Interactive Learning Environments

ISSN: 1049-4820 (Print) 1744-5191 (Online) Journal homepage: https://www.tandfonline.com/loi/nile20

Engaging youth in computational thinking practices through designing place-based mobile games about local issues

Breanne K. Litts, Whitney E. Lewis & Chase K. Mortensen

INTERACTIVE LEARNING ENVIRONMENTS
https://doi.org/10.1080/10494820.2019.1674883

Engaging youth in computational thinking practices through designing place-based mobile games about local issues

Breanne K. Litts [a], Whitney E. Lewis [a] and Chase K. Mortensen [b]

[a]Instructional Technology & Learning Sciences, Utah State University, Logan, Utah, USA; [b]Computer Science, Utah State University, Logan, UT, USA

ABSTRACT
In response to a need to equip youth to become successful contributors to our growing digital economy, educators and researchers are exploring ways to incorporate computational thinking (CT) for *all* across curricular domains. In this paper, we take a place-based approach to examine how and what CT practices youth learn through designing mobile games in and for their own communities. We conducted three after school workshops with 29 participants (13 female, 16 male, ages 10–16) in a rural city in the Western United States. Youth designed place-based, mobile games to share stories and experiences about local environmental or civic issues using the Augmented Reality and Interactive Storytelling (ARIS) programming platform. We collected and analyzed a range of data including field notes, design artifacts, screencasts, and final reflective interviews. Using Brennan and Resnick (2012) framework for studying and assessing the development of CT, we illustrate how youth engaged with CT practices and how their local topics facilitated their engagement with civic issues beyond the screen. Findings demonstrate how youth can engage with CT practices and their local communities through designing computational artifacts. We discuss implications for how leveraging place-based computational tools afford equitable and accessible CT integration in interdisciplinary contexts.

ARTICLE HISTORY
Received 13 August 2018
Accepted 27 September 2019

KEYWORDS
Computational thinking; design; place-based; augmented reality

Introduction

With the growing interest in equipping youth to be makers and not just consumers of digital artifacts, educators and researchers are exceptionally concerned with finding ways to incorporate computational thinking (CT) skills across curricular domains. CT "involves solving problems, designing systems, and understanding human behavior, by drawing on concepts fundamental to computer science" (Wing, 2006, p. 33). It consists of cognitive and problem-solving skills that outline how humans can process information like a computer scientist (Grover & Pea, 2013). Wing (2006) argued that CT is a "universally applicable attitude and skill set" for everyone to learn (p. 33), which has posed a critical design challenge to design accessible and equitable computational activities. As the Maker Movement (Anderson, 2012) collides with the CT movement, scholars have begun exploring how youth learn by designing computational artifacts as a possible solution to including diverse demographics and disciplines (Rode et al., 2015). Yet, there remain critical challenges of accessibility and equity with these approaches resulting in limited participation of learners across demographics and contexts (Kafai & Burke, 2013).

CONTACT Breanne K. Litts breanne.litts@usu.edu

As part of the larger effort toward integrating CT in education, Computer Science for All initiatives (The White House, 2016) promote the development of equitable computational tools, technologies, and approaches that support interdisciplinary, multimodal, and collaborative learning. In particular, scholars and practitioners have adopted visual programming tools for game design such as Scratch (Resnick et al., 2009) and App Inventor (Morelli et al., 2011) as one approach to achieving equitable CT integration. As learners design games within visual programming tools, they engage with CT skills through an iterative, complex design process (Lye & Koh, 2014). Yet, there remains an opportunity to think critically about how computational tools can be (re)designed and leveraged to integrate CT across diverse demographic and disciplinary contexts. In response to this need, we explore a place-based approach to designing computational tools by leveraging a location-based, mobile programming platform.

In this paper, we contribute a place-based lens toward CT integration by engaging youth in designing mobile games about civic and environmental education topics in their own community. Specifically, we examine how students engage with CT practices (Brennan & Resnick, 2012) through designing place-based, mobile games about local issues. We investigate how 29 rural youth (10–16 years old) design and develop place-based computational artifacts in the Augmented Reality and Interactive Storytelling (ARIS) platform, a narrative-based programming platform for novice or non-programmers. We were guided by the research question: What CT practices do youth engage with when designing and developing place-based, mobile games about local issues? Brennan and Resnick (2012) identify four CT practices, which are our focus in this paper: being incremental and iterative, testing and debugging, reusing and remixing, and abstracting and modularizing. We aim to not only contribute to the growing conversation of interdisciplinary CT integration by offering a new environmental education case, but also expand the conversation to include how learning through designing with computational tools can support CT integration more broadly.

Background

Visual programming tools that support creation

Scholars are working to expand inclusive designs, approaches, and tools to integrate CT, especially in K–12 settings. Schools are also working to incorporate CT practices both outright through computer science courses and through integrated discipline-based projects (Grover & Pea, 2013). Most current work around CT integration lies in the science, math, and engineering disciplines, as there are natural synergies between CT practices and the respective disciplinary practices (Weintrop & Wilensky, 2015). For instance, robotics projects have been adopted in an effort to integrate computing across all three disciplines (Baretto, 2012). Researchers have also explored more integrative curricular approaches through interdisciplinary design contexts for computation. Electronic textiles (e-textiles), for example, support novice programmers from non-dominant communities while bridging art, science, and computing (Kafai et al., 2014; Kafai, Searle, Martinez, & Brayboy, 2014). Growing evidence suggests that these interdisciplinary approaches to teaching CT practices support interdisciplinary content knowledge gains (e.g. Litts, Kafai, Lui, Walker, & Widman, 2017; Peppler & Glosson, 2013) and are widely used in curricula across the country such as the Exploring Computer Science curriculum (Goode & Margolis, 2011). With these new approaches is a need for computational tools that support interdisciplinary design.

Tools that promote creative programming spaces must have a “low threshold, high ceiling, and wide walls” to support novices’ and experts’ exploration (Resnick et al., 2005). This philosophy inspired the design of visual programming tools, thus, visual translations of text-based code are widely adopted in education contexts. Logo Turtle (Papert, 1980), for example, was one of the first efforts to translate abstract text-based code to a multimodal context using digital and physical metaphors. Since Logo, though, there has been a proliferation of platforms, approaches, and models for how to make text-based code more accessible through visual metaphors. A visual metaphor commonly adopted in K–12 education leverages a block-based interface through which color-coded, puzzle-like blocks are compiled for programming. Block-based environments such as Scratch (Resnick et al., 2009) are most commonly used to teach computational topics to youth across a range of contexts, including for mobile games with platforms like App Inventor (Morelli et al., 2011).

Though the spirit behind block-based environments is to make programming more accessible for novice programmers by supporting their ability to understand basic programming concepts (Resnick et al., 2005; Shapiro & Ahrens, 2016), there is evidence that block-based environments are limited by slower processing and less authentic environments (Weintrop & Wilensky, 2015). For instance, after building with a block-based environment in a middle school introductory programming course, students still lacked understandings of "how loops work, what variables are, and what they do in a programming context" (Grover & Basu, 2017). There is also evidence that these environments breed certain "habits of programming" that are at odds with acceptable computer science practices, which presents some particular roadblocks on the pathway from novice to expert programmer (Meerbaum-Salant, Armoni, & Ben-Ari, 2011). In response, some scholars explored alternative approaches to designing computational tools by leveraging a narrative metaphor to make programming more accessible as well as increase interest in computation, especially for girls and younger children (Kelleher, Pausch, & Kiesler, 2007; Ryokai, Lee, & Breitbart, 2009).

Drawing on this alternative approach, we employ ARIS, a narrative-based programming platform, which allows designers with limited programming skills to create production quality, location-based, narrative-centric mobile learning games (Holden, Gagnon, Litts, & Smith, 2014). ARIS affords designers the ability to program interactive stories using mobile, location-based technologies, which connect them to their local places and communities. In previous work, we have found, as a computational tool that supports production at the intersection of narrative and place, ARIS can facilitate new pathways to CT, especially for Indigenous youth who come from communities where storytelling is a strong cultural practice (Litts & Lewis, 2019; Searle et al., 2018; Searle, Casort, Litts, & Benson, 2017). With evidence of these affordances, we leverage ARIS to explore how location-based, mobile technologies enable rural youth to engage with CT practices in a civic and environmental education context.

Place-based approaches to design projects

Environmental education has traditionally consisted of scientific facts, yet there has been a recent shift toward designing more interactive and interdisciplinary outdoor learning approaches through which students critically engage with place at the intersection of disciplines such as science and history (Ernst & Theimer, 2011; Tsevreni, 2011). This shift aligns with Dewey's classic place-based approach to education with experiential learning through which students experience and learn within their local environments (Dewey, 1915). Likewise, the goal of a place-based approach is to connect students with their surrounding environment (Ernst & Theimer, 2011; Smith & Sobel, 2014) such as participating in and contributing their community through scientific observation or civic activism (Gruenewald, 2003; Hung, Lin, & Hwang, 2010; Tsevreni, 2011). As a result, there is a growing movement in education of leveraging mobile game design to engage youth in environmental and civic issues.

As part of this movement, Mathews and Holden (2012) introduce the idea of "place-based design education" through which youth become active participants in local civic issues through designing their own place-based games, stories, and experiences. For example, in the *Up River* project, Wagler and Mathews (2012) engaged youth in an ethnographic design process during where youth not only conducted interviews and collected data in their communities but also took the field data they collected and (re)created mobile games to teach others about the issues around their local estuary. In addition to several practitioner examples and stories about place-based design education (Dikkers, Martin, & Coulter, 2012; Holden, Dikkers, Martin, & Litts, 2015), there is a growing body of empirical literature exploring how youth engage with place through mobile technologies. Taylor (2017) argues that youth learn "locative literacies" through critically engaging with their communities through mobile augmented reality technologies. One outcome of developing locative literacies is participating in civic issues to (re)design place (Taylor & Hall, 2013). With this foundation, we explore how youth engage with CT practices through engaging with their communities by creating with place-based, mobile technologies.

Methods

Participants and workshop

In partnership with a local makerspace, we conducted three after school workshops with 29 participants (13 female, 16 male, 10–16) in a rural city in the Western United States. All participants assented and their parents consented to this IRB approved research study. Over the course of six two-hour meetings resulting in roughly 12 h of design time, youth designed place-based, mobile games with ARIS, a location-based, augmented reality technology. Prompts in each workshop differed slightly, but each targeted local themes whether about plants and animals or civic issues. For example, in the first workshop we prompted participants to "Make your own Pokemon Go with local plants and animals," whereas in the last workshop we pushed participants to consider how they could use ARIS to inspire and engage their local community members about a civic issue (e.g. wildfires, animal adoption, pollution, etc.). In each workshop, there were about 10 participants and at least three facilitators, who were also part of the research team. Familiarity with programming varied across participants with some having experience in text-based languages and others having no prior experience. We gave participants the option of working individually or together and eight participants opted to work in pairs (four pairs total), which resulted in 25 total games across all participants.

We modeled the ARIS design process after existing implementation models (e.g. Gagnon, Vang, & Litts, 2015; Litts, Martin, & Gagnon, 2013), so each participant went through a design process that included research, storyboarding, developing in ARIS, and playtesting and debugging (see Figure 1). Thus, each workshop generally followed the same rhythm: On Day 1, we introduced participants to the technology through playing an example game and began brainstorming and research on a topic of choice; Day 2 we guided participants through the storyboarding process for their game idea; Days 3–6 participants iteratively developed, playtested, and debugged their game. The storyboarding process we refined (Lewis, 2018) is freely available here: https://fielddaylab.wisc.edu/courses/storyboarding. The outdoor playtesting occurred in a field that bordered the building where we held the workshop.

ARIS

The ARIS platform includes a web-based editor for development, which is accessible on any web browser, and a client-based app, which is only accessible on iOS devices. Like other visual

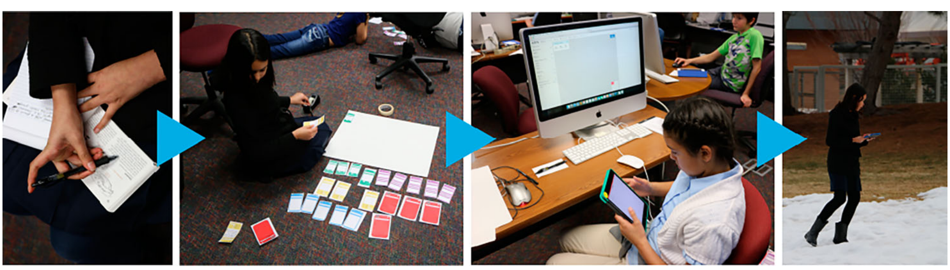

Figure 1. Gracie's storyboarding process (from left to right): research, storyboarding, developing in ARIS, and play testing.

programming platforms, ARIS is rooted in game design. Unlike other visual programming platforms, ARIS embodies a narrative metaphor for coding; thus, the fundamental objects are specific to game and story design. These include plaques (used for narration), items (which can be picked up, moved, exchanged, combined and affect other objects), characters (used within dialog and other interactions), quests (goals to move the story forward), and scenes (used to sequence game objects). These objects are instantiated within physical locations on a map or in a room with GPS, iBeacons, AR markers, or QR codes. While ARIS designers use principles of traditional computer science such as Boolean logic, variables, and conditional flow control, they do so through metaphors of "characters," "player inventories," and "scene" transitions. This accessible, narrative-based foundation of manipulating variables using logic and conditionals serves as an optimal platform for youth to tell local stories through designing place-based games.

Data collection and analysis

Across workshops, we collected a range of data including in-process audio recordings, design artifacts, daily screenshots of on-screen development, screen recordings, field notes from 3–4 researchers, and final reflective interviews. The in-progress data collection captured how participants engaged with the design process in-situ, while the final semi-structured interviews captured their reflections of the overall design process, how they utilized particular computational practices, and how the process facilitated their engagement with the topic they chose. For example, all interviews began with "Tell me a story about how you made your game," and then queried what specific successes and challenges participants achieved or encountered. After a brief explanation of what debugging is, we also specifically asked them to "give me an example of debugging that you had to do in your game." We also asked "Talk to me about how you think our time spent outside affected your design process and/or the outcome of your game." As is typical with semi-structure interviews, length varied such that interviews lasted 12–30 min.

The research team employed an iterative coding process of first cycle coding (e.g. descriptive and in vivo codes) and second cycle coding (e.g. pattern and elaborative codes) methods to develop a codebook that included definitions and examples (Saldana, 2009). We consolidated and calibrated the codebook through weekly discussion and deliberation. Research assistants applied the codebook to the participant interviews in pairs and resolved all disagreements to reach consensus on their code applications. Following this process, the research team worked together to triangulate data sources to more deeply develop salient codes. Driven by our specific interest in examining CT, we anchored our triangulation process in Brennan and Resnick's (2012) four CT practices: being incremental and iterative, testing and debugging, reusing and remixing, and abstracting and modularizing. As CT has become more commonplace, there are other well-developed frameworks of CT (e.g. Weintrop et al., 2016) to shape our understanding of CT in formal STEM contexts, yet Brennan and Resnick's (2012) framework is most suitable for our context as it was developed to understand CT skills and practices young people enact as designers of technology.

Findings

In our findings, we present how youth engaged with Brennan and Resnick's (2012) CT practices when designing and developing mobile, place-based games about local issues. We especially focus on the ways in which youth engage in CT practices that may be unique to designing mobile, place-based games.

Being incremental and iterative with ideas

The practice of being incremental and iterative takes CT into the realm of design, which is "an adaptive process, one in which the plan might change in response to approaching a solution in small

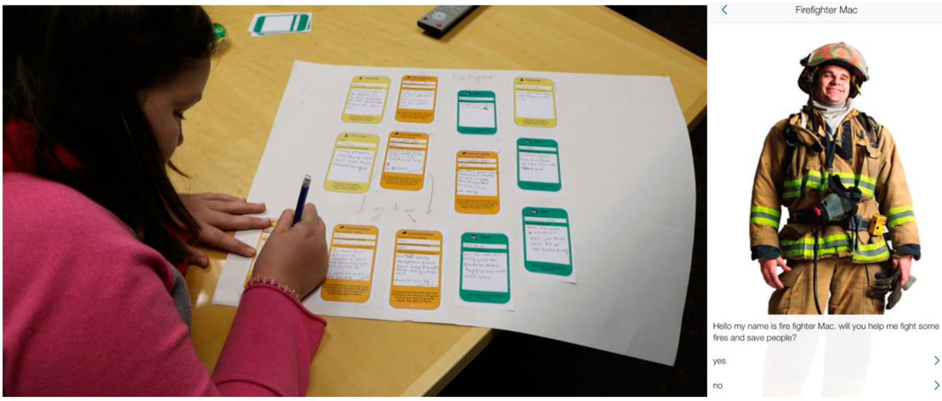

Figure 2. Lucy's storyboarding process for her "Firefighter" game (left) alongside one of her digital characters, "Firefighter Mac" (right).

steps" (Brennan & Resnick, 2012). In our workshops, this practice is structurally embedded into the design process, especially through the use of storyboarding, and how it emerged throughout the design process to shape youths' ideas. As part of the structure of the design process, participants develop a storyboard of their game idea. The storyboarding process includes ideating with large paper, markers, tape, and custom cards of various game objects within the ARIS platform (Figure 2). With these materials, youth easily and freely changed and modified their game ideas within the constraints, mechanics, and jargon specific to ARIS.

We found that youth engaged with incremental and iterative practices as their ideas developed, expanded, and even changed before even interacting with the digital design process. For example, Lucy, a 12-year-old female who made a game entitled "Firefighter" in our third workshop, initially had difficulty developing her idea: "I had all these ideas at once … so the hardest part was trying to figure out how they were connected and how make it flow easily from one to the next, instead of having it confusing" (Interview, 02/05/2018). She leveraged the storyboarding tool to help her remember these ideas and begin to establish her game idea's flow. In addition to thinking about the game's flow, she also wrote down numbers next to the cards on her storyboard to show the sequencing of her game idea.

When Lucy began the digital construction of her game, though, she realized that she could connect her different ideas through a narrative. She explains:

> Uh, well, I was gonna have at the very beginning that you had to save these people while also collecting the candles, but I had a better idea that you'd collect all the candles and stop the fire, and then you can say, like I could say, maybe, 'Oh, you missed one and the fire started. Now you have to go save those people' and … So one thing would lead to another, instead of both at the same time [laughter] with like no explanation. (Interview, 02/05/2018)

By using the storyboarding tool to help her design her programming plan, Lucy was able to overcome the challenge of figuring out how to connect all her ideas and seamlessly adapt to her expanding game ideas

Similarly, Samuel, a 12-year-old male who made a game entitled "Animal Keeper" in our first workshop, adapted his game idea to respond to the technological affordances of ARIS. Specifically, as he began to understand what it means to design a place-based game, he adapted his content to suit this concept. A workshop facilitator recounted this shift:

> I had a conversation with [Samuel] about how ARIS is location-based, so that place with specific trees can actually be in a specific real location in his neighborhood or wherever he chooses. He lit up when I said that, because it was like something clicked between linking the ideas to how it works in ARIS. (Fieldnotes, 11/3/2016)

With this newfound understanding of connecting his designed digital world with the real world, Samuel was driven to conduct more research about "animal feeding" and design his game around local ecosystems (Interview, 11/17/2016). He explains, " … if you learned about ecosystems or something like that it could have helped you change it … Like, that things depend on other things or they die" (Interview, 11/17/2016)

Across workshops, youths' design processes were not "a clean, sequential process," but rather adaptive to the narrative and place-based technological affordances of the computational design tool (Brennan & Resnick, 2012). Whether youth were expanding their idea or completely reinventing the design of their project, they deeply engaged with the CT practice of being incremental and iterative as they adapted their game idea in response to technological constraints and computational challenges.

Testing and debugging between digital and physical worlds

A necessary practice in any design process is to "develop strategies for dealing with – and anticipating – problems" (Brennan & Resnick, 2012). This practice of testing and debugging is not a new concept in design and production contexts, but we found that youth engaged with testing and debugging in ways unique to designing mobile, place-based games. At some point in the design process, all youth in our workshop began testing and debugging across digital and physical space and time such that they made digital modifications and then playtested in the real world. For example, Gracie, a 10-year-old female who made a game entitled "All Doggies are Cute" in our second workshop, worked through a sophisticated debugging process by first programming a factory, an ARIS game element that spawns an item in the game world based on an algorithm with several manipulatable variables including distance and time (see Figure 1 above for her design process). Gracie shares her challenge with testing and debugging the factory algorithm:

> Since I don't want it to be too easy I accidentally made [the item] really, really far. That I think I had to walk two of these buildings just to get it. But then when I get there, it already is gone and then I have to walk all the way back. And then I found one that I had to walk again, by the time I get there it's gone. (Interview, 03/30/17)

Gracie illustrates a conceptual challenge many youth have in our workshops: mapping digital algorithms onto physical space. Across our workshops, almost all participants have had difficulty connecting the factory algorithm to real-world time and space. For example, Doug a 12-year-old male who made a game entitled "Wolf Quest" in our second workshop, repeatedly "broke" the algorithm by inputting extreme values, running outside to experience and embody the algorithm, and repeating until he understood (see Figure 3). In practice, this meant that Doug literally sprinted or walked around the playtesting field iteratively until he understood what exact function of each variable in the algorithm meant in terms of gameplay experience. This conceptual challenge was at times due to a lack of understanding or expertise with the tool, while at other times it was due to a poor user interface that did not afford digital feedback of how computational changes would impact real-world gameplay.

Through these trial-and-error approaches to debugging, most youth actually leveraged their newfound knowledge to design their games around this unique computational game mechanic. As an illustration, Gracie expounds how she redesigned her game based on her understanding of the algorithm:

> "Well, at first, for the like three levels it's gonna be pretty easy. But then the items are gonna be harder to catch: it's gonna be further away from you, you're gonna have less time, and it's gonna pop up, some really good things are gonna pop up less, like the "PUPPY!". One percent [likelihood of appearing] in [every] 10 s and I still haven't found it" (Interview, 03/30/17).

Hence, in our workshops, youth engaged with testing and debugging process through iterations of digital adjustments followed by in-situ testing, which is unique to mobile, place-based games. Youth

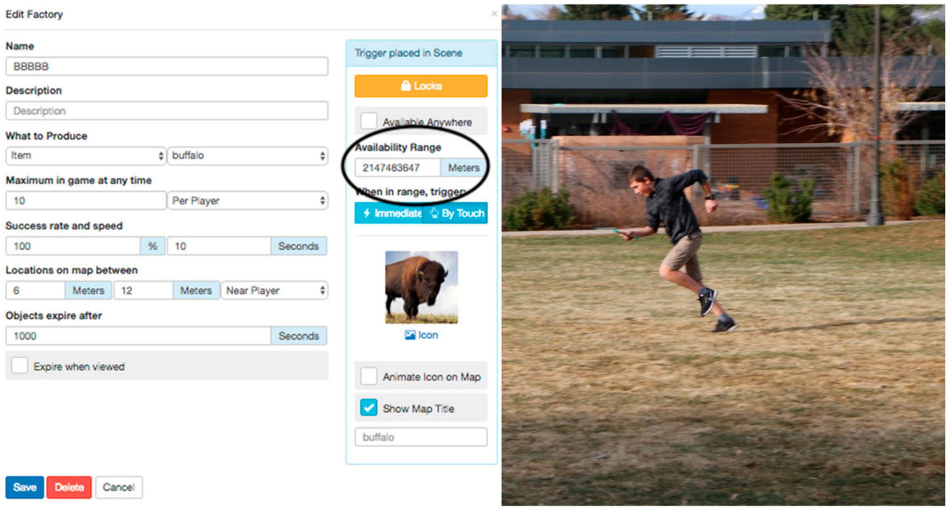

Figure 3. Doug's embodied debugging process: A bison factory with a high availability range (left) and Doug experiencing the output of that number in the world (right).

engaged in this embodied debugging process, which afforded them the ability to not only build their own understanding of the computational algorithm, but also their understanding of how the player will experience their game. Interestingly, as we traced youth's embodied debugging over their design process, we found that they applied their new knowledge within the same design process. For example, as illustrated above, Gracie describes leveraging her understanding of the computational algorithm as a game mechanic to make the game progressively more difficult. This also demonstrates how youth can develop computational literacy and expertise over the course of a singular design process.

Reusing and remixing with local and global communities

The third CT practice, reusing and remixing other's work, "has been a longstanding practice in programming, and has only been amplified by network technologies that provide access to a wide range of other people's work" (Brennan & Resnick, 2012). ARIS, as a tool, is free and open source and available to be reused and remixed by anyone; thus, it's embedded into the ethos of the tool, and the ARIS community, a public forum made up of users who share work and support each other. We found that reusing and remixing media (e.g. memes, images, art, etc.) was the most prevalent form of this practice across all participants. Moreover, the prompt for the first workshop, "make your own Pokemon Go," inherently required remixing of an existing game. In this case, youth translated game mechanics such as "evolving," which are critical in Pokemon Go, into their ARIS games. Furthermore, we found that the example games we used to introduce ARIS heavily shaped what game objects integrated in their designs. Youth first played, then remixed, and in some cases reused components of the example games in their own designs.

In addition to these more inherent examples of reusing and remixing, we also noted emergent ways youth engaged with the practice of reusing and remixing. First, youth shared their developing knowledge and expertise with others in their workshop. For instance, Samuel was the first one in the first workshop to figure out how to "evolve" an animal within the logic of ARIS and used that knowledge "to help other kids and explain things to them" and allowed them to reuse and remix his code (Fieldnotes, 11/08/2016). Additionally, Grant, a 13-year-old male who made a game entitled "Pokemon Stop" in our first workshop, carried this practice across workshops by returning as a mentor in our third workshop.

Second, a few participants reached out to the ARIS community during the workshop for help in fully realizing their game ideas. For example, Grant wanted to add collars as a way to "catch" animals in his game, but this required adding JavaScript to "hack" the template interface of ARIS. He explains, "I tried to do it and I couldn't figure out why and so I went onto that website … " and reached out to the ARIS forum (Interview, 11/12/2016). In the forum, other ARIS designers provided their JavaScript code, which Grant then reused and remixed to adapt to his particular game context and goals. Across workshops, only a few participants reached out to the ARIS forum, but the solutions to youths' problems were reusing and remixing others' code in order to program specific game mechanics they desired.

Abstracting and modularizing to build game logic

Abstracting and modularizing, or "building something large by putting together collections of smaller parts" (Brennan & Resnick, 2012), is a CT practice youth engaged with in sophisticated ways. Similar to what Brennan and Resnick (2012) describe with Scratch, we also found that creating game levels was a natural method of abstracting and modularizing for youth in our workshops. Of particular interest, though, were two youth, Samuel and Eric, who used the tools in the ARIS platform to build game logic with two very different methods. Tools for logic in ARIS include: scenes (sequencing modules), locks (conditional logic), triggers (Boolean logic), and conversations (tree or branching logic).

On the one hand, Samuel, who made "Animal Keeper" in our first workshop, used scenes to modularize his game into levels and utilized locks as the main flow control throughout the game. As a result, much of his logic in the game is not visible on the home screen (see Figure 4). On the other hand, Eric, an 11-year-old male who made "Animal Master" in our second workshop, used scenes to modularize each variation in his game and utilized locks for special events. Hence, the majority of logic in his game is visible on the home screen (see Figure 4). Both Samuel and Eric developed similar games about the local ecological food chain system, yet they built their game logic very differently and, thus, exhibited different methods of abstracting and modularizing. Samuel's method of relying primarily on locks is consistent with what most youth used across all workshops, whereas Eric's method of relying primarily on scenes solves a very real problem with the ARIS interface: the locks are not visible in the editor. Instead, Eric used the arrows between scenes to visualize his game logic. Figure 4 shows the stark visual contrast between these two methods.

Youth across workshops unanimously agreed that this was the biggest struggle of working in ARIS. Gracie expresses this frustration, " … but you wouldn't see [the lock], but it would work, but you wouldn't see it. Which that bugged me 'cause I wanted to physically see it instead of go test my game" (Interview, i03/30/2017). Eric's method of abstracting and modularizing presented an

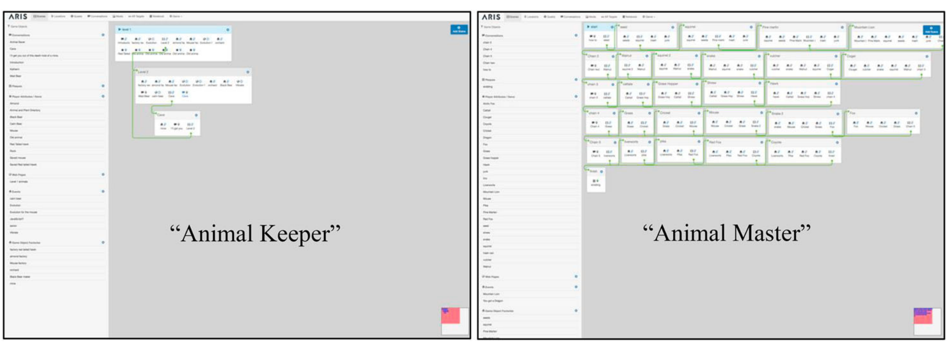

Figure 4. Comparison of two game logic strategies in the ARIS editor: Samuel's "Animal Keeper" (left) and Eric's "Animal Master" (right).

alternative method for visualizing game logic to design around an interface issue all of his peers had with the ARIS platform.

Discussion

In this paper, we investigated how youth engaged with CT practices through designing and developing place-based, mobile games about local issues. Our findings offer insights for how situating youth as creators can facilitate their engagement with CT practices and how a place-based approach affords interdisciplinary CT integration. The design considerations we contribute are relevant to the design of computational tools and activities, especially those embedded in interdisciplinary contexts across a range of settings where matters of accessibility and equity are critical.

Youth as designers with computational tools

In our workshops, youth engaged in a rich design process through which they realized and responded to the tensions between environmental or civic disciplinary content, computer science principles, and the affordances and constraints of the computational tool. Within our place-based design context, we found that youth engage with CT practices in ways that are unique to designing with location-based, augmented reality technologies. We identified scaffolds and strategies that support ideation, design, and development along digital and physical dimensions as well as local and global dimensions. Here, we spotlight two computational problem spaces that youth encountered when designing place-based, mobile games.

Embodied debugging

When considering a multidimensional problem space such as designing place-based, mobile games, it is not surprising that youth realize new ways of engaging with CT practices. Youth in our workshops engaged in an *embodied debugging* through which they experienced the effects of a computational algorithm in an augmented reality world. Embodied cognition and learning (Lindgren & Johnson-Glenberg, 2013; Wilson, 2002) suggests that humans often work out their thinking by interaction with the world through bodily or physical interactions. When situating learners as designers, scholars draw on an embodied understanding of knowledge to examine *bodystorming,* a design method for brainstorming "in the wild" and for testing and experiencing design in embodied way (Oulasvirta, Kurvinen, & Kankainen, 2003; Smith, 2014). In our study, we found that young designers are able to embody the debugging process by designing with location-based technologies. Thus, embodied debugging extends our current understanding of bodystorming beyond brainstorming to include what "in the wild" debugging looks like.

While embodied debugging might not be an efficient method of debugging with these technologies (or feasible to replicate in a K-12 classroom), it was certainly effective in enabling youth to learn the algorithm through a trial-and-error process of simultaneously mapping the algorithm onto digital and physical space and time. They persisted and endured several iterations of tweaking the computational algorithm and running outside to determine whether the output achieved their personal design goals. Much like Papert's (1980) Logo Turtle afforded real-world feedback in programming, the concept of embodying the CT practice of debugging prompts us to reimagine how we concretize, obscure, and abstract CT practices in the context of location-based, augmented reality technologies, especially when learners themselves are designers.

Mapping logic

Our study also expands the conversation of CT integration to include the design of computational tools, especially ones that support youth's computational design process. We note that transparency to the user is a critical design consideration. Youth in our study made apparent that the tools for logic are not transparent enough for novice programmers and may not be designed to support *mapping logic* across space and time, which is vital for supporting place-based, mobile game design. In the case of ARIS, a simple fix is to make the logic more transparent by providing visual feedback similar to how Eric achieved this by leveraging the visual feedback afforded by scenes, which resulted in a visual representation of logic using arrows. Even more, though, the logic tools available to youth did not easily enable them to map their logic onto their place-based, mobile games. Designers of learning technology challenge us to design beyond these "black box" instruments by making tools that are at the right "level" of understanding making key concepts visible as appropriate to learning goals (Resnick, Berg, & Eisenberg, 2000). Designing for transparency rather than embedding or assuming knowledge of computer science principles or privileging certain types of disciplinary practices will afford more flexible computation tools and is a critical consideration in designing for inclusive computing cultures.

Place-based approach to CT integration

We adopted a place-based approach to CT integration with youth in a place-based design education context. Though we structured workshops around environmental and civic issues, most youth designed games that spanned multiple disciplines. Anchoring design in their local communities enabled youth to consider multiple perspectives on their specific topic. Moreover, youth were also motivated by their chosen content and particular design goals to engage with CT practices in more sophisticated ways. At least two youth transitioned to text-based coding in order to add specific place-based game mechanics. This connection between content, design, and computation offers insights to how place-based designs might empower youth to probe diverse interdisciplinary content and more deeply engage with CT practices.

Furthermore, our findings bring to light key CT practices embedded in this particular environmental and civic context; however, there may be different patterns of engagement with CT practices in other disciplinary contexts such as history or english, which could require the use of different ARIS game objects. For example, the factory in ARIS was especially relevant to the youth's design goals of making animals spawn in the world automatically, however, in prior more narrative-focused implementations of ARIS (Gagnon et al., 2015; Litts et al., 2013) scholars noted that designers more predominantly use other game objects, such as characters. We argue that, as a result, the specific content could impact how youth engage with CT practices, which suggests there could be a correlation between disciplinary context and what engagement with CT practices looks like. Thus, when considering CT integration across domains, it is crucial to align the potential impact of context with desired learning outcomes.

Delimitations and limitations

While our work contributes unique insights to how computational thinking practices are engaged in place-based design contexts, we highlight delimitations and limitations to ensure findings and implications are not overstated. A key delimitation to our work is the visual programming tool, ARIS, that we chose to use for these workshops. While ARIS affords the type of place-based design we sought to investigate, implementations with other tools are necessary in order to more deeply understand the aspects of our findings that are particular to the tool vis-à-vis learning through place-based design. Moreover, there are two critical limitations in this study. First, as stated our participants signed up through a local makerspace, which meant that many of them already had programming experience or self-selected as someone who wants to learn to program. Hence, our findings only reveal what is possible with young people who are already interested in or have experience with coding. Second, the workshops were not structured to suit a formal learning contexts, thus, additional design research is required to better understand how our findings might or might not be relevant to formal contexts and appropriate learning standards.

Conclusion

As educators and researchers continue to work toward CT integration across curricular domains, issues of access and equity come to the fore. We provoked a conversation around leveraging a place-based approach to achieving CT integration across demographics and settings. Our findings prompt scholars to consider what CT practices look like when situating youth as designers with diverse technologies and to consider anchoring their designs in interdisciplinary problem space, such as place, and for an authentic audience, such as a local community, to enable equitable computation participation. Opportunities for future research include critically examining how present computational tools facilitate youth's design process and engagement with CT practices and how youth design place-based, mobile games in other disciplinary contexts.

Disclosure statement

No potential conflict of interest was reported by the authors.

Notes on contributors

Breanne K. Litts, an Assistant Professor of Instructional Technology and Learning Sciences and director of the Learn Explore Design (L.E.D.) lab at Utah State University, investigates how people learn by and collaborate through making, designing, and producing and develops technologies and learning environments to support these activities.

Whitney E. Lewis, a Master's of Science graduate from Utah State University, is interested in how people generate ideas and how supports can aid in the ideation process. Through her education and professional experiences, Whitney has committed to core design principles such as co-design, problem exploration, and rapid-prototyping.

Chase K. Mortensen, an Undergraduate Honors student at Utah State University, is currently pursuing a Bachelor's degree in Computer Science. He is interested in the intersection of mobile computing and augmented or virtual reality. His research examines how youth learn computational thinking across of range of mobile technologies.

ORCID

Breanne K. Litts http://orcid.org/0000-0003-4981-3515
Whitney E. Lewis http://orcid.org/0000-0003-3883-6148
Chase K. Mortensen http://orcid.org/0000-0002-0785-889X

References

Anderson, C. (2012). *Makers: The new industrial revolution*. New York: Crown Business.

Baretto, F. B. V. (2012). Exploring the educational potential of robotics in schools: A systematic review. *Computers & Education*, *58*, 978–988.

Brennan, K., & Resnick, M. (2012, April). New frameworks for studying and assessing the development of computational thinking. *Proceedings of the 2012 annual meeting of the American Educational Research Association, Vancouver, Canada* (pp. 1–25).

Dewey, J. (1915). *The school and society* (Rev. ed). Chicago, IL: The University of Chicago Press.

Dikkers, S., Martin, J., & Coulter, B. (2012). *Mobile media learning: Amazing uses of mobile devices for learning*. Pittsburgh, PA: Carnegie Mellon University: ETC Press. http://repository.cmu.edu/etcpress/62

Ernst, J., & Theimer, S. (2011). Evaluating the effects of environmental education programming on connectedness to nature. *Environmental Education Research*, *17*(5), 577–598.

Gagnon, D., Vang, R., & Litts, B. K. (2015). *Learning through design: ARIS*. In proceedings of the eleventh annual Games + Learning + Society Conference, ETC Press PA Pittsburgh, 282–288.

Goode, J., & Margolis, J. (2011). Exploring computer science: A case study of school reform. *ACM Transactions on Computing Education (TOCE)*, *11*(2), 12.

Grover, S., & Basu, S. (2017, March). Measuring student learning in introductory block-based programming: Examining misconceptions of loops, variables, and Boolean logic. *Proceedings of the 2017 ACM SIGCSE technical symposium on computer science education* (pp. 267–272). ACM.

Grover, S., & Pea, R. (2013). Computational thinking in K–12: A review of the state of the field. *Educational Researcher*, *42*(1), 38–43.

Gruenewald, D. A. (2003). The best of both worlds: A critical pedagogy of place. *Educational Researcher*, *32*(4), 3–12.

Holden, C., Dikkers, S., Martin, J., & Litts, B. (2015). *Mobile media learning: Innovation and inspiration*. Pittsburgh, PA: Carnegie Mellon University: ETC Press. http://repository.cmu.edu/etcpress/31

Holden, C. L., Gagnon, D. J., Litts, B. K., & Smith, G. (2014). ARIS: An open-source platform for widespread mobile augmented reality experimentation. In F. Neto (Ed.), *Technology platform innovations and forthcoming trends in ubiquitous learning* (pp. 19–34). Hershey, PA: IGI global. doi:10.4018/978-1-4666-4542-4.ch002

Hung, P.-H., Lin, Y.-F., & Hwang, G.-J. (2010). Formative assessment design for PDA integrated ecology observation. *Journal of Educational Technology & Society*, *13*(3), 33–42. Retrieved from http://www.jstor.org/stable/jeductechsoci.13.3.33

Kafai, Y. B., & Burke, Q. (2013). Computer programming goes back to school. *Phi Delta Kappan*, *95*(1), 61–65.

Kafai, Y. B., Lee, E., Searle, K., Fields, D., Kaplan, E., & Lui, D. (2014). A crafts-oriented approach to computing in high school: Introducing computational concepts, practices, and perspectives with electronic textiles. *ACM Transactions on Computing Education (TOCE)*, *14*(1), 1–20.

Kafai, Y., Searle, K., Martinez, C., & Brayboy, B. (2014, March). Ethnocomputing with electronic textiles: Culturally responsive open design to broaden participation in computing in American Indian youth and communities. *Proceedings of the 45th ACM technical symposium on computer science education* (pp. 241–246). ACM.

Kelleher, C., Pausch, R., & Kiesler, S. (2007, April). Storytelling Alice motivates middle school girls to learn computer programming. In *Proceedings of the SIGCHI conference on human factors in computing systems* (pp. 1455–1464). ACM.

Lewis, W. E. (2018). *Design scaffolding for computational making in the visual programming tool ARIS* (Unpublished Master's Thesis). Utah State University. Retreived from https://digitalcommons.usu.edu/cgi/viewcontent.cgi?article=8346&context=etd

Lindgren, R., & Johnson-Glenberg, M. (2013). Emboldened by embodiment: Six precepts for research on embodied learning and mixed reality. *Educational Researcher*, *42*(8), 445–452.

Litts, B. K., Kafai, Y. B., Lui, D. A., Walker, J. T., & Widman, S. A. (2017). Stitching codeable circuits: High school students' learning about circuitry and coding with electronic textiles. *Journal of Science Education and Technology*, *26*(5), 494–507.

Litts, B. K., & Lewis, W. E. (2019). Mobile augmented reality: Exploring a new genre of learning. *GetMobile: Mobile Computing and Communications*, *22*(3), 5–9.

Litts, B. K., Martin, J., & Gagnon, D. J. (2013). *Learning through design with ARIS Iteration 1: Digital literacies course [white paper]*. Madison, WI: Mobile Learning Incubator.

Lye, S. Y., & Koh, J. H. L. (2014). Review on teaching and learning of computational thinking through programming: What is next for K-12? *Computers in Human Behavior*, *41*, 51–61.

Mathews, J., & Holden, R. (2012). Place–based design for civic participation. In S. Dikkers, J. Martin, & B. Coulter (Eds.), *Mobile media learning: Amazing uses of mobile devices for teaching and learning* (pp. 131–148). Pittsburgh, PA: ETC Press.

Meerbaum-Salant, O., Armoni, M., & Ben-Ari, M. (2011). Habits of programming in Scratch. *Proceedings of the 16th annual ITiCSE conference* (Darmstadt, Germany, 2011) (pp. 168–172).

Morelli, R., De Lanerolle, T., Lake, P., Limardo, N., Tamotsu, E., & Uche, C. (2011). Can android app inventor bring computational thinking to k–12. *Proceedings of the 42nd ACM technical symposium on computer science education* (SIGCSE'11) (pp. 1–6).

Oulasvirta, A., Kurvinen, E., & Kankainen, T. (2003). Understanding contexts by being there: Case studies in bodystorming. *Personal and Ubiquitous Computing*, *7*(2), 125–134.

Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. New York, NY: Basic Books, Inc.

Peppler, K., & Glosson, D. (2013). Stitching circuits: Learning about circuitry through e-textile materials. *Journal of Science Education and Technology*, *22*(5), 751–763.

Resnick, M., Berg, R., & Eisenberg, M. (2000). Beyond black boxes: Bringing transparency and aesthetics back to scientific investigation. *The Journal of the Learning Sciences*, *9*(1), 7–30.

Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., … Kafai, Y. (2009). Scratch: Programming for all. *Communications of the ACM*, *52*(11), 60–67.

Resnick, M., Myers, B., Nakakoji, K., Shneiderman, B., Pausch, R., Selker, T., & Eisenberg, M. (2005). Design principles for tools to support creative thinking. Workshop on creativity support tools sponsored by the National Science Foundation, Washington, DC.

Rode, J. A., Weibert, A., Marshall, A., Aal, K., von Rekowski, T., El Mimouni, H., & Booker, J. (2015, September). From computational thinking to computational making. In *Proceedings of the 2015 ACM international joint conference on Pervasive and Ubiquitous Computing* (pp. 239–250). ACM.

Ryokai, K., Lee, M. J., & Breitbart, J. M. (2009, October). Children's storytelling and programming with robotic characters. In *proceedings of the seventh ACM conference on creativity and cognition* (pp. 19–28). ACM.

Saldana, J. (2009). *The coding manual for qualitative researchers*. Thousand Oaks, CA: Sage.

Searle, K. A., Casort, T., Litts, B. K., & Benson, S. (2017). Connecting space and narrative in Culturally Responsive making in ARIS with Indigenous youth. In *proceedings of 2017 FabLearn conference on creativity and fabrication in education*. ACM.

Searle, K. A., Casort, T., Litts, B. K., Brayboy, B. M. J., Dance, S. L., & Kafai, Y. B. (2018, June). Cultural repertoires: Indigenous youth creating with place and story. *International conference of the learning sciences*.

Shapiro, R. B., & Ahrens, M. (2016). Beyond blocks: Syntax and semantics. *Communications of the ACM*, *59*(5), 39–41.

Smith, B. K. (2014). Bodystorming mobile learning experiences. *TechTrends*, *58*(1), 71–76.

Smith, G. A., & Sobel, D. (2014). *Place-and community-based education in schools*. New York, NY: Routledge.

Taylor, K. H. (2017). Learning along lines: Locative literacies for reading and writing the city. *Journal of the Learning Sciences*, *26*(4), 533–574.

Taylor, K. H., & Hall, R. (2013). Counter-mapping the neighborhood on bicycles: Mobilizing youth to reimagine the city. *Technology, Knowledge and Learning*, *18*(1-2), 65–93.

Tsevreni, I. (2011). Towards an environmental education without scientific knowledge: An attempt to create an action model based on children's experiences, emotions and perceptions about their environment. *Environmental Education Research*, *17*(1), 53–67.

Wagler, M., & Mathews, J. (2012). Up River: Place, ethnography, and design in the St. Louis River estuary. In S. Dikkers, J. Martin, & B. Coulter (Eds.), *Mobile media learning: Amazing uses of mobile devices for teaching and learning* (pp. 41–60). Pittsburgh, PA: ETC Press.

Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., & Wilensky, U. (2016). Defining computational thinking for mathematics and science classrooms. *Journal of Science Education and Technology*, *25*(1), 127–147.

Weintrop, D., & Wilensky, U. (2015, July). Using commutative assessments to compare conceptual understanding in blocks-based and text-based programs. In *ICER* (Vol. 15, pp. 101–110).

White House. (2016). *Computer science for all*. Retrieved from https://obamawhitehouse.archives.gov/blog/2016/01/30/computer-science-all

Wilson, M. (2002). Six views of embodied cognition. *Psychonomic Bulletin & Review*, *9*(4), 625–636.

Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, *49*(3), 33–35.