staffing. The process also allows teachers to build curriculum around their classroom's unique circumstances, opening new opportunities to promote CS in classrooms otherwise constrained by insufficient resources.

3.2 Inclusive Design

Integrating features to improve accessibility has long been a goal of curriculum designers in the CS education space [6]. Despite this, there are many communities that still face challenges engaging with the available resources; language barriers, physical or learning disabilities, and other obstacles introduce relevant complexities. Plenty of studies have attempted to address this issue by designing tools with the specific intent of promoting access to students who face such obstacles [7, 8, 9]. With support from the Project FUTURE development team, teachers were able to design and and implement digital tools that incorporated accommodations tailored to the specific needs of their student populations.

3.3 Teacher Confidence and Preparedness

CS is a relatively new and niche field when compared to other sciences, which results in a population of educators who have had minimal exposure to CS. This means current educators face lessened preparedness, confidence, and background to teach this foreign subject. Coupled with a limited amount of space in the curriculum to supplement with additional units, the integration of CS in elementary schools requires creative adaptations that better suit the needs of the teacher.

Project FUTURE aims to boost teacher confidence and preparedness in a number of ways. Through professional learning workshops, teachers are able to deepen their CS knowledge and foster their ability to teach CS in the classroom. The recurring academic year sessions give teachers a chance to reflect on their progress with the Project FUTURE team and brainstorm ways to improve their approach to CS instruction. Additionally, they are provided an extensive support system through their connection with the Project FUTURE team, as well as the network of other teacher participants.

4 Framework

The culminating summer institute convened in each state over subsequent weeks: in Wisconsin with a cohort of 12 teachers from Milwaukee Public Schools, and then in Connecticut with a cohort of 26 teachers representing various school districts. The teachers assembled into professional learning communities (PLCs), each of which were expected to design one CS-integrated unit involving some sort of digital tool or component. The PLCs were typically comprised of teachers from the same district, or neighboring districts, who shared a similar topic or standard. Throughout the week of the institute, the teachers would rotate between formal professional development modules and flexible time to work toward conceptualizing their CS-integrated unit. Each day, teachers would have an opportunity to consult with a team of developers about the digital component of their unit. A team of faculty and facilitators from both universities oversaw the overall week of professional development.

The co-design work of the summer institute can be broken down into six phases:

1. *Conceptualizing*: Working from their overall unit design, each PLC conceptualizes a digital tool to provide an integrated and authentic application of CS and the concepts of the target content for students.
2. *Discussion*: The PLC groups outline the core idea and key details of their conceptualized digital tool to the developer team in breakout sessions. If necessary, the developers contribute their own background expertise to help draw out where and how CS learning can be expressed further.
3. *Research*: The developer team conducts independent research surrounding related work and technological considerations for designing to the teachers' specifications.
4. *Prototyping and Continued Discussion*: The developers present a prototype of the digital tool to the teachers. Teachers provide their initial feedback on the digital tool.
5. *Polishing*: Using the feedback provided by teachers, the developers further refine the tool and make adjustments where necessary.
6. *Professionalizing*: The developers make final considerations to improve sustainability, accessibility, scalability, and efficiency to promote the tool's widespread adoption.

5 Development Team

For the 2023 summer institutes, the development team was comprised of 8 undergraduate and 2 graduate student researchers from Marquette, plus a faculty mentor. For the first Wisconsin summer institute, the entire development team participated in the breakout discussions with the teacher PLCs. Due to the distance (1000 miles) between the two states, the second institute split the development team into an *away team* of 6 students plus the faculty mentor who would lead the PLC meetings on site, and a *home team* who provided remote research and technical support focused on prototype development.

5.1 Preparation

The summer institute included hour-long windows of time for PLC groups to consult with the development team each day. To promote productivity and a positive experience, the team needed to ensure the breakout sessions were led efficiently and captured as many details as possible about their digital tool design.

Digital Tool Design Template

To keep the discussions on track, we created a template to guide the conversation and prevent crucial details from being missed. The template was split into three sections: “Hardware Constraints”, “Digital Tool Description”, and “Storyboarding”. PLC groups were provided time to preview the template and provide written responses ahead of the breakout sessions, but it was mostly expected that the development team would need to ask follow-up questions to elicit a strong set of requirements for each digital tool. The document is ordered to address hardware constraints first so that the development team could get an idea of what level of technology access was available to the students before thinking about the digital tool’s specifications (tech resources varied considerably across the breadth of school districts participating). The template then outlines a series of questions meant to draw out key details regarding the design of the digital tool. PLC teachers were then asked to illustrate storyboards demonstrating different examples of student interactions with the digital tool. This was meant both to provide the developers with a better visual understanding of the digital tool’s design as well as to capture additional details that may have been missed with the previous questions. Questions included in the Digital Tool Design Template are outlined in Table 1.

Existing Classroom Technology

It was expected that many teachers might not be aware of the wide breadth of existing CS education curricula and digital education tools for K-5 classrooms beyond those presented in their introductory workshops. To give the teachers a sense of what was available, each member of the development team selected a popular computing or robotics education tool and prepared a demo to present to the teachers on the first day of each institute. Each team member served as a subject expert on a tool throughout the week. Examples of the tools presented include: Scratch [10], BeeBot [11], Lego Mindstorm [12], Ozobot [13], Edison Robot [14], Sphero [15], and the Micro:bit [16]. If PLC groups wanted to design their CS-integrated unit around pre-existing tools and did not require a custom solution, they were encouraged and supported in exploring those options. While not all teachers had access to these tools through their schools, Project FUTURE

Hardware Constraints
What hardware will the students have access to? (Chromebook, iPad, Windows desktop, etc. . .) Will the students need to interact with the tool through their internet browser? Will the tool need to be accessible when not connected to the internet? Do the students have a reliable internet connection at school (or at home if necessary for the activity)? Does the school have any internet restrictions that may limit student access? What additional materials might you need so that students can use your digital tool? (i.e. faster internet, fewer internet restrictions, computer mice/keyboards, robots, etc.)
Digital Tool Design
Are there any existing artifacts that inspired the idea for your digital tool or accomplish something similar to what you hope to create? Which elements will be preserved? Which elements will be different? (i.e. existing Scratch projects, Code.org lessons, etc.) Will the project require custom images/sprites/backgrounds or use existing materials? If not, what might you need us to create for you to make your tool more accessible, engaging, and differentiated for your students? How will the students interact with the digital tool? (via mouse clicks or keyboard inputs on a computer screen, via touch inputs on a tablet computer, via voice commands, remote control, motion sensing, etc.) What instructions/context will the students be given for the activity surrounding the digital tool in the classroom? (Will they be written instructions, a video demonstration, mixed media? And what are the specific instructions that will be given to students to inform them on how to complete the activity?) How will student success be measured? What might an unsuccessful interaction look like? Describe an example of a student interacting with the digital tool step by step on a successful attempt. If the student can experience an unsuccessful attempt interacting with the digital tool, please describe an example of what that might look like. At what points in using the digital tool do you expect students might have some problems? (set-up, running a program, following your given instructions, utilizing their computers, etc.)
Storyboarding
Please illustrate/storyboard key moments during user interaction using the following questions as a guide: How will the tool be presented before the student starts interacting with it? How will the tool appear once the student has successfully completed their task with it? When the student fails the task? What will the change look like after a student gives specific input? What other key moments can be described/illustrated to help the development team?

Table 1: Digital Tool Design Planning Document Questions

was able to provide access through a checkout system while they were implementing their units. The demonstrations were also meant to show the PLCs different possibilities they could explore when designing a custom solution with the development team.

6 Summarizing the Summer Experience

On the morning of Day 1 of the Wisconsin summer institute, the development team presented the demonstrations of existing digital tools to the teachers. As anticipated, the teachers were unfamiliar with many of the tools and were eager to experiment with the technology immediately following the demonstrations. As the teachers familiarized themselves with the technology, they were able to converse with the development team about potential use cases and cost constraints surrounding the tools. In the time following the demonstrations, the teachers were then given time to meet with their PLC groups to brainstorm their plan for incorporating a digital component into their CS-embedded units. While many still needed a custom solution, others chose to explore pre-existing tools that could fit their needs. The first breakout session with a PLC group was held on the afternoon of Day 1, and by the end of the third day, all PLC groups requiring a custom solution had arrived at a concept for their digital tool.

The subsequent summer institute in Connecticut was run following the same format. This time, PLCs were encouraged to meet in breakout sessions with the development team early in the week, regardless of whether or not they had fully conceptualized the idea for their digital tool. This allowed PLCs to gather input from the development team early in the design decision process. PLCs were then able to schedule additional breakout sessions with the development team to share additional ideas they came up with as the week progressed. Overall, this iterative design strategy helped many PLCs in Connecticut construct more thoughtful designs for their digital tools.

In both states, a subset of the development team was designated to begin researching and prototyping the PLCs' digital tools as soon as we had started the initial breakout sessions. On the final day of the institute the development team showcased these prototypes to the teachers to collect early feedback on their design. This helped to ensure alignment between the development team and what teachers had envisioned in their design.

6.1 Observations and Reflection

While some PLC groups arrived at the institute with already formed ideas for their digital tool, others groups seemed more intimidated by the task of coming up with a design. The professional learning provided throughout the week, as well as the opportunity to brainstorm with the development team and other Project FUTURE teachers, helped the struggling PLCs to generate ideas that could be suitable for their classrooms. By the end of the institute, even the PLCs who were initially a bit reluctant grew in excitement surrounding their digital tools and were eager to contribute new ideas.

Many PLCs took the opportunity to integrate accessibility features where it was necessary to accommodate their students of varying abilities. Some of these features included translation options for students who speak English as a second language, dictation options for students who are preliterate or hard of hearing, and other user interface design decisions to cater toward different learners. The next section will exhibit some examples of how the teachers incorporated such features.

Our experiences led to several planned changes in the structure of future iterations of the summer institute. We realized during the second institute that the development team could help speed up the initial brainstorming process for the PLCs by scheduling a breakout session regardless of where the PLC was in the process. Consequently, PLCs had more time to improve upon their original design and schedule additional breakout sessions with the development team, leading to higher impact tools overall. We recognized the advantages of putting greater emphasis on accessibility considerations in the digital tool design template and demonstrations of preexisting tools. The inclusion of questions focused on accessibility features more effectively revealed key parts of the design – expressed as integral components by the teachers. Additionally, placing greater emphasis on accessibility considerations inspired even more ideas which aimed to provide more equitable student access.

7 Post-Institute Development

In total, the PLC groups designed 17 novel CS-integrated units across the two summer institutes. Of these, five required novel digital artifacts to be developed. The rest of the units were designed around an existing CS education tool. A record of each project and its requirements was created, and each member of the development team was assigned one project to deliver. We could not possibly fit detailed descriptions of every project within this report, so we have selected two stand out artifacts to examine more closely.

Artifact 1, Digital Mood Meter: The PLC group behind the *Digital Mood Meter* project was inspired by the concept of the “Mood Meter” outlined in the *RULER approach* developed by the Yale Center for Emotional Intelligence [17]. RULER is an evidence-based approach to social-emotional learning that aims to teach students to understand and regulate their emotions [18, 19]. The Mood Meter is one of the various tools utilized by the RULER Approach, and is specifically designed to build students’ self-awareness and broaden their vocabulary for communicating their emotions. In the original Mood Meter concept, students would choose from an array of words to describe their current emotional state.

The Mood Meter PLC was comprised of teachers from a district that had mandated this curriculum across all of their K-5 classrooms. While conceptualizing a digital translation of the Mood Meter exercise, the PLC saw an opportunity not only to integrate computing concepts and social-emotional learning, but to create an exercise that was more engaging, and more accessible, to their students. They also realized the potential of the digital tool to measure students’ emotional growth over time. While there may be other digital re-imaginings of Mood Meter [20], none exist that combine all of these elements.

During the discussion phase, the PLC’s general concept for a Digital Mood Meter was that the students would be presented with a grid containing a spectrum of positive and negative emotions with a similar layout to the paper-based original. The students would then be able to use familiar Scratch-style blocks to navigate their avatar to the emotion that best reflected their current state. The teacher could then view metrics on how students’ responses changed over multiple attempts

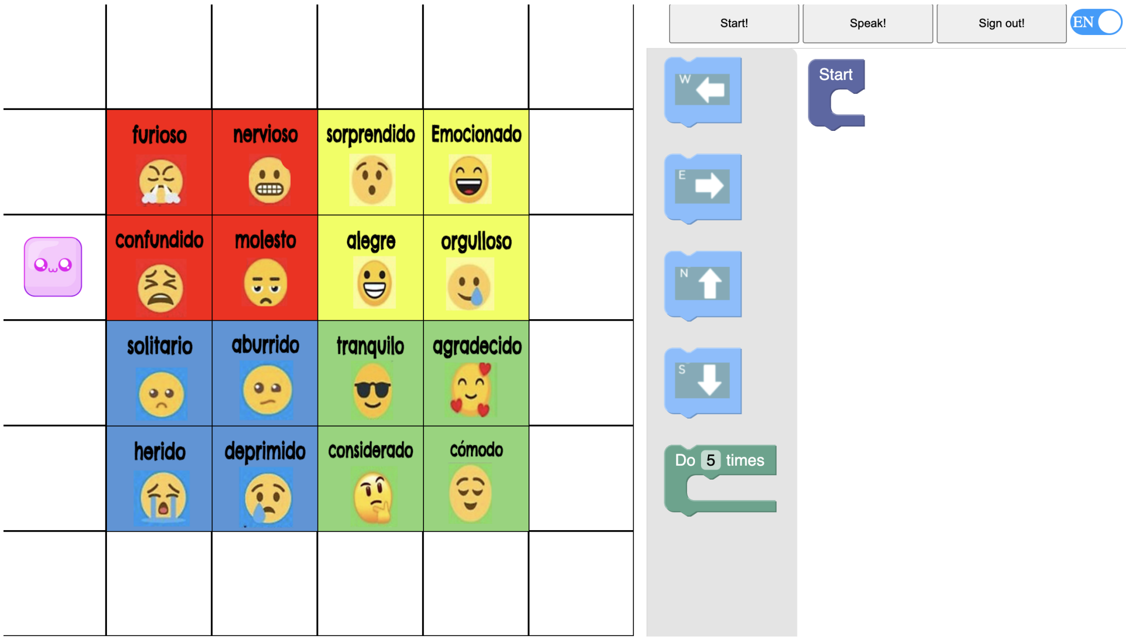

Figure 1: Mood Meter ”Jr.” interface for younger students

on the same day, and throughout the course of the year as they gained more experience describing their emotions.

With this description, the PLC had already accomplished the primary goal of integrating computing concepts, but the flexibility of re-imagining Mood Meter in a digital environment compelled them to explore additional ways to enhance student learning. To provide accessibility accommodations to students who speak English as a second language, a toggle button was added to translate the descriptions from English to Spanish. Additionally, dictation support was added for students who may have vision impairments or experience other reading challenges. The pronunciation of the word is followed by the definition as well, for students who may not be familiar with a particular emotion. The PLC group also requested a junior version of the digital mood meter with a more limited vocabulary for younger students who were just beginning to broaden their emotional intelligence. This version paired each word with a visual expression of the emotion to help students connect the meaning of each description. Both applications feature the ability for students to customize their avatar, adding another level of personal connection and engagement to the activity.

Artifact 2, Storytellers: The ”Storytellers” project built upon an unplugged lesson that one of the PLCs had developed with Project FUTURE in the previous year. In the original concept, students were tasked to write their own story, and then describe the proceeding scenario by drawing out action blocks under basic conditionals. An example would be: IF I am called names, THEN I would go sit with my friends; ELSE I would keep playing basketball. The lesson also aimed to encourage students to make good and emotionally conscious decisions. The teachers worked with younger students and felt that simply migrating this exercise into a block-based code editor would be a valuable learning experience. However, their stretch goal was to devise a way to simplify animation programming so that the students were capable of having their scenarios play out on screen. To accomplish this, the development team created custom action blocks within the Scratch environment that the students could use to play out their scenarios. This would allow students to practice with conditionals, and play out their unique stories, without the additional complexity of animating the sprites themselves.

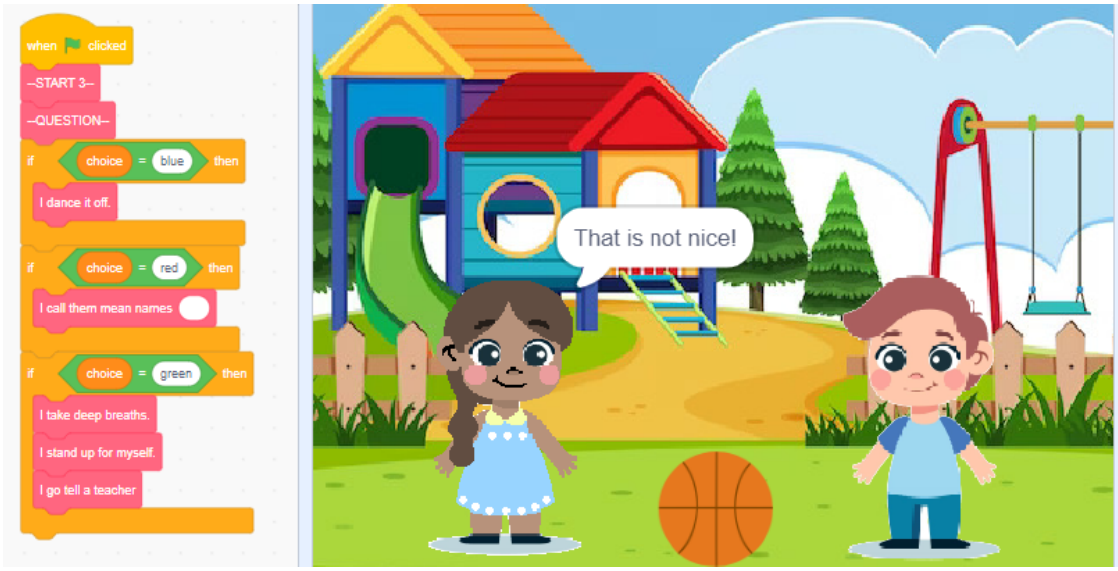

Figure 2: Example of scenario using custom action blocks in the Storytellers project

The blocks could be used to animate various actions students might want to depict in their story, many examples of which were pulled directly from student story submissions from the year prior. Examples of potential actions include "I stand up for myself," "I call them mean names," and "I dance it off," and it would be up to the students to determine whether the scenario depicted positive or negative actions in the broader activity. The assortment of context-dependent and general-use blocks allowed for seamless interchanging and sequencing to play out endless unique scenarios. Additionally, a multiple-choice selection block allows the students to create choose-your-own-adventure stories that use conditionals to branch in different directions based on the user's selections during runtime. To allow even more creative power and to help the students personally identify with their project more, *player* and *teacher* assets were expanded to include a variety of diverse characters that the students can choose to include in their story. The digital tool provided students the ability to make a simplified program, exposing them to sequencing, function input/output, basic conditionals, and troubleshooting.

8 Conclusion and Future Work

Putting teachers at the forefront of designing these units and tools realized the potential impact this collaboration can have in the classroom. Not only can these tools be used to integrate computing concepts across all areas of study, but many accomplish this while also improving other aspects of the student learning experience. The tools produced thus far could only have been possible through this collaboration as neither the development team, nor the teachers, wielded both the knowledge of curriculum design and technical expertise to create them independently.

The collaboration also addresses many of the barriers that prevent equitable access to CS education. Teachers employed numerous and varied strategies to make the tools accessible to students of varying abilities. Additionally, the CS-embedded units are designed to integrate with their standard curriculum, allowing the teachers to provide CS instruction in classrooms where it would not normally be feasible.

Figure 3: Students interacting with one of the digital artifacts titled "Long Division Helper"

At the time of writing, most of the digital artifacts have been developed and piloted in the classroom. This initial pilot has allowed teachers to observe how their students interacted with the tool, informing additional improvements to their original design. For example, after piloting the digital artifact *Long Division Helper*, shown in Figure 3, the teacher realized that introducing dictation support would grant her students of varying reading abilities a more level playing field when learning the target math content.

During the coming 2024/2025 academic year, many additional artifacts that were in continued development will make their debut in the classroom. Participating teachers will continue to refine their digital artifacts in response to the feedback gathered after deployment. As teachers iteratively improve their designs, we will workplement greater data collection and assessment tools to measure the learning outcomes of participating students.

9 Acknowledgments

This work was supported in part by the National Science Foundation REU site grant #1950826, “Data Science Across the Disciplines”, and by U.S. Department of Education EIR Award #U411C190254, “Project FUTURE: FUndamentals Teachers Unit Research Exemplars, Innovations in Embedded Computer Science for Elementary Curricula” at Marquette University and Sacred Heart University.

References

[1] "Announcing the computer science for all initiative," US Department of Education, 02 2016. [Online]. Available: https://sites.ed.gov/whhbcu/2016/02/05/announcing-the-computer-science-for-all-initiative/

[2] "2022 state of computer science education," Code.org, 2022. [Online]. Available: https://advocacy.code.org/stateofcs

[3] "Computer science fundamentals," Code.org, 2023. [Online]. Available: https://code.org/educate/csf

[4] "Computer science fundamentals deep dive workshop," Code.org, 2023. [Online]. Available: https://code.org/professional-development-workshops

[5] J. E. Dolan, "Splicing the divide: A review of research on the evolving digital divide among k–12 students," *Journal of Research on Technology in Education*, vol. 48, no. 1, pp. 16–37, 2016.

[6] E. Gellenbeck, "Integrating accessibility into the computer science curriculum," *J. Comput. Sci. Coll.*, vol. 21, no. 1, p. 267–273, oct 2005.

[7] M. Alper, "Making space in the makerspace: Building a mixed-ability maker culture," *Proceedings of the Interaction Design and Children (IDC-13), New York, NY, USA*, pp. 24–27, 2013.

[8] A. Hurst and S. Kane, "Making "making" accessible," in *Proceedings of the 12th International Conference on Interaction Design and Children*, ser. IDC '13. New York, NY, USA: Association for Computing Machinery, 2013, p. 635–638. [Online]. Available: https://doi.org/10.1145/2485760.2485883

[9] M. Worsley and D. Bar-El, "Inclusive making: designing tools and experiences to promote accessibility and redefine making," *Computer Science Education*, vol. 32, no. 2, pp. 155–187, 2022. [Online]. Available: https://doi.org/10.1080/08993408.2020.1863705

[10] M. Resnick, J. Maloney, A. Monroy-Hernández, N. Rusk, E. Eastmond, K. Brennan, A. Millner, E. Rosenbaum, J. Silver, B. Silverman *et al.*, "Scratch: programming for all," *Communications of the ACM*, vol. 52, no. 11, pp. 60–67, 2009.

[11] S. Papadakis and M. Kalogiannakis, "Learning computational thinking development in young children with bee-bot educational robotics," in *Research anthology on computational thinking, programming, and robotics in the classroom*. IGI Global, 2022, pp. 926–947.

[12] N. C. Zygouris, A. Striftou, A. N. Dadaliaris, G. I. Stamoulis, A. C. Xenakis, and D. Vavougios, "The use of lego mindstorms in elementary schools," in *2017 IEEE Global Engineering Education Conference (EDUCON)*. IEEE, 2017, pp. 514–516.

[13] N. Körber, L. Bailey, L. Greifenstein, G. Fraser, B. Sabitzer, and M. Rottenhofer, "An experience of introducing primary school children to programming using ozobots (practical report)," in *The 16th Workshop in Primary and Secondary Computing Education*, 2021, pp. 1–6.

[14] C. Ververi, T. Koufou, A. Moutzouris, and L.-V. Andreou, "Introducing robotics to an english for academic purposes curriculum in higher education: The student experience," in *2020 IEEE Global Engineering Education Conference (EDUCON)*. IEEE, 2020, pp. 20–21.

[15] M. Ioannou and T. Bratitsis, "Teaching the notion of speed in kindergarten using the sphero sprk robot," in *2017 IEEE 17th International Conference on Advanced Learning Technologies (ICALT)*. IEEE, 2017, pp. 311–312.

[16] M. Kalogiannakis, E. Tzagaraki, S. Papadakis *et al.*, "A systematic review of the use of bbc micro: bit in primary school," in *Conference Proceedings. New Perspectives in Science Education*, 2021.

[17] M. A. Brackett, C. S. Bailey, J. D. Hoffmann, and D. N. Simmons, "Ruler: A theory-driven, systemic approach to social, emotional, and academic learning," *Educational psychologist*, vol. 54, no. 3, pp. 144–161, 2019.

[18] C. Hagelskamp, M. Brackett, S. Rivers, and P. Salovey, “Improving classroom quality with the ruler approach to social and emotional learning: Proximal and distal outcomes,” *American journal of community psychology*, vol. 51, 02 2013.

[19] C. Reyes, M. Brackett, S. Rivers, M. White, and P. Salovey, “Classroom emotional climate, student engagement, and academic achievement,” *Journal of Educational Psychology*, vol. 104, pp. 700–712, 08 2012.

[20] M. A. Brackett, R. S. Stern, D. N. Simmons, and N. A. Elbertson, *Implementing social and emotional learning (SEL) in classrooms and schools*. National Professional Resources, Inc., 2019.