Digital and Physical Fabrication as Multimodal Learning: Understanding Youth Computational Thinking When Making Integrated Systems Through Bidirectionally Responsive Design

GABRIELA T. RICHARD and SAGUN GIRI, The Pennsylvania State University, USA

This article proposes and explores the kinds of computational thinking, creative practices, design activities, and inclusive learning opportunities provided to diverse high school youth when designing integrated systems through simultaneously physically and digitally responsive wearable games and systems. Previous work in this area, conducted by Richard, coined the term “bidirectionally responsive design” (BRD) to describe the design of dual-feedback systems using multiple digital and physical interfaces. BRD also emphasizes using simplified fabrication tools, media and coding platforms, and microcontrollers common in youth content creation communities and makerspaces. This study provides a framework to analyze computational concepts, practices, and perspectives that leverage an integrated systems and multimodal learning approach, such as BRD, adding to, building on, and integrating previous analytic approaches to looking at Scratch coding, media design, physical computing and e-textiles. Using a detailed case study of one team during one of the early workshop iterations, we conduct a multimodal analysis of bidirectionally responsive making activities and discuss the ways that they present novel understanding of integrating diverse interests and encouraging collaborative and distributed computational thinking. We further examine how BRD operationalizes and extends multimodal learning theory by adding tangible and integrative dimensions as additional modalities learners can leverage to facilitate meaning making, metacognition, and agency. We also discuss how designing integrated systems, as facilitated through BRD, provides an opportunity to engage in authentic practices around the design of complex systems.

CCS Concepts: • **Social and professional topics** → **Computational thinking**; **Informal education**; **K-12 education**; *Model curricula*; *Computing literacy*; • **Applied computing** → **Interactive learning environments**; **Collaborative learning**; *Personal computers and PC applications*; *Computer games*; • **Hardware** → *Emerging interfaces*;

Additional Key Words and Phrases: Education/learning, schools/educational setting, participatory design, design research methods, collaboration, creativity support, games/play, sensors, tangible, wearable computers, embodied interaction, fashion/clothing, content analysis, interview, teens

This work was supported by Penn State University and a postdoctoral fellowship awarded to G. T. Richard by the University of Pennsylvania. Additional support for materials was funded through NSF grant #1027736, PI: Y. B. Kafai. The views expressed represent the authors and not necessarily those of the funding organizations or affiliated universities.

Authors' addresses: G. T. Richard and S. Giri, 301 Keller, Learning and Performance Systems, Penn State University, University Park, PA 16803; emails: {grichard, sqg5512}@psu.edu.

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

Copyright is held by the owner/author(s). Publication rights licensed to ACM.

1946-6226/2019/01-ART17 \$15.00

<https://doi.org/10.1145/3243138>

ACM Reference format:

Gabriela T. Richard and Sagun Giri. 2019. Digital and Physical Fabrication as Multimodal Learning: Understanding Youth Computational Thinking When Making Integrated Systems Through Bidirectionally Responsive Design. *ACM Trans. Comput. Educ.* 19, 3, Article 17 (January 2019), 35 pages.
<https://doi.org/10.1145/3243138>

1 INTRODUCTION

Maker activities and makerspaces have increasingly shown capacity for engaging learners in 21st-century literacies and skills. In an effort to capitalize on a variety of interests, many digital and physical toolkits (such as Scratch, Makey Makey, and the Lilypad Arduino) have emerged, along with various workshops and activities structured around designing with these tools. However, most workshops have focused on coding in Scratch, designing e-textiles, or creating physically responsive Scratch projects with Makey Makey. In other words, there have been limitations in understanding what kinds of computational thinking, interests, and practices might emerge if different kinds of technological capabilities are combined. Herein, we discuss the computational thinking skills and perception changes that occurred for a team of students learning bidirectionally responsive design—the process of creating projects with simultaneously digital and physical responsiveness by combining Scratch media design, e-textiles, and physical computing with Makey Makey.

While some initial studies have explored the computational concepts, practices, and perspectives involved in bidirectionally responsive design [73, 74], this study extends this work by providing detailed analysis of one case study of a bidirectionally responsive project produced during the second iteration of the Maker Innovator workshop (the first iteration of the workshop is documented in previous studies [73, 74]). This study also provides a framework to analyze computational concepts, practices, and perspectives in a bidirectionally responsive project adding to, building on, and integrating previous analytic approaches to looking at Scratch coding, and the design of physical and digital artifacts. It further investigates our understanding of computational thinking in physical fabrication and digital media design and coding more generally.

We explore two major research questions to this aim: (1) How does designing integrated systems, as facilitated through digital games and physical controllers—using e-textiles, the Lilypad Arduino, ModKit, Makey Makey, and Scratch—influence the complexity with which students engage in and understand computational thinking? and (2) How does the process of engaging in bidirectionally responsive design impact learners’ meaning making with and application of computational concepts?

2 BACKGROUND**2.1 Making and Computing Education**

Making and maker activities, while a relatively new phenomenon, have been thought to hold promise for “playful, but rich, engineering and design activities [in] K-12 education” [50, p. 31]. Engineering and computer science education at the collegiate and precollegiate level have been undergoing extensive reconsideration and development to respond to increasingly interactive and responsive technologies and their related pedagogies. In the former case, higher education is taking on the challenge of making curricula interdisciplinary to address the pedagogies necessary for embedded and hybrid systems design [39, 90]; in the latter case, scholars are addressing the need for frameworks and standards related to the emerging field of K-12 engineering education [55]. In fact, many of the challenges that STEM education is seeking to address are actually a result of moving away from tinkering and making sensibilities fostered through shop classes in the early part of the 20th century to more theoretical and didactic classes in more recent decades [6].

At the same time that formal education is addressing structural and pedagogical needs for 21st-century learning, in informal learning environments, the maker movement—infused with increasingly accessible digital toolkits and physical fabrication kits—is broadening access to technological literacies, efficacies, and creativity [30, 32, 50]. In fact, many have asserted that the increasingly standards-based structure of formal schooling discourages scientific curiosity, creativity, and inventiveness, and drives students away from science [32]. It has been proposed that makerspaces, maker literacies, and their associated accessible toolkits are reinvigorating precollegiate interest in STEM, creativity, and design [32, 50].

In many ways, “making” is a newer term for a longer trajectory of practices and ideas. As discussed in detail by Blikstein [6], the maker ethos is partially a return to the fabrication and design sensibilities inherent in early 20th-century K-12 engineering education, but also partially a new application of constructionism [61], or the concept that digital and physical toolkits can serve as “object[s]-to-think-with” (p. 11) and their resulting “constructions in the world [can serve] as a support for those in the head” (p. 142). “Maker” spaces and supporting informal learning spaces have grown out of do-it-yourself (DIY) sensibilities [38]. The Maker and DIY movements “disrupt” [36] traditional learning—perceptually and materially—but also have a reflexive nature with it: as lower-threshold and transparent learning materials are designed, the extensibility of DIY is expanded. As tools became more user friendly and proliferated through magazines, such as *Make* magazine, the activities attracted a wider audience and took on the name “making” in response to both its broader lure and multidisciplinary applications [20].

However, novice-accessible design needs to consider not only the tools themselves but the entry points to learning provided through the learning activities. In other words, effective scientific thinking involves the use of transparent tools, which learners can use to not only think and experiment with but also to learn how to design in support of that inquiry process [68]. Moreover, the curriculum should provide scaffolds and stepping stones to enable transfer and be widely accessible [67]. Making has been attractive for many reasons but primarily because it can nurture a so-called “maker mindset,” [20] by fostering playfulness and creativity, a growth-oriented belief around intelligence, appreciating productive failure as integral to tinkering and learning, and valuing collaboration as integral to design [50]. This mindset is further seen as cultivated through increasingly novice-friendly digital toolkits and digital fabrication tools. Furthermore, it helps promote computational thinking and a fluency approach, which are replacing a more antiquated skills-based approach that cannot keep up with rapid technological changes [6].

2.1.1 Critiques of Making: Implementation, Learning Outcomes and Inclusion. Despite its widely touted proposed benefits for informal learning, scholars have expressed skepticism regarding whether making contributes to meaningful or transferable learning. For example, a recent review of the literature found that more research is needed to demonstrate how differently designed maker activities and tools contribute to classroom learning (e.g., [60]). Moreover, while some have suggested that making can include a wide array of applications—including art, cooking, and sewing [20]—others have found that technology and computing-related applications and examples are overemphasized [8]. Martin [50] suggests that focusing on maker tools is easier for formal educational implementation but, in a similar vein to the approach taken to computing in schools, “a tool-centric approach to integrating making into education will certainly fail, as it will neglect the critical elements of community and mindset” (p. 37).

Further, scholars have noted that many equity and inclusion barriers exist in maker activities—particularly for women and girls, racial and ethnic minorities, and learners with dis/abilities—in a similar vein to other computing and technology-oriented domains [7, 10, 11, 79]. There have been several suggestions for addressing historical equity barriers in making and informal

computing activities, such as purposeful recruitment, and culturally situated or critical activities [75, 79]. Moreover, scholars such as Kafai and Peppler [38] argue that producing and creating with new media and participatory culture is crucial to engaging not only in inventiveness but critical understanding of media design. Others have suggested that the “material affordances” of different tools and their implicit associations can encourage equity [73]; for example, electronic textiles can be instrumental for the inclusion, agency, and engagement of diverse participants interested in crafting [10, 36, 75], and women and girls for whom crafting and sewing are often stereotyped interest areas [11]. In addition, certain techniques have been found to be beneficial for recruitment and inclusion of nondominant learners in technology-oriented informal environments, such as representing diverse and counter-stereotypical participants and experts on recruitment materials and as mentors in learning activities, encouraging collaborative learning on teams with mixed expertise, and enabling the creation of personally meaningful projects. Many of these suggestions are instrumental for equitable, inclusive, and accessible participation across gender, race/ethnicity, and dis/ability in making [80] and are thus intersectional. Scholars further advise that meaningful integration of these activities is important for successful outcomes [72, 75].

2.2 Digital and Physical Fabrication

Scholars have grappled with different tools and activities to teach making, physical and digital fabrication, design, and computational thinking to novice and youth learners, from LOGO to the integration of modern digital and physical fabrication toolkits [5, 53, 61, 69, 70, 82, 83, 89, 96]. In recent years, new tools have extended these activities to incorporate sewing, crafting, and textile-based wearable computing [11, 36, 37, 64, 66]. Constructionist gaming [34] has also expanded through digital construction kits and DIY sharing environments, such as Scratch, Alice, and Kodu [29, 70, 94]. Some scholars have endeavored to teach novice and youth learners how to blend digital and physical designs through “bifocal modeling” [5], “hybrid crafting” [30], and the creation of wearable game controllers [45, 92]. With one notable exception [73, 74], most of these efforts have focused on one-way responsiveness, often with sensors in the physical world triggering digital or physical interaction, instead of both concurrently.

2.2.1 Bidirectionally Responsive Design. Bidirectionally responsive design (BRD) [73], or bidirectionally responsive making [72] models complex embedded and hybrid systems design, is found in high-level engineering and computing applications but uses simplified digital and physical toolkits to fabricate simultaneously digitally and physically responsive systems. In particular, BRD creates a connection such that triggering a sensor in the physical world would produce an interaction on the screen and in the physical environment or artifact at the same time. The dual feedback is both simultaneous and recurring. In this sense, learners would be able to fabricate interactive systems akin to the ones that they have become familiar with in their lives, such as gaming controllers, which contain physical feedback in the form of lights and vibration [73]. Learners can become engaged in multimodal forms of technological expression and multiple forms of computational thinking, such as those evidenced through the design of e-textiles [36, 37], coding and media design [9], plus other fabricated interfaces.

Using the design of complex embedded and hybrid systems for learning is not a new concept. Before the maker movement became categorized as such, many designers, educators, and researchers were exploring similar learning affordances through physical computing [71]. Physical computing is the use of simplified and hobbyist technologies, digital interfaces, and sensors—typically by artists, designers, tinkerers, and other nonengineers—to create interactive physical and digital systems [59]. In fact, physical-computing scholars and practitioners were instrumental in creating Arduino [42], a low-cost microcontroller and open-source integrated development

environment (IDE), which has become the backbone for many computational maker activities. Historically, many of the constructionist development tools, such as LOGO [61] and, later, PicoCrickets, [69] and TangleBits [91], for example, attempted to provide easy access to the development of interactive systems. However, most of these tools and systems - with the exception of TangleBits and a few short-lived designs [96] - remained focused on robotics over other interactive applications.

2.2.2 Complexity and Complex Systems. Complex systems consist of “members, elements or agents” that interact with each other and the environment and have the capacity to generate new collective behavior [23, p. 793]. However, complexity still remains ill-defined due to its intricacy [88]. Some scholars have likened connections between general systems theory and complexity theory, positing that “both theories emphasize a shift in thinking from seeing parts to seeing the organisation of parts, recognizing that the interactions of those parts are not static and constant, but dynamic processes” [16, p. 112]. A unifying framework among disparate viewpoints and approaches lies in understanding its interdisciplinarity, which serves as an outgrowth of its recognition of multiple interacting actors and stems from some of the earliest attempts to understand complex systems [2].

Attention to integrated systems is often proposed in response to complexity in a variety of ways that include the organizational and design process [81]. Multiple complex systems—such as distributed networks, virtual environments, computing systems, and hardware infrastructures—introduce new considerations for integration, including how different specialty groups, often across disciplinary lines, address a variety of emergent and dynamic yet also interrelated interactions across systems [48, 81]. This is proposed to be of particular importance across computer science and engineering disciplines that work on interconnected systems to address interconnected problems. For example, computer science and engineering have used terms such as *embedded systems design* [44], *hybrid design* and *cyber-physical systems* [43] to describe what many in human-computer interaction (HCI) designate as tangible design [94] or physical computing [59]. Each terminology refers to the same underlying challenge and opportunity: designing systems that use physical and digital control and response mechanisms to provide unidirectional or dual feedback (bidirectional) interactivity. Take, for example, these two definitions—the first of tangible user interfaces (in HCI) and the second of embedded systems (in computer science and engineering):

Tangible user interfaces (TUIs) couple physical representations (e.g., spatially manipulable physical objects) with digital representations (e.g., graphics and audio), yielding user interfaces that are computationally mediated but generally not identifiable as “computers” per se. [94, pp. 916–917]

Although they are similar to other computing systems because they have software, hardware, and an environment, embedded systems differ in an essential way: Since they involve computation that is subject to physical constraints, the powerful separation of computation (software) from physicality (platform and environment)—traditionally, a central enabling concept in computer science—does not work for embedded systems. Instead, embedded systems design requires a more holistic approach that integrates essential paradigms from hardware and software design and control theory. [31, p. 33]

The main difference between definitions are their foci: tangible design orientations focus on the opportunities provided through creating integrated systems with relatively simplified tools, whereas embedded systems orientations focus on the challenges for interdisciplinary education and development with the underlying software and hardware requirements.

Within computer science and engineering, in particular, disciplinary emphases have made it difficult to come up with a model for teaching embedded systems design. For example, computer science has traditionally ignored embedded systems design [31], and the requirements for the creation of embedded systems require expertise beyond that emphasized in electrical engineering: “the embedded systems design problem... requires building a new scientific foundation that will systematically and even-handedly integrate computation and physicality from the bottom up” (p. 32) [31]. On the other hand, HCI—which incorporates computer science, cognitive science, psychology, design, and many other related disciplines—has embraced tangible design as an important area for further opportunities in light of concurrent emerging technologies. For example, reflecting on new innovations in architecture and virtual reality, Streitz and colleagues proposed that “combining [mental space, information space, social space and architectural space] in an integrated design allows us to develop enabling interfaces that build on the best affordances of everyday reality and virtuality in parallel” [87, p. 561].

2.2.3 Integrated Design and Collaboration. Engineering research has emphasized three major thrusts in the study of complexity: design and product development, manufacturing processes and systems, and the global design and management chain [23]. Specific to this study, two areas of emphasis for complexity are important: (1) product design and development and (2) collaborative teamwork. Engineering design introduces several areas of complexity through which several variables that are directly and indirectly controlled by designers interact [88]. For example, engineers start with problems to be solved through design, engage in design processes in service of these problems, and create a product to meet the needs outlined around the problem. Throughout this process, designers engage their background knowledge (conceptual knowledge) and modify the problem or product through design tasks (procedural knowledge).

However, design complexity is not limited to how multifarious a problem, process, or product ends up being or how interdependent each is on the whole. Increasingly complicated and interdisciplinary demands, such as those introduced by integrated systems, require that more attention be paid to different composite parts and actors within the design process. Drawing on a body of work from engineering, the social sciences and, in particular, computer-supported collaborative work, Lu and colleagues [46] provide a holistic definition of collaborative engineering as “a human-centered socio-technical activity, consisting of a dynamic co-construction process to maximize the synergy between the technical task-work by individuals and the social teamwork by groups,” which further involves “the collaborative exchange of resources among a team of stakeholders focused on a technology-intensive project with some overall common and creative purposes” (p. 617). Many aspects of this view of collaboration have been influenced by research and theories from the learning sciences and computer-supported collaborative learning, which emphasize distributed and group-level meaning making [86]. Much of this body of work has focused on social learning models and recognizes how disciplinary communities influence practices and outcomes. Integrating this body of work, Lu et al. [46] make a distinction between coordination (linking a collective set of tasks or activities), cooperation (working in common with shared activities or resources) and collaboration (furthering coordination and cooperation through a mutually shared goal and collective results that could not be achieved alone). They propose that engagement in collaborative decision-making by each stakeholder in the various interrelated and hierarchical aspects of the design process positively benefits organizational outcomes.

2.2.4 BRD, Complexity, and Collaboration. We propose that BRD models authenticate collaborative design practices and approaches that are responsive and adaptive to a variety of complex and interdisciplinary considerations. By using simplified digital and physical toolkits

in an integrated and collaborative way, learners are challenged to engage in considerations that would extend beyond the use of the tools themselves. They learn how to engage in the “assemblages of components that are themselves significantly complex” [48, p. 269], for example, a “system-of-systems” approach. In this approach, they must consider the intricacies of each system and its parts (e.g., a digital game as a whole, with the sprites and coding blocks as its parts), along with composite and distributed arrangements important for the design process and outcome of the integrated system as a whole (e.g., collaboratively distributing expertise of each system and engaging in the construction of a complex integrated system-of-systems with its own affordances, limitations, and considerations for the unified whole).

2.3 Multimodal Learning

Multimodal learning, a term that comes from the instructional design literature, captures the effects of engagement with different methods, modes, tools, and processes of learning [41, 51]. It also focuses on the integration of various multimedia and hypermedia tools in ways that provide students with opportunities to learn (e.g., [57, 62, 78]). Research shows that individuals retain information and understand concepts deeply, improve attention, and improve learning performance when information is presented in different modes, such as through visual representations, audio, and text [13, 26, 57]. Findings from the past 20 years have debunked the cone of experience model [14], which assumed various retention proportions for different modes of learning [26]. Instead, research has shown that using multiple modalities to relay information can have positive effects on student learning, metacognition, and agency. For example, multimodal learning has facilitated learning through computer-based environments and stimulated metacognition in nontraditional learners [56, 78]. Multimodal learning has also accommodated various learning processes and allowed students to become more self-directed [65]. For example, studies demonstrate that multimodal representations can assist student agency by allowing learners to select objects, tools, and representations that best suit their preferences [19]. Multimodal learning theory has been applied to learning with multimedia and simulations [1, 62], language learning [76, 77], formal and informal classrooms [33, 41], and even machine and neural robot learning [58, 85, 93].

However, scholars have cautioned against assumptions in similar learning theories that presume optimal learning somehow works best with minimal instruction [40]. With its strong emphasis on using technology to design instructional materials for which learning can become individualized, multimodal learning theory runs the risk of falling into this trap. Instead, guided instruction, particularly with highly complex learning material, even for students with prior knowledge, is strongly advised as a core instructional strategy [40].

Until recently, research has focused on screen-based multimedia representations, but emerging work explores the affordances of adding physicality in different ways. Smith [84] found that integrating 3D paper designs with 2D digital fabrication at different points in the curriculum aided language arts content learning and creative writing. Anderton et al. [3] employed multiple representations and activities when teaching an undergraduate anatomy course, which included traditional teaching instruments and technologies (anatomic dissections, models, and textbooks) and active-learning techniques (anatomic modeling and body painting, online multiple-choice assessments, and laboratory work). While they did not use each of these at the same time, they found that multimodal teaching strategies increased students’ grades and course confidence. Even though neither of these studies used an integrated systems approach such as BRD, they focused on the importance of multimodality in instructional processes that employ different aspects of guided instruction and project-based learning through physical representations (i.e., 3D paper designs and anatomic modeling).

2.3.1 Multimodal Learning with Physical and Digital Interaction. More recently, scholars have argued for multimodal learning that extends beyond multimedia and screen-based applications, without specifically citing or integrating multimodal learning theory. Some have proposed the use of digital and physical models and fabrication tools (e.g., bifocal modeling [5]), and tangible and full-body interfaces [22] that combine computational ideas with physical artifacts, tools, and materials. Bifocal modeling emphasizes “an approach to science pedagogy that challenges students to build and compare physical and virtual models in real time” (p. 296) with digital and physical toolkits (in this case with NetLOGO and the GoGo Board) [5]. The “hybrid crafting” approach [30] could help novices create tangible interfaces through the design of systems that can be easily integrated.

Both approaches are distinctly different from bidirectionally responsive design. Bifocal modeling focuses almost exclusively on formal and informal K-12 science learning, emphasizing end designs that are unidirectional (i.e., the physical inputs control the virtual representations) [73, 74]. Further, while there have been calls to incorporate tangible interfaces in learning, many of these are previously designed for learners (including the systems under development for hybrid crafting) for which there are only limited ways that they can implement personally meaningful designs. BRD is not system dependent, per se; it explores the possibilities and affordances for bidirectional design in currently available tools so that learners can construct their own integrated, multi-interface systems.

2.3.2 Bidirectionally Responsive Design as Extending Multimodal Learning. We propose that BRD adds an additional mode of representation within multimodal learning theory by operationalizing and applying both tangible interaction and bidirectional responsiveness. When implemented as intended (i.e., following constructionist models of creating personally meaningful, project-based learning within a guided instruction curriculum), it further integrates effective instructional strategies from multimodal learning theory. In the BRD maker workshop, digital multimedia presentations of concepts plus physical and digital buildable models of applied content knowledge were integrated with digital and physical fabrication tools (Scratch, Modkit, Lilypad Arduino and Makey Makey) separately, then incorporated. This was followed by design thinking for ideation and negotiation of personally meaningful BRD projects. Richard’s multimodal system of instruction and multimodal buildable models of embedded integrated systems prototypes (i.e., physically and digitally responsive designs) helped students both *learn* the underlying concepts and *apply* content-specific knowledge on circuitry, electricity, crafting, and fabrication in an incremental then integrated fashion. Even after the brief multimodal presentations of material, the main instructor (Richard) and 2 course assistants¹ provided just-in-time guidance. Further, learners would extend their understanding through the creation of their own embedded systems, first by understanding the underlying modalities (or affordances) of the individual tools and then by collaboratively integrating their individual contributions to the collective design.

2.4 Computational Thinking

Computational thinking is a “universally applicable attitude ... [that] involves solving problems, designing systems, and understanding human behavior, by drawing on the concepts fundamental to computer science” [95, p. 33]. It is further defined as thinking that involves troubleshooting, iteration, modularization, incrementality, abstraction, analysis, and interpretation.

2.4.1 Framing and Analyzing Computational Thinking through Scratch. Computational thinking has been operationalized by Brennan and Resnick [9] and Maloney and colleagues [49] through

¹See acknowledgments.

frameworks to analyze Scratch projects and trace student learning and development. Maloney and colleagues' framework focused on what Brennan and Resnick [9] later described as computational concepts, which refer to the programming structures apparent in most coding environments. Centering on the application of those concepts, they collected 536 Scratch programs created by students at a Computer Clubhouse and looked at diverse learners' use of computational concepts in Scratch over time through a frequency analysis. Their analysis concentrated on seven computational concepts—interaction loops, conditional statements, communication and synchronization, Boolean logic, variables, and random numbers—finding that five of them (variables, communication and synchronization, loops, Boolean logic, and random numbers) showed significant increases over the course of a school year.

Fields and colleagues [27] built on the work of Maloney et al. [49] and targeted their analysis on the number of computational concepts used by a random sample of 5000 Scratch users over a three-month period. They looked at the amount and complexity of computational concepts used in relation to the students' level of participation, gender, and account length. They further narrowed down Maloney and colleagues' computational concepts to focus on loops, Booleans, operators, broadcast commands, and variables. Based on their longitudinal analysis of Scratch projects, they identified four Scratch programming profiles (beginner, intermediate, advanced, and experienced) based on the knowledge and frequency of use of computational concepts. They identified beginners as users who created small projects using only loops (such as the forever loop prevalent in basic Scratch projects for continuous interaction); intermediate users as creating midsize projects using all computational concepts except Booleans; advanced users as creating midsize projects including Booleans; and experienced users as creating large, complex projects that incorporated Booleans. While they reflected that the differences between advanced and experienced users may not represent programming sophistication, experienced users were more likely to be high networkers on the site and were also the smallest group, making up about 10% of all users, whereas other programming profiles were equally distributed among the majority of Scratcher members.

Fields et al. [28] noted that frequency analysis alone could not assess the learning of computational concepts. Using mixed methods, Pantic et al. [63] qualitatively documented incremental changes among different learners and then quantitatively measured the changes in application of computational concepts, focusing on initialization (a computational concept that they felt other studies overlooked), events, and parallelism. Their findings suggest that the implementation of both qualitative and quantitative measures at frequent intervals provided a better understanding of "key moments" of the learning process, how learners applied the concepts that they learned through instruction, and the ways that each of their projects influenced learning patterns at camp (i.e., by remixing other campers' appealing projects).

Brennan and Resnick's framework [9] extended previous ones by looking beyond computational concepts. They incorporated two additional computational thinking categories: (1) computational practices, which refer to processes around coding and construction, such as troubleshooting or remixing; and (2) computational perspectives, which describe how students connect and value computational practices and computation in their lives. They further developed three approaches to assess these dimensions of computational thinking, which consisted of project portfolio analyses, artifact-based interviews, and design scenarios. Dasgupta et al. [15] further refined Brennan and Resnick's computational concepts as encompassing loops, parallelism, events, conditionals, operators, and data (they omitted sequencing, since the Scratch coding block infrastructure enforces a natural sequencing). They found that remixing is an important pathway to learning coding, as it has a significant positive relationship with increased understanding, command, and use of all computational concepts.

2.4.2 Extending Computational Thinking Analysis through E-Textiles and Circuits. The expanded framework of computational thinking [9] has been further applied to understanding learning with electronic textiles (e-textiles). The design of interactive textile-based or wearable systems through circuitry (popularly termed *e-textiles*) has become more novice accessible with the development of the Lilypad Arduino [11], which works like the Arduino but has been modified to be integrated with conductive thread and fabric instead of breadboards and wires. Kafai and colleagues [37] looked at students' circuit designs, Arduino code (used to program the Lilypad Arduino), and final e-textile artifacts to assess learners' understanding of circuitry, digital inputs and outputs, and computational concepts. Observational analysis was used to understand learners' design practices or how they conceptualized and designed their e-textile artifacts. Kafai et al. [37] deviated from using Brennan and Resnick's dimensions of computational perspectives (expressing, connecting and questioning) to analyze changing computational "perceptions" among students through interviews. They found that after making their e-textile projects, learners had a greater appreciation for computing and its relevance to their lives and careers, with many expressing positive interest in computer science. They also learned about the diversity of computing as a field to include tangible components and tinkering beyond the digital screen.

2.4.3 Limitations of Computational Thinking Frameworks and Analyses. We note limitations in past research that has explored computational thinking in Scratch and e-textiles. Kafai et al. [37], along with others [15, 28], noted that just using, remixing, or editing various coding blocks does not necessarily demonstrate learning or understanding of computational concepts. However, many of these past studies, despite the recognition of these limitations, still contained these as measures of learning. They note that without a detailed analysis of the learning context or students' explanations for using certain coding concepts, we cannot conclude that students have understood and learned how to apply various computational concepts. While remixing has shown a significant and positive relationship to the sophistication of use of computational concepts, many studies lack details regarding which projects or artifacts may have employed remixed code and to what extent it impacted computational thinking practices or perspectives. Furthermore, while Kafai and colleagues [37] provide an excellent guide for analyzing computational thinking with e-textiles, including diagrams of circuit designs, learners' logic behind why and how they connected various digital inputs/outputs, conductive thread, and fabric was not explored.

Exploring learners' iterative process of creating and connecting various components would help provide additional evidence of their understanding of computational concepts and provide further insight into learners' computational practices. Moreover, we propose that a bidirectionally responsive, multimodal approach may help gain deeper insights into how students think about and apply computational concepts and practices in two different contexts simultaneously: digital media development through Scratch and physical fabrication through e-textiles.

3 METHODS

3.1 Setting and Participants

3.1.1 The Mini-Course and Curriculum. The setting for learning activities was an elective mini-course titled "Maker Innovators" offered in 2015 through a public magnet high school in a large city in the northeastern part of the United States as part of an out-of-school learning requirement. Once a week, the 9th grade students were required to travel to a partnering informal science institution a few blocks away to attend their chosen mini-course (among dozens of science or technology learning opportunities that varied in content and structure) for up to 8 weeks. Each session lasted 1.5 hours during the school day (a total of 2 hours was designated, which included travel). This mini-course was structured as an informal experimental workshop to teach

Fig. 1. How bidirectionally responsive artifacts scaffold integrated systems design through purposefully connecting digital and physical construction kits. © Gabriela T. Richard.

bidirectional integrated systems design through the creation of wearable or e-textile-based games with digital and physical responsiveness.

The workshop was originally conceptualized, taught, and refined by Richard [73, 74] in 2014. She then designed and taught a second iteration in 2015. These two mini-courses ran for eight sessions each (a time limitation structured by the school and informal science institution) and taught learners how to combine Makey Makey [82] and Scratch [70], along with the Lilypad Arduino [11] and ModKit [4, 54] to create interactive games and wearable controllers with digitally and physically responsive elements.

Each toolkit was chosen not only for its material affordances but also for its ease of use and modality match (Figure 1). For example, ModKit is a visual, block-based coding environment graphically similar to Scratch, which was incorporated to lessen the cognitive demands that would occur when learning another coding environment, particularly in a short amount of time² [74]. The Lilypad Arduino is a microprocessor developed from the open-source Arduino platform, which facilitates processes (e.g., reading sensor values and controlling actuators, such as LEDs) and is specifically tailored to create e-textiles and interactive wearables. Makey Makey is a toolkit that simplifies physical computing by making anything conductive into a plug-and-play interface.

In this study, we focus on the second of the two mini-courses. During the first mini-course (reported elsewhere [73, 74]), the learners were taught about each tool and then asked to design their own virtually interactive games in Scratch that would be controlled by wearable controllers with simultaneous physically responsive elements, using the Lilypad Arduino and actuators—such as LEDs, vibrators, and speakers—that played musical notes. Bidirectionally responsive projects employed conductive fabric to activate a response on the screen (through a Scratch game connected with Makey Makey) and in the textile controller (through the Lilypad Arduino), as programmed through Scratch and ModKit, respectively. Figure 1 illustrates how learners would use each system to create their designs. The first course operated as a proof of concept and focused on instruction followed by worked examples (without explicit models) from presentation slides and worksheets.

²The significance of these cognitive challenges was documented during the first mini-course, when the Arduino IDE—the integrated text-based, syntax-oriented coding environment for Arduino—was used for a short duration to show learners how to read and interpret sensor values through serial communication, which ModKit could not do, and then apply the values to their ModKit code.

In other words, the first course was anchored around problem-based learning. This instructional strategy was cognitively complex for students as a result of the inventive, experimental approach for which no related examples could be located publicly.

In part owing to this, during this prior mini-course and study, out of the six groups, one of the 4 final working prototypes was bidirectional. The other 3 were collaboratively created and imaginative unidirectional wearable games in which the wearable controllers manipulated a Scratch game without physical responsiveness. One project did not work as intended, which would have been a standalone e-textile design. However, the unidirectional designs were notable in their resourcefulness, attention to modality considerations, and creativity when including team members [73]. They were further noteworthy in the level of refinement team members dedicated to their respective design contributions. One team created a fighting game that was controlled by swords made from repurposed coat hangers, which would activate actions in a digital fighting game they designed in which a player's "sword" made contact with conductive patches on the other player's wearable vest controller. They were deliberate in creating a fighting game because it seemed most like a "very small simulator" (p. 84) [73] that would imbue the affordances of embodied interaction. They originally sought to create a bidirectional design but were unable to do so primarily because of time limitations. Team members adjusted for these time constraints by taking on new dedicated roles, with one crafting the physical controllers, another designing the digital game characters, and the third coding in Scratch. This resulted in downtime, with one or two at a given time observing while another completed a task. Nevertheless, given how well they understood the affordances of the system (see [73]), it could be said that they effectively used this time to engage in peripheral learning.

In response to limitations noted in the first mini-course, the second course, which we report on here, was designed by Richard so that learners spent more time engaging with the material hands on by building models inspired by previously designed wearable and e-textile projects of increasing complexity and responsiveness (i.e., project-based). Table 1 provides details of the mini-course activities, and Figure 2 provides more specifics on the buildable models as provided through curricular handouts. The first author worked with a former graduate student to create "buildable models" based on the student-conceived designs from the first workshop and idealized designs created by the first author that would help students understand BRD.

While learners were encouraged to collaborate throughout, during the first five weeks, the instructors (Richard and course assistants) ensured that each learner received exposure to the concepts and procedures behind each toolkit and design process through project-based activities that involved reinforcing one aspect of the bidirectional process. The first project (buildable model 1) involved creating a simple glove controller, with conductive fabric, which would work with Makey Makey to control a simple Scratch game. The instructors first taught the learners how a simple space key or mouse click interaction could be controlled by Makey Makey with a Flappy Bird game, recreated in Scratch. Using a Scratch game allowed learners to first play, then see the code, and finally remix it to further their understanding. They then extended the glove's design (buildable model 1, part 2) by adding a piece of conductive fabric to the palm to make it work collaboratively with other learners' gloves. This allowed learners to experiment with different kinds of interactive experiences and design considerations. Afterward, they further explored the functionality of their gloves with other games on Scratch, followed by remixing those games. Next, they created games or projects of their own design in Scratch. The next project (buildable model 2) was a more complex yet traditional standalone e-textile design (i.e., no digital interfacing), which involved creating a necklace (based on Iron Man's Arc Reactor) that would change color depending on how they programmed the Tricolor LED connected to the Lilypad (with ModKit). During this process, they also engaged in e-textile circuitry and sewing. The final buildable model (3) was a glove controller that

Table 1. Curriculum of the Second BRD Mini-Course (2015), a Project-Based Iteration on the First Instantiation in 2014

Day	Topic	Activities
1	Introduction to the mini-course and concepts	Learners were introduced to the course theme (being “Maker Innovators”) by Richard through presentation slides. They looked at YouTube videos of inspirational e-textile projects and then games created with Scratch and Makey Makey (i.e., unidirectional designs [73]) and were asked to explore the possibilities of putting both e-textile designs and Makey Makey games together with prompts from the instructor (20 min). They were introduced to how a circuit works and shown an animated explanatory video (10 min), then applied their knowledge through creating buildable model 1 (60 min).
2	From remixing to designing in Scratch	Learners continued buildable model 1 by repurposing the design to create collaborative gloves (BM1, part 2). Then, they remixed games in Scratch that would work with the glove, exploring its functionality. They were given example Scratch projects that they could create through handouts with sample code and spent the remaining time creating or remixing (1.5 hr).
3	Intro to e-textiles (activity delayed)	Learners were introduced to e-textiles and how to program the Lilypad Arduino with ModKit. The informal science institution staff forgot to provide laptops that could run ModKit as planned; thus, the activity was tabled to lesson 4. Learners instead continued to design and code in Scratch (1.5 hr).
4	Basic e-textile activity	Learners were reintroduced to e-textiles and started making buildable model 2, a standalone e-textile project (5 min). They used ModKit to program a Tricolor LED and sewed the complete design together (60 min). Afterwards, they were introduced to advanced functions, such as reading in simple switches created by touching two conductive pieces of fabric together (25 min).
5	Advanced e-textiles, bidirectional design, and design thinking	Owing to technical problems during lesson 3, there was not enough time to complete buildable model 3. Instead, they integrated the pieces of conductive fabric (sensors) and actuators (LEDs, speaker, and buzzer) with alligator clips to Makey Makey and Lilypad Arduino and started programming the connections to see how it would all work together (45 min). Then, they engaged in a design thinking exercise (15 min), followed by storyboarding their ideas (15 min). Finally, they negotiated those ideas with classmates of their choosing, and picked a final project design (15 min).
6-8	Final project work and design showcase	The last 3 classes involved collaborative design. Each team had 2–3 members, and each member self-selected a different role on the team. They worked together to create their designs, using the handouts and calling on one of the three workshop facilitators for assistance when needed. At the end of session 8 (30 min), they demonstrated their projects to each other. Three weeks later, they showcased their designs to the entire 9th grade class as part of an open 30-minute event supported by the school.

would make sounds, light up an LED, and vibrate as well as control elements of a Scratch game depending on which of the pieces of conductive fabric or tape were touched. Learners would then be provided with different example games that they could try their glove controller on based on their functionality match with the 3 discrete (on/off) conductive touch points on their gloves but ultimately could design or remix Scratch game-s of their choosing. Due to a coordination error with the informal science institution hosting the mini-courses, laptops were not provided when

scheduled during session 3 (see Table 1). The students needed the loaner laptops because the school provided only Chromebooks, which could not run downloadable programs such as ModKit. As a result, remaining activities, which relied on ModKit, were postponed by a week and the students were able to start prototyping the interactions of buildable model 3 with Makey Makey, the Lilypad Arduino, and Scratch, though they did not have time to fabricate the final product.

Before creating their final projects, learners engaged in a design thinking and storyboarding exercise. This exercise helped to reinforce how BRD—or bidirectionally responsive making—is not just about learning how to create the integrated systems themselves but also concerns the facilitation of learning how to craft an experience for another person, akin to participatory design [21]. Using design thinking as an anchor, they reflected on their own experiences using various interactive, multi-interface technologies, as well as on how their design ideas would impart immersive or evocative engagement for someone else. Thus, the design process could be understood by learners as less about adding lights, sounds, or vibration but instead involving deep reflection about the affordances of different toolkits and components for translating the experience that they want to create.

3.1.2 The Facilitators. Richard introduced the activities and was the primary workshop facilitator. Two master’s students³—one majoring in computer graphics and the other in interactive arts—were trained on the curriculum material, youth instruction, and data collection protocols by Richard. With her guidance, one helped design the buildable model curricular materials and the other led an activity on design thinking. Both assisted with workshop activities, including helping individual participants or groups with instructional material, along with Richard. Each participant was provided with detailed handouts on each activity and troubleshooting tips.

3.1.3 The Participants. Participants during the second mini-course, as with the first [73, 74], were diverse ninth grade youth from a public science magnet high school in the northeastern part of the United States, ranging from 14 to 15 years of age. Eighteen students (10 girls, 8 boys) were enrolled in the course; 17 completed all course activities (one girl left the school halfway through the mini-course). Each student created a final project in groups of 2 to 3. A total of 3 of the 6 final projects were fully BRDs, meaning that both the wearable or e-textile controller and the digital interface responded to sensors in the physical environment. The other 3 projects were unidirectional, meaning that they created games in which the wearable or textile designs controlled a Scratch game but did not provide physical interaction or feedback.

3.1.4 The Case Study Team. In our analysis, we focus on one case study team (see Section 3.2.1). The team consisted of three members⁴. On surveys and in interviews, Sadia (self-identified as Black/African American, female, age 15 years) and Andre (self-identified as Black/African American, male, age 15 years) stated that they did not have prior experience with the Lilypad Arduino, e-textiles, crafting, Scratch, or Makey Makey. However, Sadia had some limited knowledge of circuitry; Andre knew how to sew and went to a summer camp where he learned programming but “did not really learn a lot of stuff there.” Elisa (self-identified as Hispanic/Latina, female, age 14 years) had some experience in coding and circuit design through an afterschool robotics club.

3.1.5 Instruments and Protocols. Each participant completed pre- and postsurveys, and postinterviews. We used a survey derived from the computing attitudes scales, previously validated by Ericson and McKlin [24] and used in similar maker research (e.g., [37]). The survey contained Likert-scale items such as “I can become good at computing,” “I like the challenge of computing,”

³See acknowledgment.

⁴Names have been changed to protect confidentiality.

“I will use what I learned in this workshop,” and “I am interested in a career in computing.” Along with demographic questions (gender, age, race/ethnicity, and grade level), they were also asked how many hours they used computers at home, if at all, to do a variety of things, such as homework, learning, playing games, or chatting with friends. We added open-ended questions for fill-in answers, which included “What do you think you will learn during the workshop?” and “Have you ever [worked with Makey Makey, programming, engaged in crafting, etc.] before?” in the presurvey, and “What did you learn?,” “What did you like [best/least] about the workshop?,” and “How did you feel working with [Makey Makey, Scratch, etc.]?” in the postsurvey.

The interview was semi-structured and occurred 1 to 2 weeks after the workshop, depending on availability, with each participant individually. Each interview lasted 20 to 30 minutes, during which time participants answered questions, including “Did you have [programming, design, e-textiles, etc.] experience before the workshop?,” “Did you know about circuits before the workshop?,” “How did you feel [working with Makey Makey, Scratch, Modkit, Lilypad Arduino; making wearable electronics or e-textiles]?,” “Can you tell me about the project your team made (and describe it to me)?,” “Why did your team decide to make this project?,” “What did you have to learn in order to make it?,” “What was it like working with partners on the project?,” “How did you feel combining all of the tools together? Did you understand why we were doing that? Did it add to your design at all?,” and “Did the workshop make you feel differently about [programming, design, wearable technologies, careers, etc.]?” In addition, they were asked about their video game playing experience (since the workshop theme focused on making interactive games) and whether there was anything they would change or improve about the workshop.

Each class session was videotaped and audio recorded with one camera from a wide angle. One of the graduate student facilitators was tasked with taking the camera to closely examine an individual or group’s activities during each session every 10 to 15 minutes. We also collected their Scratch files by having them fill out their project numbers on a sheet each session and archiving a remixed copy. Because ModKit would only run on the institution-provided computers, the same graduate assistant or the lead facilitator took pictures of their coding process for documentation and analysis. Each session’s recording was viewed by the authors, and detailed field notes were taken about the learning interactions, including how the groups were positioned, their utterances (when audible), and their on-task and off-task actions.

3.2 Data Analysis

First, we reviewed the corpus of data as a whole and then we segmented out the digital and physical artifacts of final projects. We looked at their bidirectional (and multimodal) computational thinking concepts and practices as evidenced through content analyses of their digital (Scratch and Modkit codes) and physical artifacts (Lilypad Arduino and Makey Makey circuit designs). Audio interviews and video data were transcribed and analyzed by three team members using a deductive constant comparison approach [12] to look for themes related to computational practices and perspectives. We further used microanalytic techniques of video records and field notes of classroom interactions to understand student design practices and articulations of their design processes [17, 47]. Specifically, here, we identified salient moments using both analytic induction [17] and constant comparison techniques. We viewed the salient moments multiple times and wrote out thick descriptions, which were further analyzed until we had identified and triangulated themes among the data. This multimodal analysis also helps fill in the gaps noted in past research on computational thinking.

3.2.1 Case Study Rationale. Herein, we focus on a case study of one of the three bidirectional design project teams, as analyzed through the artifacts, interviews, and classroom observations,

to provide an in-depth multimodal analysis of the ways that learners engage in individual and distributed meaning making with and through BRD. We should note that this case study group demonstrated neither the most nor least cohesive of the bidirectional designs. Rather, through analytic induction, we found this group to provide salience among the themes that we saw across groups. In particular, this group's design process and final product evidenced strengths and weaknesses that help illustrate the importance of collaboration and distributed learning during the design of complex systems. We chose to focus on one team in order to (1) illustrate and describe the kinds of computational thinking involved as derived from design artifacts and observed design processes, and (2) explicate the ways that learners engage in understanding complex systems design through a form of authentic practice.

3.2.2 Computational Thinking Analysis. To analyze computational thinking in students' physical artifacts, we used Brennan and Resnick's [9] more holistic framework of computational thinking (concepts, practices, and perspectives). We looked at video data showing students' iterative design process, student interviews, and analytic memos for evidence of the three dimensions of computational thinking as highlighted by Brennan and Resnick. Similarly, drawing on Kafai and colleagues' analysis of e-textile projects [37], we diagrammed students' final circuit designs to look at their wiring, sewing, knowledge of inputs and outputs, and digital and analog pins. Computational thinking is not only about being mindful about computational concepts and implementing them in coding platforms such as Scratch and Modkit; it is also thinking about operational circuit design when sewing and connecting e-textiles. Conductive thread is a replacement for wires, which enables seamless, embedded circuit design. However, its lack of shielding makes it simultaneously susceptible to short-circuiting when not properly laid out and fabricated. As such, circuit designs—as artifacts—helped provide evidence of learners' ability to apply computational concepts and practices. Furthermore, Makey Makey, as a plug-and-play physical computing prototyping tool, is typified by the use of alligator clipping [82]. While this can make for a messy design, it can also rapidly aid in the understanding of circuits and tangible design for novice learners. Within the time constraints of the mini-course, we did not require a fully fabricated and stitched design, which would have involved embedding Makey Makey within the e-textile along with Lilypad. Instead, we explore learners' understanding of proper circuitry through functional integration of Makey Makey in order to control their Scratch games.

In order to analyze circuit design for evidence of understanding through applying computational concepts and practices, we built on the analysis of e-textiles by Kafai et al. [37]. First, we looked at connection, which meant observing how the Lilypad Arduino, sensors (conductive fabric), and actuators (LEDs, speakers, and the like) were sewn together and how the alligator clips were connected to Makey Makey. We also looked at how learners sewed in or connected the correct digital input/output pins on the Lilypad Arduino to the conductive fabric, as well as whether the conductive thread overlapped (which would cause a short) or worked around the other traces. Functional e-textile circuitry is also an important aspect of computational thinking because it shows that students are actively engaging conceptual knowledge to create a working circuit. Third, we tested the final project to see if the design was operational. Playtesting helped illustrate whether a group had grasped computational concepts that they mentioned in interviews and observations.

3.2.3 Block Coding Analysis. We looked at all of the coding employed in each project. In this case, it meant analyzing both the object-oriented Scratch codes in the remixed copies and the pictures of the ModKit codes for each project. To analyze computational thinking in Scratch, we used Brennan and Resnick's framework to look at the computational concepts employed in students' projects and provided a frequency analysis, as employed by related work [27, 49]. However, note

Fig. 2. How the Dino Party game worked with the physical and digital artifact; (left) no interaction; (middle) touch point 1 activated; (right) touch point 2 activated.

that a frequency analysis may not provide an accurate picture of whether students learned the computational concepts and whether students engaged in computational practices. To provide a better analysis of students' understanding, we also looked at whether the codes were original or remixed. As Dasgupta et al. [15] point out, remixing can be an important pathway to learning programming and developing computational thinking.

We tried to emulate similar analysis for the Modkit coding images. However, due to documentation inconsistencies, we could not retrieve all of the relevant ModKit images for each project. To fill in the gaps in these instances, we tested the final projects to observe the hardware functionality and how it related to coding blocks that were most likely employed. In our case study example, we have most of the code used for the final project, which we expand on in our analysis.

3.2.4 Interview and Observation Analysis. Interviews with students were analyzed for individual and collaborative understanding of computational practices and concepts among the team. For example, we recognize that, while an artifact can provide representational evidence of this understanding, it does not, in and of itself, afford a complete picture of that understanding. We further complemented interviews with observational instances in which students explicitly engaged in computational practices, such as testing and debugging, or used computational concepts, such as programming loops, using interaction analyses of microgenetic systems (specific actions and design activities) and mesogenetic systems (activities over time) [25, 47]. Similarly, interviews were analyzed to understand the computational perspectives of students while engaging in creating their own bidirectional or unidirectional designs. In this regard, we incorporated Brennan and Resnick's [9] computational perspectives (expressing, connecting, and questioning) to look at learners' evolving ways of seeing and applying computing-inspired ideas in their lives. We also analyzed the interviews through the three dimensions of perceptions (relevance of computing, ability to see oneself as a computing professional, and perceptions of computing as a field) highlighted by Kafai et al. [37] in their interviews with youth.

3.2.5 Bidirectionally Responsive Project. *Dino party* (Figure 2) is a pet care game, created by the case study team, in which a Dinosaur sprite jumps, eats, and talks depending on discrete actions prompted through the wearable controller. The wearable was inspired by the design of

Fig. 3. Dino Party circuit diagram (left); physical artifact (right).

Fig. 4. Close-up of Dino Party physical artifact.

a watch, which could be large enough to accommodate the surface area of the Lilypad Arduino. As a result, it was creatively reconceptualized to attach to a player's hand. Depending on which of the conductive tape "buttons" were touched, the lights would change color on the wearable controller, while actions would simultaneously correspond in the digital game.

4 FINDINGS

4.1 Circuit Design of Physical Artifact

4.1.1 Computational Concepts: Circuitry and Integrated Systems. Dino Party can be understood as bidirectional from observing the physical artifact and circuit (Figures 3 and 4). The team designed the physical controller so that both Makey Makey and the Lilypad were connected. One end of a black alligator clip was connected to earth (ground) on Makey Makey and another end to a piece of conductive tape. One end of another black alligator clip was connected to ground on the Lilypad Arduino and the other end to the same piece of conductive foil. Thus, both were similarly representationally and functionally grounded. Further, the use of a black alligator clip helps to denote a common visual model of grounding used in circuitry. Correspondingly, they connected digital output pins 5 and 6 on the Lilypad to two pieces of conductive fabric and connected the spacebar and up arrow pins on Makey Makey to the corresponding two pieces of conductive fabric. By doing this, they were able to facilitate commands that would, for example, control a "when

What can be implied from Dino Party's circuit design is that the students evidenced conceptual and procedural understanding of embedded design through BRD. For example, they understood that the feedback had to be simultaneous—on screen and on the physical artifact. They also demonstrated understanding of e-textile circuitry by sewing around the conductive thread traces when connecting the microprocessor and components (i.e., not crossing unshielded traces).

Now, we turn our attention to the digital interface of their BRD project. Based on their Scratch code (Figure 5), when the space button is pressed, the Dinosaur goes to the apple. When the up arrow is pressed, the Dinosaur jumps up and down on the trampoline and exclaims in delight. When right arrow is pressed, the Dinosaur goes to the trampoline and thinks that it would be fun to play with.

ACM Transactions on Computing Education, Vol. 19, No. 3, Article 17. Publication date: January 2019.

Table 2. Dino Party Scratch Coding Frequency Analysis

Concepts	Measure	Scratch Blocks	Freq
Loops	Uses forever loops, etc.	forever, foreverIf, repeat, repeatUntil	0
Sequences	Series of events to final action/behavior	Go to, switch costume to, wait, say, change x by, etc.	10
Parallelism	Parallel scripts with same “hat” block	startHatTriggered, eventHatTriggered, etc.	0
Events	Uses “when <>” hat blocks	eventHatTriggered, keyHatTriggered, bounceOffEdge, touching, keyPressed, etc.	4
Conditionals	Uses conditional blocks (e.g., “if” block)	waitUntil, foreverIf, if, ifElse, repeatUntil, lessThan, equalTo, and, or, not, listContains, etc.	0
Operators	Uses operator blocks (e.g., “+” or “or” blocks)	lessThan, equalTo, greaterThan, and, or, not, add, subtract, multiply, divide, pickRandomFromTo, concatenateWith, letterOf, stringLength, etc.	0
Data	Uses data blocks (e.g., variable block)	setVarTo, changeVarBy, showVariable, readVariable, addToList, listContains, etc.	0

4.2.2 Computational Concepts: Integrated Systems. While the code by itself would work as intended, there were some missing elements that would have affected the connection with the physical artifact. There are two buttons but three reference points in the code (space, up arrow, right arrow). When playing with the design, Sadia and Elisa noted the problem but did not correct it before the end of the workshop. Instead, when demonstrating the project, they moved the trampoline sprite behind the apple so that the right arrow sequence of actions would not be needed (though they did not save this final change). We elaborate on this more in Section 4.4. These issues may represent some conceptual misunderstandings, which could be aided through more directed scaffolding or more time focused on certain aspects of coding. On the other hand, the design limitations could be explained by team dynamics (which we expand on in Section 4.4). This team was very collaborative and, though each member focused more specifically on one aspect of the design process, they often took turns. As a result, when compared with teams who evidenced more compartmentalized task distribution, they may have collectively understood and applied some aspects of coding or design over others.

4.3 ModKit Coding of Digital Artifact

4.3.1 Computational Concepts: Loops and Conditionals. Analysis of their ModKit code demonstrates that there was a better conceptual and procedural understanding of loops when coding microcontrollers, such as the Lilypad Arduino (Figure 6). In order for a microprocessor to continuously send and receive signals, the actions need to be contained within loops. The functional understanding of loops for coding microcontrollers could be due to one of two reasonable interpretations: (1) loops were more directly reinforced as a foundational aspect of microcontroller coding for which there was immediate feedback (i.e., the microcontroller would work with a loop or appear not to work without one because of how fast the signal could be sent once) or (2) it was a function of individual understanding because each group member focused more on one aspect of the final design.

Their code also demonstrates their understanding of conditionals in order to change the state of the Tricolor LED, which can blend colors and change color effects. They wanted to integrate purple for one of the color states in the physical environment; in doing so, they demonstrated their understanding of color blending with code (i.e., turning the foundational red, green, and blue

Fig. 6. Dino Party Modkit Code: (left) code with prototyped circuit; (right) close-up of code.

colors—each connected to an individual numbered pin—on or off to blend them). Figure 6 shows their ModKit code during one of its iterative coding phases when they were testing different color choices and functions. Their final design contained three color states (purple, blue, and red); purple was the default color, and the Tricolor LED would change to red or blue depending on which one of the two corresponding conductive points (“buttons”) was activated.

4.3.2 Computational Practices: Experimenting and Iterating. During their ModKit coding process, the team was observed collaboratively coding, play testing their designs, and making changes to coding blocks that did not result in intended interactions, which helps demonstrate that the team was engaging in experimenting and incremental design. It was in part through this process that they came to understand how to code the color blending functionality of the Tricolor LED.

4.4 Interviews and Observations as Reflecting Computational Thinking and Complex Systems Understanding

In this analysis, we examined computational concepts, practices, perspectives, and perceptions as discussed by learners during postinterviews and as they happened during specific design activities and over time when creating bidirectionally responsive projects. According to Brennan and Resnick’s framework, computational practices include experimenting and iterating (incremental design), testing and debugging (troubleshooting), reusing and remixing (building on an existing project or ideas), and abstracting and modularizing (exploring connections between the whole and its parts). Computational perspectives include expressing (seeing computation as creation), connecting (recognizing the power of creating with others), and questioning (feeling empowered to make sense of computing in their lives).

4.4.1 Computational Concepts and Practices. Class videos and individual semi-structured interviews helped reveal the roles that members engaged in during the project as well as their rationale for choosing the project, both of which were highly collaborative, albeit a bit challenging. For example, when asked why the team chose to make the project, Andre discussed how some games, such as *PacMan*, which another team remixed in Scratch for their design, were too complicated, whereas Elisa emphasized their indecisiveness. Sadia, on the other hand, provided an explanation that considered some of these complications but focused on the process as a whole:

Fig. 7. Demonstration of Dino Party by team. Left: Sadia points at the circuit design on Andre's hand as he demonstrates its functionality. Right: Elisa interjects to show that they used conductive tape as touch points.

Interviewer: Why did your team choose to make this project?

Sadia: At first, we were just—we didn't know what to do. We were just really clueless, because every idea we made there was, like, pros and cons. Then we just started listing ideas and then somebody came up with the dinosaur so we just went with that. Again, we listed the pros and cons and listed the challenges and stuff and then... Andre sewed, I made the glove and then Elisa programmed, but we alternated. One time somebody would switch, so we all got a fair chance of doing something.

Sadia took on the leadership and coordinator role on the team, reorienting her team to the collective design idea and contributing to different aspects of the design across roles. This is further emphasized through the video data of an instance when the team is asked to demonstrate their project and how it all worked together on the last day of the workshop. During this interaction, Sadia provides a holistic explanation of each role, while Andre models the final design and Elisa interjects to provide additional information (Figure 7). Before Andre starts the demonstration, Sadia points to the two interaction points and informs him, “this one is up arrow, this one is space.”

Sadia: At first, we thought of an idea for the game and we wanted, like, a physical thing so...first we were going to do a watch and then we made it a bigger thing than a watch, if it's an LED and the Lilypad. So, we thought of doing, like, a flower thing in the middle so we cut it out and [flips Andre's hand over to show the glove-like attachment to the conductive fabric] then there's a little thing that attaches to your fingers, so we could attach this and this [pointing to the Lilypad and the Tricolor LED] and we could sew it without intersecting the conductive fabric so it will be, like, really smooth...

Elisa: We didn't sew but we stuck those on... [pointing to the two pieces of conductive tape sewn to the Lilypad with the conductive thread]

Sadia: ...and then we connected it to [pointing to the Lilypad and LED]... we coded it and then it worked so [pointing to the two pieces of conductive tape and then to Makey Makey] we just connected it to the Makey Makey and we already had our game... [pointing to the screen with the Scratch code]

While the interaction helps reinforce her role in coordinating the design process, it also assists in illustrating that Sadia understood that for the Lilypad and LED to work together properly, the wires

Fig. 8. Elisa and Sadia discovering and discussing the design problem.

connecting the digital output pins from the Lilypad to the output pins in the Tricolor LED could not intersect and create a short circuit. In her interview, she further reinforces her understanding when asked to describe the project that she and her team made:

Okay, basically, we came up with a game on Scratch and then we made this little glove that connected with the game, so if you pressed the button on the fabric, it would activate whatever the game was. If you pressed a button on the fabric that we made with the Lilypad... it would light up and then if you press the space bar it would move and it would connect with the glove so that was pretty cool. It was like an intertwined connection.

Sadia is able to fully articulate the design process to make an artifact bidirectionally responsive. They did this by first connecting the Lilypad Arduino and the LED and then connecting digital output pins 5 and 6 to the two pieces of conductive tape by sewing the connective trace with the conductive thread. Then, they used two alligator clips to connect the space and up arrow button on Makey Makey to the same pieces of conductive tape. As a result, whenever the conductive sensors were touched, there would be dual feedback onscreen (where the dinosaur sprite would perform some action) that was simultaneous in the physical world (where the Tricolor LED colors would change). She not only expresses how their project works but demonstrates understanding of both the parts and the whole of the project (i.e., modularizing and abstracting).

However, as noted in Section 4.2, the team discovered a problem between their physical and digital interface. While the code referenced three interaction sequences (spacebar, left arrow, and right arrow), they had only two conductive action points on their wearable controller, which they discovered when discussing their final design with the instructor (Richard) during the last class session (Figure 8).

Sadia: It's cool because we have this... [she shows the wearable controller]

Elisa: [laughs and looks at Sadia. She covers her mouth, trying to conceal what she is saying as she points to Scratch] It's supposed to be jumping on the trampoline...

Sadia: Yeah, I don't know how to do the trampoline. [Elisa laughs and looks at the controller.] Unless we can move this over there... [Sadia moves the trampoline sprite behind the apple so that they will not need the additional action.]

Instructor: [moves closer] Let me see what's happening on the screen.

Elisa: He was supposed to move to the trampoline so, like, he could jump on the trampoline, but I don't know how so we're just gonna...

Sadia: [interjecting] So basically... yeah let's restart it. See now. [She demonstrates the actions now that the trampoline has been moved.] He eats the apple. He's

jumping on the trampoline supposedly... We can't move it over there.... [pointing to its original position]

The exchange helps to highlight the ways that they engaged in testing and debugging. Elisa had primarily taken on the coding role on the team, creating the Scratch code and the ModKit code, though Sadia and Andre assisted occasionally with the Scratch design. Elisa is able to immediately locate that there is a problem in the actions. However, since she was primarily involved in coding and not integrating the wearable controller, she seems to be at a loss when considering which part of the complex system has caused the design problem. While Sadia admits that she is not sure why it is not working correctly, having primarily worked on the controller design with Andre, she is able to apply her working knowledge of Scratch to provide a rational solution by moving the trampoline. This demonstrates some awareness of the limitations in the code and in the wearable interface that will interfere with it working properly. Though Elisa did not come up with the immediate solution, she does recognize how the location change will impact the code, being able to articulate it to the instructor as Sadia performed the action.

Computational thinking is not only about being mindful regarding computational concepts and implementing those computational concepts in coding platforms such as Scratch and Modkit. In our artifact analysis, for example, we saw evidence that the team created a functional e-textile circuit that was carefully crafted to complement their design aesthetics while also effectively mapping out the circuitry. However, that alone was not necessarily evidence of understanding.

The interviews and observations help to highlight how each member articulated understanding of the process (i.e., computational practices). For example, the team demonstrated instances of experimenting, iterating, testing, and debugging. When asked to discuss her experience with the wearable sensors and e-textiles, Elisa described how “it was kind of frustrating at first just to figure out what you were doing and why something wasn't working, but other than that it was good once you figured it out, like, why your wires were tangled.” While Sadia was more positive about the process, she similarly reflected on aspects of iterating: “I think that was kind of, like, the best part because we made a glove and then we made a—kind of like—a little fabric on your palm that connects to your fingers... [then] we coded it and we put the Lilypad on there and conductive fabric [and] we sewed it.” Sadia's reflection on the process not only revealed her understanding of each system and how to troubleshoot them to work together but also her shifting understanding of the design process:

We had to learn to work with Lilypad and how to code, and if there's a challenge, you just had to overcome it or an obstacle, you just have to compromise and value your teammate's opinions and try to include whatever they think, try to compromise as well. Don't just think of yourself. With coding, you have to be really patient because sometimes it won't work.

The team further reflected on and explained the troubleshooting process during circuit design and coding with the Lilypad Arduino. As illustrated in Figure 6, the team first created their circuit with alligator clips in order to make sure that the coding worked before sewing the circuits. Sadia explained that “you just had to make sure it would work before you actually sewed it [together].” During their circuit design and coding process, they first tested out their Modkit codes with alligator clips connecting sensors and actuators to the Lilypad. Then they incrementally added code to create the tangible responses that they wanted. Finally, they sewed the circuits together on their wearable controllers (Figure 9).

In interviews, team members also reflected on the complexity of creating a BRD. When asked to reflect on combining the different components (the Lilypad Arduino, Modkit, Makey Makey,

Fig. 9. Dino Party team members sewing together parts of their wearable interface and circuit.

and Scratch), while Sadia first felt it was all “sometimes overwhelming,” she was able to express how the iterative design process assisted with meeting their design goals: “you just look at it in separate parts [then] it’s just easier to process.” Elisa similarly communicated how complex the process was but how being deliberate became an aspect of it:

Elisa: [It] was kind of hard because you had to learn what’s what and you had to do all this stuff for one certain thing, but then you had to make it connect with the other.

Interviewer: How did you overcome that challenge?

Elisa: I don’t know. I just did everything really slow.

Andre, on the other hand, focused less on frustration and more on its intricacy. He was also thoughtful about the ways that the process of adding each of the technologies helped to amplify his contribution:

Andre: It was very busy because you had three different things going on at the same time, but it was fun to associate them together.

Interviewer: Did you understand why we were doing that? Why we’re combining all of those things?

Andre: To show how technology could work together.

Interviewer: Did it add to your design at all? To combine all these things?

Andre: Yeah, it made my wearable device associate better with the game.

While the complexity of the process was not immaterial, it is through their expressions that we come to understand how they each contributed to the whole and from different vantage points. On the one hand, Sadia is more expressive in breaking down the process; on the other hand, Elisa and Andre both indicate in different ways how they contributed to those incremental steps and, in Andre’s case, how purposeful that process was to his part of the design.

4.4.2 Computational Perspectives and Perceptions. Each team member, at different points during the design activity and afterward in interviews, conveyed computational perspectives related to connecting or seeing the value in creating with others. For example, Sadia was highly integrative of each member’s role in the design process, emphasizing what “we” were able to

do together throughout her interview, which was further reinforced when asked about her experience working with her team:

...they were my friends and we didn't really argue too much. I guess maybe one time somebody just didn't code right or something like that. We didn't really argue about it, but we just talked it out and just, "Okay, yeah, be careful next time." We were at least supportive of each other...

Elisa and Andre, who each took on more distinctive roles in the development process, reflected on other aspects of connecting. Elisa discussed that the team employed division of labor but that they helped each other out and checked out each other's work once one member was done: "We all did separate things...we all had our own job, so then we could see, 'Did you finish? Could you check out mine?'" Elisa is more specific in reflecting on the value of collaboration when asked what it was like to work with her partners:

It was a lot easier, because it's a lot...you have to do a bunch of small things. They're very detailed and you can't mess up on things because if you mess up one thing it messes up the whole thing. So, it was really good to have two sets of eyes just to make sure you're doing everything correctly.

In his response to the question, Andre is similarly reflective: "It was actually easier to me than working just by myself because we all had different ideas and we could combine our ideas." Andre further offers that each team member was "able to try different things" through this collaboration.

Perhaps due to their specific functions on the team, Elisa and Andre were more focused on how collaborating with others eased the integrated system complexity because they could reinforce their understanding of ideas or of their conceptual and procedural knowledge. Sadia focused more on the psychological safety and camaraderie of the group. Specifically, she was more intentional in connecting their social bonds with their task cohesion and completion. Observations revealed that, whereas Sadia and Andre were, in every instance recorded, seen working on the project, Elisa was more likely to wander off and socialize with learners on other teams. This could perhaps have been due to her very limited role in the design process (focusing on coding in ModKit and Scratch) or it could have been due to her lack of focus that the team encouraged her to take on these roles. Alternatively, her frustration with certain aspects of the process could have disengaged her. None of the team members provided insight in their interviews regarding this aspect of their dynamic. However, though she intermittently wandered off, she always came back to check in and reengage when necessary or requested. Andre was consistently observed sewing the design and circuit, whereas Sadia was involved in multiple tasks, including sewing and crafting parts of the wearable controller and working on the Scratch design. For Sadia, the salience of their ability to positively reinforce each other could be due to taking on team leadership functions, her vantage points with multiple aspects of the project, or her commitment to the final design.

Team members also reflected on perspectives related to expressing. When focusing on several aspects of the Scratch design, which she coordinated with Elisa, Sadia felt that the design and coding process helped her create and express her ideas: "It was cool, because you get to pick your avatar and, like, choose a color. You get to change the dance moves and the moves you just wanted to create." Similarly, Andre felt that the Lilypad Arduino "was kind of fun because you can make your own wearable accessories. You could also have electricity and electronics hooked up to that." He further reflected on how ModKit "wasn't as confusing as [he] thought it would be" and "was actually pretty simple." Combining the Lilypad and ModKit allowed him to "make stuff that is complex to people who don't know how to make it." In other words, learning with this novice-friendly platform provided an opportunity to deconstruct what he thought originally was difficult

in order to allow him to express his design ideas. He further reflects on how he was able to make what could look “complex to people” only because they don’t know how to make it. In a sense, this sentiment is self-referential, since he now understands how coding and fabricating with a microcontroller is not as complicated as he used to think it was.

Each of the participants also vocalized shifts in computational perceptions, sometimes in tandem with computational perspectives, in different ways. An ecological view of perceptual shifts around computing are not limited to seeing oneself as a computer scientist or professional, but instead emphasize changing ideas on its personal relevance and application generally, such as now understanding that computing is not limited to screen-based coding or design and can include a variety of forms, such as e-textiles, or can facilitate different expressive capacities, such as augmenting playful interactions with wearables. While Elisa felt that she “used everything” and “learned everything” during the workshop, when asked directly if the workshop made her feel differently about computing, she responded: “I learned how to do stuff... [it] kind of helped me in engineering. I’m, like, ‘You can do it this way instead of that way.’” However, when asked about computing careers, she is firm in her assertion that she has “always wanted to be... a specialist doctor or a pediatric neurosurgeon,” and while the workshop has not changed that resolve, she reflects that she feels more positive about computing careers because they “could be fun.”

Andre felt that the workshop demystified computing and design. He expressed interest in all of the tools used in the workshop, feeling that Scratch “was a new experience because it combined science that [he] knew and other things [he] didn’t know and made it like a combined subject,” while Makey Makey was instrumental in that “you could hook it up with your Lilypad and make a game on the computer and work with it.” Not only did he evidence an understanding of how each of the parts worked together but he could articulate an empowerment as fostered through such an understanding: “you could... make stuff that is complex to people who don’t know how to make it.” He further conveyed newfound appreciation for both programming and design as a process, which can be learned. For example, reflecting on coding, he considered this: “If you practice programing... you would be able to create more complex games... and things.” Andre’s shifting perceptions are revealed in his newfound understanding of ways that digital content creation tools such as Scratch connect to science. They are also reflected in his changing views on computing complexity; for example, he has evolved from viewing coding in ModKit as difficult to a task that wasn’t as complicated as he thought to, finally, one that can be learned incrementally and through deliberate practice.

Out of all of the team members, the shift appeared to be most profound for Sadia, who demonstrated metacognitive awareness of the ways that the design process affected her:

Sadia: I actually always liked design before the workshop. Just looking at different things and different artwork, or just different designs in general, like buildings and stuff like that. I always like looking at different structures... being exposing to programming and coding and you can kind of see the science behind it. How people actually go through the process and make those things...

Interviewer: Did the workshop make you feel differently about design?

Sadia: Yeah, because when you’re programming, you’re kind of designing. Design code and stuff like that.

Interviewer: How do you feel about creating wearable technologies?

Sadia: That was one of the funnest parts of the workshop. When we got the glove and the PlayDoh, that was kind of cool. You just put it on and click it and it would move, so that was pretty cool. You were controlling it.

Interviewer: Did the workshop make you feel differently about that?

Sadia: Yeah, because I thought they were really super complicated and a kid wouldn't make it. It was so hard. We made it, so that kind of got rid of that mindset, I guess.

Interviewer: How do you feel about a career in computers or technology?

Sadia: I'm pretty open to them. I don't know really what I want to do right now, but I definitely have an interest in that. That would be something cool to do. Programming, coding, stuff like that.

Interviewer: Did the workshop make you feel differently about a career in computers and technology?

Sadia: Yeah, at first, I was like, "Oh, I probably wouldn't be able to do this." At first, I found it boring, people just sit there and type up stuff. It's actually a lot of interactive work with your people. You're making your codes or stuff like that, or just making wearable devices. Definitely did change my mindset about that.

Interviewer: Do you feel better or worse about computing or computer careers now that you've done the workshop?

Sadia: Probably better, because it's kind of a fun experience. Those eight weeks we did... it was pretty cool. It does give me a different mindset of them and just changed my thinking of them. That's kind of more positive than it was, I guess. It wasn't negative, but it was just, "Oh, it doesn't seem very appealing to me." But now it kind of changed my mind on that.

Through this exchange in her postinterview, Sadia reveals several aspects of perceptual change. Like Andre, she now reflects on "the science behind" coding. She has made new associations between different creative processes that she connects to computing, such as coding and design ("when you're programming you're kind of designing"). She has further shifted from a "mindset" of thinking of certain kinds computing as being "complicated" and unimaginable for youth creation to computing as a diverse set of practices that involve a "lot of interactive work with your people."

5 CONCLUSIONS, LIMITATIONS, AND FUTURE DIRECTIONS

Throughout this article, we explored how designing complex systems (or systems-of-systems) as facilitated through BRD with contemporary digital and physical toolkits found in making and content creation could (1) influence the complexity with which students engage in and understand computational thinking and (2) impact learners' meaning making with and application of computational thinking. Specifically, we explored two overarching questions: (1) How does designing integrated systems as facilitated through digital games and physical controllers—using e-textiles, the Lilypad Arduino, ModKit, Makey Makey, and Scratch—influence the complexity with which students engage in and understand computational thinking? and (2) How does the process of engaging in BRD impact learners' meaning making with and application of computational concepts? This study helps fill the gap noted in a recent review of the literature that found that more research is needed to demonstrate how differently designed maker activities and tools contribute to learning. Specifically, we emphasized how BRD incorporates and extends multimodal learning theory by adding tangible and integrative dimensions as additional modalities learners can leverage to facilitate learning, metacognition, and agency. We engaged in a multimodal, mixed-methods analysis of diverse high school youth learning to design bidirectionally responsive, wearable games, focusing on a case study of one of the teams who created a BRD project.

First, we engaged in a multipronged qualitative and quantitative analysis of practices that are uniquely enabled through BRD. BRD involves Scratch coding and media design, e-textile coding, and design with the Lilypad Arduino and ModKit, plus integration through simplified physical computing with Makey Makey. Thus, it helps us leverage a multimodal analysis that examines

varied physical and digital artifacts of knowledge (e-textile circuits, ModKit code, Scratch code, and multimedia design). Through this analysis, we aimed to extend our understanding of how youth engage in computational thinking through novice-friendly physical and digital toolkits found in maker spaces and digital content creation communities.

Through our analysis, we were able to find evidence for diverse entry points and practices; complex and wide-ranging applications of computational thinking; and meaningful, yet divergent, shifts in learners' perceptions of computing in their lives. First, through content analysis of digital and physical artifacts, we were able to view the team's collaborative evidence of conceptual and procedural understanding of multimedia coding, microcontroller coding, e-textile circuit design, crafting and fabrication, physical computing, and systems integration to facilitate BRD. Examining their holistic design, we see that there is a good understanding of integration through BRD. They were able to accurately complete the connections between the physical and digital aspects of the design. Second, we found that their understanding of e-textile circuitry was very strong given that they accurately sewed their conductive traces so that they did not short the circuit. Third, looking at their Scratch code revealed some structural weaknesses in their understanding of continuous communication and reference points of interaction with their wearable controller. Their ModKit code revealed that they had applied knowledge of loops and conditionals, which the process of serial communication may have reinforced for them.

Looking at the artifacts alone did not reveal all of their computational thinking processes. By using observational analyses and interviews, we came to a better understanding of their knowledge and application of computational concepts, practices, perspectives, and perceptions. Observational analysis revealed that, by not incorporating loops, they had to constantly restart their Scratch project. Also, by having three sequences for three different key interactions and only two interactive controls on their wearable controller, they had to adjust their Scratch design for the intended interaction. It was through these episodes that they also demonstrated their understanding of computational practices, such as testing, experimenting, and debugging, in the face of novel challenges, such as when Sadia and Elisa were able to quickly see and respond to the problems in their BRD design with a reasonably deduced solution.

Through the qualitative data, we also came to a better understanding of the various ways each team member applied computational practices, which was somewhat mediated by their role and attentiveness to the integrated design. Andre and Elisa were more compartmentalized in their design roles than Sadia, who coordinated their combined efforts and oversaw the completion of the BRD as a whole while also contributing to different parts. Her unique vantage point allowed her to discuss more pointedly ways that the team engaged in modularizing and abstracting by visualizing and implementing the incremental steps and integrating them to create the final design. In part, it also helps to reveal how other team members may not have expressed a holistic view of the project, having taken on specific roles, and instead discussed, as in Andre's case, how the integrated design elements complemented their contribution. It is difficult to know whether this was due to their role specifically, since Elisa, for example, was often preoccupied and her distraction could have contributed to her lack of expression of the integrated process and product. In industry, many professionals contribute to different parts of the final design, with often only a few colleagues overseeing and coordinating the comprehensive vision; thus, this may not necessarily be a weakness as much as it is part of authentic practice.

The observations and interviews also helped us better understand their computational perceptions, which now reflected greater diversity, such as computing involving e-textiles or as a possible field or platform for personal expression or professional aspirations. Taking an ecological view, we see that each team member articulated perceptual changes, such as Elisa discussing connections to engineering or, more directly, Andre and Sadia, who now see "the science behind" coding that

they had not seen before. Sadia, in particular, is metacognitive and reflective of several perceptual shifts, including connections to coding and design and the ability for youth to design complex systems. She is also pointed in her shift from thinking that coding and computing were “boring” or something she was incapable of to one that was interesting, “fun,” and involved “interactive work with... people.”

Overall, we found value for teaching a curriculum around bidirectionally responsive making, specifically for (1) inclusive collaboration, (2) applications to authentic practice, and (3) understanding of complex systems. While coding with Scratch, making physical computing projects with Makey Makey, and designing e-textiles all provide valuable components of learning computing and design, each in isolation have limits for leveraging a variety of interests and practices. For example, if we had focused the entire curriculum around e-textiles or Scratch and Makey Makey, at least one member of the team, even if it consisted of a pair, would have had a more peripheral role in the process at a given time. This, in and of itself, may not have meant that they would not have engaged in meaningful learning but that they may not have had the same opportunity to apply and contribute their understanding to the design as a whole. We see this with Elisa, who took on the role of coder and who did not actively engage in the design process once the Lilypad and Scratch programming was completed. Had she been more active during the integration process to discuss the coding features, they may have avoided having a mismatch between the physical and digital interaction design.

Although this analysis is limited to one case study group, we expand on this point in other work that includes other design teams, such as those who engaged in unidirectional versus bidirectional designs [72]. Here, there is some evidence to suggest that the perspective shifts noted by Sadia—and, to an extent, Andre—are in part a response to being active throughout the process and engaging in different aspects of computing that they had not been able to before and, particularly for Sadia, her unique vantage point on the entire process.

5.1 Future Directions and Limitations

One next step in this line of research is understanding the best ways to teach such curricula in terms of the modes of instruction and the tools themselves. Here, we discussed the value of multi-modal instruction, in which we have purposefully expanded the use in subsequent iterations. We still need to better understand the value of project-based and problem-based instruction versus curriculum time. The first-time that Richard designed and taught the workshop, it was structured around a series of lectures with short expository exercises (i.e., problem-based instruction). The second-time she taught it, the curriculum used activities created during the first workshop as incremental instructional prototypes (“buildable models”). The more didactic model in the first workshop may have been better at transmitting conceptual knowledge to students, since project teams that did not create bidirectionally responsive projects could still articulate how the toolkits worked together [73] and many could also communicate the value of integrating them. Another limitation for understanding the efficacy of project-based instruction may be that learners were unable to fully complete buildable model 3 due to the error noted before.

The toolkits themselves may have helped make the learning process accessible for students; both Scratch and Modkit contained comparable block-based coding structures. However, Modkit is no longer being supported by its development team; thus, alternative block-based coding platforms need to be explored. Ardublock was not used because of its complexity and less intuitive affordance match to Scratch; Snap4Arduino was in its alpha stage at the time of the course without adequate documentation. We have currently been testing mBlock [52], a coding environment intended for the Makeblock mBot, a robotic kit for kids, which has recently added features for

Arduino microprocessors. While we have found some limitations, which we will expand on in a future article, thus far we have been able to redesign most of our curriculum with it.

A potential limitation we further acknowledge is that we used a frame of computational thinking as opposed to computational literacy [18], which predates the former, and separates the tools and affordances from the cognitive and social aspects of computing. However, given that overarching literature has deconstructed the computational thinking framework for relevant analyses and that we used the framing of both Brennan and Resnick [9] and Kafai et al. [37], we have, albeit in an indirect way, integrated a computational literacy approach.

Another area to explore further is the inclusive affordances of BRD. As noted for this case design team and others, as well as with similar adult activities [75], male team members often engaged in the sewing aspects of the design process and were more comfortable expressing their interests in crafting. We feel, as discussed in related research (e.g., [37, 75]), that this was in part because of crafting's connection to computing, which is viewed as a masculine domain. We want to further understand why creating integrated systems that leverage e-textiles seems to highlight this trend more than standalone e-textile workshops, which also tend to have more female representation (this workshop, conversely, had almost equal participation). Exploring ways that tools themselves offer implicit affordances for equity is an area ripe for further exploration.

In this article, we provided a case study of youth creating a bidirectionally responsive project in order to demonstrate that combining different affordances and associated practices can tap into creative computational possibilities and increase youths' appreciation of the diversity of computing, its relevance to their lives and interests, and the realities of the team-based practices embedded in design and making. Moreover, we provide evidence of young learners' abilities to engage in and have a better understanding of complex systems through integrated design. We encourage the field to consider similar diverse and complex approaches through purposeful integration of multiple tools and activities as opposed to learning different toolkits by themselves. As we have seen, computational toolkits are constantly evolving, and newer tools continue to simplify the design process. What remains constant are the design possibilities and computational literacies that we can help instill beyond the tools themselves.

ACKNOWLEDGMENTS

Radium Yang and Emily Dieckmeyer provided course assistance and helped with curricular design. Additional data analysis was conducted by Robert William Ashley. The views expressed represent the authors and are not necessarily those of the funding organizations or affiliated universities.

REFERENCES

- [1] Shaaron Ainsworth and Nicolas Van Labeke. 2002. Using a multi-representational design framework to develop and evaluate a dynamic simulation environment. In *Proceedings of the International Workshop on Dynamic Visualizations and Learning*. Knowledge Media Research Center, Tübingen, Germany, 1–9. Retrieved November 10, 2018 from <https://www.iwm-tuebingen.de/workshops/visualization/proceedings.htm>.
- [2] Phillip Anderson. 1999. Perspective: Complexity theory and organization science. *Organization Science* 10, 3, 216–232.
- [3] Ryan S. Anderton, Li Shan Chiu, and Susan Aulfrey. 2016. Student perceptions to teaching undergraduate anatomy in Health Sciences. *International Journal of Higher Education* 5, 3, 201–216.
- [4] Edward Baafi, Collin Reisdorf, and Amon Milner. 2013. ModKit Alpha. Retrieved September 30, 2016 from <http://www.modk.it/alpha>.
- [5] Paulo Blikstein, Tamar Fuhrmann, Daniel Greene, and Shima Salehi. 2012. Bifocal modeling: mixing real and virtual labs for advanced science learning. In *Proceedings of the 11th International Conference on Interaction Design and Children*. ACM, New York, NY, 296–299.
- [6] P. Blikstein. 2013. Digital fabrication and “making” in education: The democratization of invention. In *FabLabs: Of Machines, Makers and Inventors*, J. Walter-Herrmann and C. Böching (Eds.). Bielefeld: Transcript Publishers, 1–21.

- [7] Tara Brady, Camille Salas, Ayah Nuriddin, Walter Rodgers, and Mega Subramaniam. 2014. MakeAbility: Creating accessible makerspace events in a public library. *Public Library Quarterly* 33, 4, 330–347.
- [8] Lisa Brahms and Kevin Crowley. 2016. Making sense of making: Defining learning practices in make magazine. In *Makeology, Vol 2: Makers as Learners*, Kylie Peppler, Erica Halverson, and Yasmin Kafai (Eds.). Routledge, New York, 13–28.
- [9] Karen Brennan and Mitchel Resnick. 2012. New frameworks for studying and assessing the development of computational thinking. In *Proceedings of the 2012 Annual Meeting of the American Educational Research Association*. AERA, Vancouver, BC, Canada, 1–25. Retrieved November 10, 2018 from https://web.media.mit.edu/~kbrennan/files/Brennan_Resnick_AERA2012_CT.pdf.
- [10] Beth Buchholz, Kate Shively, Kylie Peppler, and Karen Wohlwend. 2014. Hands on, hands off: Gendered access in crafting and electronics practices. *Mind, Culture, and Activity* 21, 4, 278–297.
- [11] Leah Buechley, Michael Eisenberg, Julian Catchen, and Ali Crockett. 2008. The LilyPad Arduino: Using computational textiles to investigate engagement, aesthetics, and diversity in computer science education. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems Languages (SIGCHI'08)*. ACM Press, New York, NY, 423–432.
- [12] Kathy Charmaz. 2008. Constructionism and the grounded theory method. In *Handbook of Constructionist Research*. The Guilford Press, New York, NY.
- [13] Gongxiang Chen and Xiaolan Fu. 2003. Effects of multimodal information on learning performance and judgment of learning. *Journal of Educational Computing Research* 29, 3, 349–362.
- [14] Edgar Dale. 1950. *Audio Visual Methods in Teaching*. The Dryden Press, Inc., New York, NY.
- [15] Sayamindu Dasgupta, William Hale, Andres Monroy-Hernández, and Benjamin M. Hill. 2016. Remixing as a pathway to computational thinking. In *Proceedings of the 19th ACM Conference on Computer-Supported Cooperative Work & Social Computing (CSCW'16)*. ACM Press, New York, NY, 1438–1449.
- [16] Stefan Decuyper, Filip Dochy, and Piet Van denBossche. 2010. Grasping the dynamic complexity of team learning: An integrative model for effective team learning in organisations. *Educational Research Review* 5, 2, 111–133.
- [17] Sharon J. Derry, Roy D. Pea, Brigid Barron, Randi A. Engle, Frederick Erickson, Ricki Goldman, Rogers Hall, Timothy Koschmann, Jay L. Lemke, Miriam Gamoran Sherin, and Bruce L. Sherin. 2010. Conducting video research in the learning sciences: Guidance on selection, analysis, technology, and ethics. *The Journal of the Learning Sciences* 19, 1, 3–53.
- [18] Andrea A. diSessa. 2001. *Changing Minds: Computers, Learning, and Literacy*. MIT Press, Cambridge, MA.
- [19] Peter E. Doolittle, Andrea L. McNeill, Krista P. Terry, and Stephanie B. Scheer. 2005. Multimedia, cognitive load and pedagogy. In *Interactive Multimedia in Education and Training*, S. Mishra and R. C. Sharma (Eds.). Idea Group, Inc., London, 184–212.
- [20] Dale Dougherty. 2013. The maker mindset. In *Design. Make. Play. Growing the Next Generation of STEM Innovators*, Margaret Honey and David Kanter (Eds.). Routledge, New York, NY, 7–16.
- [21] Allison Druin. 2002. The role of children in the design of new technology. *Behaviour and Information Technology* 21, 1, 1–25.
- [22] Michael Eisenberg and Narcis Pares. 2014. Tangible and full-body interfaces in learning. In *The Cambridge Handbook of the Learning Sciences*, R. Keith Sawyer (Ed.). Cambridge University Press, 339–357.
- [23] Waguih ElMaraghy, Hoda ElMaraghy, Tetsuo Tomiyama, and Laszlo Monostori. 2012. Complexity in engineering design and manufacturing. *CIRP Annals-Manufacturing Technology* 61, 2, 793–814.
- [24] Barbara Ericson and Tom McKlin. 2015. Helping African American students pass advanced placement computer science: A tale of two states. In *Research in Equity and Sustained Participation in Engineering, Computing, and Technology (RESPECT'15)*. IEEE, 1–8.
- [25] Frederick Erickson. 2006. Definition and analysis of data from videotape. In *Handbook of Complementary Methods in Education Research*, Judith L. Green, Gregory Camilli, and Patricia B. Elmore (Eds.). AERA/Lawrence Erlbaum, Washington, DC, 177–192.
- [26] Charles Fadel and Cheryl Lemke. 2008. *Multimodal Learning Through Media: What the Research Says*. Cisco Systems, San Jose, CA.
- [27] Deborah Fields, Michael T. Giang, and Yasmin Kafai. 2014. Programming in the wild: trends in youth computational participation in the online scratch community. In *Proceedings of the 9th Workshop in Primary and Secondary Computing Education (WIPSC'14)*. ACM, New York, NY, 2–11.
- [28] Deborah Fields, Michael T. Giang, and Yasmin Kafai. 2016. Coding by choice: A transitional analysis of social participation patterns and programming contributions in the online Scratch community. In *Mass Collaboration and Education*, Ulrike Cress, Johannes Moskaliuk, and Heisawn Jeong (Eds.). Springer International Publishing, Switzerland, 209–240.
- [29] Allan Fowler and Brian Cusack. 2011. Kodu game lab: improving the motivation for learning programming concepts. In *Proceedings of the 6th International Conference on Foundations of Digital Games*. ACM, 238–240.

- [30] Connie Golsteijn, Elise Van Den Hoven, David Frohlich, and Abigail Sellen. 2013. Hybrid crafting: towards an integrated practice of crafting with physical and digital components. *Personal and Ubiquitous Computing* 18, 3, 593–611.
- [31] Thomas A. Henzinger and Joseph Sifakis. 2007. The discipline of embedded systems design. *Computer* 40, 10, 32–40.
- [32] Margaret Honey and David Kanter. 2013. *Design, Make, Play: Growing the Next Generation of STEM Innovators*. Routledge, New York, NY.
- [33] Carey Jewitt. 2008. Multimodality and literacy in school classrooms. *Review of Research in Education* 32, 1, 241–267.
- [34] Yasmin B. Kafai. 2006. Playing and making games for learning: Instructionist and constructionist perspectives for game studies. *Games and Culture* 1, 1, 36–40.
- [35] Yasmin B. Kafai, Quinn Burke, and Chad Mote. 2012. What makes competitions fun to participate?: The role of audience for middle school game designers. In *Proceedings of the 11th International Conference on Interaction Design and Children*. ACM, 284–287.
- [36] Yasmin B. Kafai, Deborah Fields, and Kristin Searle. 2014. Electronic textiles as disruptive designs: Supporting and challenging maker activities in schools. *Harvard Educational Review* 84, 4, 532–556.
- [37] Yasmin B. Kafai, Eunkyoung Lee, Kristin Searle, Deborah Fields, Eliot Kaplan, and Debora Lui. 2014. A crafts-oriented approach to computing in high school. *TOCE ACM Transactions on Computing Education* 14, 1, 1–20.
- [38] Yasmin B. Kafai and Kylie A. Peppler. 2011. Youth, technology, and DIY: Developing participatory competencies in creative media production. *Review of Research in Education* 35, 1, 89–119.
- [39] Ivan Kaštelan, Nikola Teslic, and Miodrag Temerinac. 2016. Challenges in embedded engineering education. In *Embedded Engineering Education*, Roman Szweczyk, Kaštelan Ivan, Miodrag Temerinac, Moshe Barak, and Vlado Sruk (Eds.). Springer International Publishing, Switzerland, 1–27.
- [40] Paul A. Kirschner, John Sweller, and Richard E. Clark. 2006. Why minimal guidance during instruction does not work: An analysis of the failure of constructivist, discovery, problem-based, experiential, and inquiry-based teaching. *Educational Psychologist* 41, 2, 75–86.
- [41] Gunther Kress, Carey Jewitt, Jon Ogborn, and Tsatsarelis Charalampos. 2001. *Multimodal Teaching and Learning: The Rhetorics of the Science Classroom*. United Kingdom: Bloomsbury, London.
- [42] David Kushner. 2011. The making of Arduino. *IEEE Spectrum*. Retrieved February 6, 2016 from <https://spectrum.ieee.org/geek-life/hands-on/the-making-of-arduino/0>.
- [43] Edward A. Lee. 2015. The past, present and future of cyber-physical systems: A focus on models. *Sensors* 15, 3, 4837–4869.
- [44] Edward A. Lee and Sanjit A. Seshia. 2017. *Introduction to Embedded Systems: A Cyber-Physical Systems Approach* (2nd ed.). MIT Press, Cambridge, MA.
- [45] Eunkyoung Lee, Yasmin B. Kafai, Veena Vasudevan, and Richard Lee Davis. 2014. Playing in the arcade: Designing tangible interfaces with MaKey MaKey for Scratch games. In *Playful User Interfaces*, Anton Nijholt (Ed.). Springer, 277–292.
- [46] SC-Y Lu, Waguih ElMaraghy, Günther Schuh, and Robert Wilhelm. 2007. A scientific foundation of collaborative engineering. *CIRP Annals* 56, 2, 605–634.
- [47] Jay L. Lemke. 2000. Across the scales of time: Artifacts, activities, and meanings in ecosocial systems. *Mind, Culture, and Activity* 7, 4, 273–290.
- [48] Mark W. Maier. 1998. Architecting principles for systems-of-systems. *Systems Engineering: The Journal of the International Council on Systems Engineering* 1, 4, 267–284.
- [49] John H. Maloney, Kylie Peppler, Yasmin Kafai, Mitchel Resnick, and Natalie Rusk. 2008. Programming by choice. In *Proceedings of the 39th SIGCSE Technical Symposium on Computer Science Education*. ACM, 367–371.
- [50] Lee Martin. 2015. The Promise of the Maker Movement for Education. *Journal of Pre-College Engineering Education Research (J-PEER)* 5, 1, 30–39.
- [51] Richard E. Mayer. 2003. The promise of multimedia learning: using the same instructional design methods across different media. *Learning and Instruction* 13, 2, 125–139.
- [52] mBlock. n.d. Retrieved November 10, 2018 from <http://www.mblock.cc/>.
- [53] Amon D. Millner. 2010. *Computer as chalk: cultivating and sustaining communities of youth as designers of tangible user interfaces*. Ph.D. dissertation. Massachusetts Institute of Technology, Cambridge, MA.
- [54] Amon D. Millner and Edward Baafi. 2011. Modkit: Blending and extending approachable platforms for creating computer programs and interactive objects. In *Proceedings of the 10th International Conference on Interaction Design and Children*. ACM, 250–253.
- [55] Tamara J. Moore, Aran W. Glancy, Kristina M. Tank, Jennifer A. Kersten, Karl A. Smith, and Micah S. Stohlmann. 2014. A framework for quality K-12 engineering education: Research and development. *Journal of Pre-College Engineering Education Research (J-PEER)* 4, 1, 1–13.
- [56] Roxana Moreno. 2002. Who learns best with multiple representations? Cognitive theory predictions on individual differences in multimedia learning. In *Proceedings of EdMedia: World Conference on Educational Media and Technology*. AACE, 1380–1385.

- [57] Roxana Moreno and Richard Mayer. 2007. Interactive multimodal learning environments. *Educational Psychology Review* 19, 3, 309–326.
- [58] Jiquan Ngiam, Aditya Khosla, Mingyu Kim, Juan Nam, Honglak Lee, and Andrew Ng. 2011. Multimodal deep learning. In *Proceedings of the 28th International Conference on Machine Learning (ICML'11)*. International Machine Learning Society, 689–696.
- [59] Daniel O'Sullivan and Tom Igoe. 2004. *Physical Computing: Sensing and Controlling the Physical World with Computers*. Thomson, Boston, MA.
- [60] Sofia Papavasiliou, Michail N. Giannakos, and Letizia Jaccheri. 2017. Empirical studies on the maker movement, a promising approach to learning: A literature review. *Entertainment Computing* 18, 57–78.
- [61] Seymour Papert. 1980. *Mindstorms: Children, Computers and Powerful Ideas*. Basic Books, New York.
- [62] Allan Paivio. 1986. *Mental Representations: A Dual Coding Approach*. Oxford University Press, Oxford, UK.
- [63] Katarina Pantic, Deborah A. Fields, and Lisa Quirke. 2016. Studying situated learning in a constructionist programming camp: A multimethod microgenetic analysis of one girl's learning pathway. In *Proceedings of the 15th International Conference on Interaction Design and Children*. ACM, New York, NY, 428–439.
- [64] Kylie Peppler, Melissa Gresalfi, Katie Salen Tekinbas, Rafi Santo, and Leah Buechley. 2014. *Soft Circuits: Crafting e-Fashion with DIY Electronics*. MIT Press, Cambridge, MA.
- [65] Anthony G. Picciano. 2009. Blending with purpose: The multimodal model. *Journal of the Research Centre for Educational Technology* 5, 1, 4–14.
- [66] Kanjun Qiu, Leah Buechley, Edward Baafi, and Wendy Dubow. 2013. A curriculum for teaching computer science through computational textiles. In *Proceedings of the 12th International Conference on Interaction Design and Children*. ACM, New York, NY, 20–27.
- [67] Alexander Repenning, David Webb, and Andri Ioannidou. 2010. Scalable game design and the development of a checklist for getting computational thinking into public schools. In *Proceedings of the 41st ACM Technical Symposium on Computer Science Education*. ACM, New York, NY, 265–269.
- [68] Mitchel Resnick, Robbie Berg, and Michael Eisenberg. 2000. Beyond black boxes: Bringing transparency and aesthetics back to scientific investigation. *The Journal of the Learning Sciences* 9, 1, 7–30.
- [69] Mitchel Resnick and Brian Silverman. 2005. Some reflections on designing construction kits for kids. In *Proceedings of the 2005 Conference on Interaction Design and Children*. ACM, New York, NY, 117–122.
- [70] Mitchel Resnick, John Maloney, Andres Monroy-Hernández, Natalie Rusk, Evelyn Eastmond, Karen Brennan, Amon Millner, Eric Rosenbaum, Jay Silver, Brian Silverman, and Yasmin B. Kafai. 2009. Scratch: programming for all. *Communications of the ACM* 52, 11, 60–67.
- [71] Gabriela T. Richard. 2008. Employing physical computing in education: How teachers and students utilized physical computing to develop embodied and tangible learning objects. *The International Journal of Technology, Knowledge and Society* 4, 3, 93–102.
- [72] Gabriela T. Richard and Sagun Giri. 2017. Inclusive collaborative learning with multi-interface design: Implications for diverse and equitable makerspace education. In *Proceedings of the 12th International Conference of Computer Supported Collaborative Learning*. ISLS, Philadelphia, PA, 415–422.
- [73] Gabriela T. Richard and Yasmin B. Kafai. 2015a. Responsive Make and Play: Youth making physically and digitally interactive and wearable game controllers. In *More Playful User Interfaces: Interfaces that Invite Social and Physical Interaction*, Anton Nijholt (Ed.). Springer, Singapore, 71–93.
- [74] Gabriela T. Richard and Yasmin B. Kafai. 2015b. Making Physical and Digital Games with E-Textiles: A Workshop for Youth Making Responsive Wearable Games and Controllers. In *Proceedings of the 14th International Conference on Interaction Design & Children*. ACM, New York, NY, 399–402.
- [75] Gabriela T. Richard, Yasmin B. Kafai, Barrie Adleberg, and Orkan Telhan. 2015. StitchFest: Diversifying a college hackathon to broaden participation and perceptions in computing. In *Proceedings of the 46th ACM Technical Symposium on Computer Science Education*. ACM, New York, NY, 114–119.
- [76] Terry Royce. 2002. Multimodality in the TESOL classroom: Exploring visual-verbal synergy. *TESOL Quarterly* 36, 2, 191–205.
- [77] Bernd Rüschoff and Markus Ritter. 2001. Technology-enhanced language learning: Construction of knowledge and template-based learning in the foreign language classroom. *Computer Assisted Language Learning* 14, 3–4, 219–232.
- [78] Michael Sankey, Dawn Birch, and Michael Gardiner. 2010. Engaging students through multimodal learning environments: The journey continues. In *Proceedings of ASCILITE 2010: 27th Annual Conference of the Australasian Society for Computers in Learning in Tertiary Education: Curriculum, Technology and Transformation for an Unknown Future*. University of Queensland, Brisbane, Australia, 852–863.
- [79] Kimberly A. Scott, Kimberly M. Sheridan, and Kevin Clark. 2015. Culturally responsive computing: a theory revisited. *Learning, Media and Technology* 40, 4, 412–436.

- [80] JooYoung Seo and Gabriela T. Richard. 2018. Accessibility, making and tactile robotics: Facilitating collaborative learning and computational thinking for learners with visual impairments. In *Proceedings of the 13th International Conference of the Learning Sciences*, Vol. 3. ISLS, 1755–1757.
- [81] Sarah A. Sheard and Ali Mostashari. 2009. Principles of complex systems for systems engineering. *Systems Engineering* 12, 4, 295–311.
- [82] Jay Silver, Eric Rosenbaum, and David Shaw. 2012. MaKey MaKey: improvising tangible and nature-based user interfaces. In *Proceedings of the 6th International Conference on Tangible, Embedded and Embodied Interaction*. ACM, New York, NY, 367–370.
- [83] Arnan Sipitakiat and Paulo Blikstein. 2013. Interaction design and physical computing in the era of miniature embedded computers. In *Proceedings of the 12th International Conference on Interaction Design and Children*. ACM, New York, NY, 515–518.
- [84] Shaunna Smith. 2013. Through the teacher’s eyes: Unpacking the TPACK of digital fabrication integration in middle school language arts. *Journal of Research on Tech in Education* 46, 2, 207–227.
- [85] Kihyuk Sohn, Wenling Shang, and Honglak Lee. 2014. Improved multimodal deep learning with variation of information. In *Proceedings of Advances in Neural Information Processing Systems*. NIPS, 2141–2149.
- [86] Gerry Stahl, Timothy Koschmann, and Dan Suthers. 2006. Computer-supported collaborative learning: An historical perspective. In *Cambridge Handbook of the Learning Sciences*, R. Keith Sawyer (Ed.). Cambridge University Press, Cambridge, UK, 409–426.
- [87] Norbert A. Streitz, Peter Tandler, Christian Müller-Tomfelde, and Shin’ichi Konomi. 2001. Roomware: Towards the next generation of human-computer interaction based on an integrated design of real and virtual worlds. *Human-Computer Interaction in the New Millennium*. Addison Wesley, New York, NY, 551–576.
- [88] Joshua D. Summers and Jami J. Shah. 2010. Mechanical engineering design complexity metrics: Size, coupling, and solvability. *Journal of Mechanical Design* 132, 2, 021004-1-11.
- [89] Karen Swan. 1991. Programming objects to think with: Logo and the teaching and learning of problem solving. *Journal of Educational Computing Research* 7, 1, 89–112.
- [90] Jaons Sztipanovits, Gautam Biswas, Ken Frampton, Aniruddha Gokhale, Larry Howard, Gabor Karsai, T. John Koo, Xenofon Koutsoukos, and Douglas C. Schmidt. 2005. Introducing embedded software and systems education and advanced learning technology in an engineering curriculum. *ACM Transactions on Embedded Computing Systems* 4, 3, 549–568.
- [91] Brygg Ullmer and Hiroshi Ishii. 2000. Emerging frameworks for tangible user interfaces. *IBM Systems Journal* 39, 3.4, 915–931.
- [92] Veena Vasudevan and Yasmin B. Kafai. 2016. Bridging Crafting and Computing in Making: Designing Interactive Touchpads and Board Games with MaKey MaKey. In *Makeology: Makerspaces as Learning Environments*. Vol. 1, Kylie Peppler, Erica Halverson, and Yasmin B. Kafai (Eds.). Routledge, New York, NY.
- [93] Stefan Wermter, Cornelius Weber, Mark Elshaw, Christo Panchev, H. Erwin, and Friedemann Pulvermüller. 2004. Towards multimodal neural robot learning. *Robotics and Autonomous Systems* 47, 2, 171–175.
- [94] Linda Werner, Jill Denner, Sharon Campe, and Damon Chizuru Kawamoto. 2012. The fairy performance assessment: measuring computational thinking in middle school. In *Proceedings of the 43rd ACM Technical Symposium on Computer Science Education*. ACM, New York, NY, 215–220.
- [95] Jeannette M. Wing. 2006. Computational thinking. *Communications of the ACM* 49, 3, 33–35.
- [96] Oren Zuckerman, Saeed Arida, and Mitchel Resnick. 2005. Extending tangible interfaces for education: Digital Montessori-inspired manipulatives. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, New York, NY, 859–868.

Received November 2016; revised June 2018; accepted July 2018