TechTrends (2021) 65:196–212
https://doi.org/10.1007/s11528-020-00562-9

ORIGINAL PAPER

Broadening Participation in Computing: Promoting Affective and Cognitive Learning in Informal Spaces

Hui Yang[1] · Diane Codding[2] · Chrystalla Mouza[3] · Lori Pollock[4]

Accepted: 5 November 2020 / Published online: 5 January 2021

Abstract
In this work we examine youth learning in an informal computing program implemented through a library-university partnership. In particular, we introduce and illustrate a culturally responsive computing framework which served as a foundation for the design of the program. Subsequently, we examine youth collaboration as well as affective and cognitive learning outcomes. Data were collected from university program facilitators and 30 youth over one semester. Data were collected through observations, lesson plans, computational artifacts and interviews with two case study youth. Results indicated that youth formed a variety of learning communities during the collaborative development of computing artifacts. Frequent participants were found to work with a greater number of peers compared to less frequent participants. Results from case study participants also indicated improvements in their computational competencies. Findings from this work have implications for the design of informal learning environments that help broaden participation in computing.

Keywords University-library partnerships · Computational thinking · Informal learning environments

Introduction

As technology continues to permeate every aspect of our lives, computational thinking (CT) has emerged as an essential body of knowledge needed to work and function in the twenty-first century. Broadly speaking, CT is a problem-solving process that can be executed with computers and can be transferred and applied across disciplines (Barr and Stephenson 2011). CT is also an analytical skill that is widely required for both personal and professional decision-making as well as civic participation (Blikstein and Moghadam 2019; Computer Science Teachers' Association [CSTA] & International Society for Technology in Education [ISTE] 2011; Wing 2006). Importantly, CT has been recognized as an important mechanism for promoting computer science (CS) education into K-12 settings (Yadav et al. 2018). Specifically, early exposure to CT has shown promise for engaging students in developing a range of computing concepts as well as promoting the development of computing identities, particularly among females and underrepresented minorities (e.g., Bers et al. 2014; Estrada et al. 2018; Shute et al. 2017).

A key challenge in engaging all students with CT at an early age is access to both resources and knowledgeable teachers who can teach computing in pedagogically sound ways (Mouza et al. 2018). Students from underrepresented backgrounds, for instance, are more likely to attend under-resourced schools that lack opportunities for computing, particularly opportunities that connect computing with culturally relevant frameworks (Margolis et al. 2017). Culturally relevant frameworks integrate knowledge relevant to students' identities and communities with computational learning activities, maximizing the potential for increasing the engagement, competence, and belonging of all youth in computing (Eglash et al. 2013; Kafai et al. 2014; Ryoo 2019).

✉ Hui Yang
huiyang@cornell.edu

Diane Codding
dcodding@udel.edu

Chrystalla Mouza
cmouza@udel.edu

Lori Pollock
pollock@udel.edu

1 Cornell University, 211 Gates Hall, 107 Hoy Road, Ithaca, NY 14853, USA

2 University of Delaware, 116 Willard Hall, Newark, DE 19716, USA

3 University of Delaware, 113 Willard Hall, Newark, DE 19716, USA

4 University of Delaware, 436 Smith Hall, Newark, DE 19716, USA

Informal institutions, such as libraries, could play an active role in supporting formal school computing efforts, providing resources potentially unavailable in K-12 classrooms. Specifically, libraries have recently reinvented themselves as educational hubs that aim to democratize and diversify computing education by providing access to resources, tools, and programs more readily available to community members (Lee and Phillips 2019). Given their informal nature, library-based programs are more likely to support participants' goals and interests in ways that differ from school-based learning (Penuel et al. 2019). Similar to teachers, however, librarians are not experts in CS, thus making it difficult to facilitate pedagogically-sound computing programs (Martin 2019; Rogowski et al. 2018).

One way to address this challenge and fully recognize the affordances of libraries is through partnerships with university researchers (Lee et al. 2017; Penuel et al. 2019). Such partnerships can provide facilitators with content and pedagogical expertise essential for facilitating meaningful experiences that support youth interests, collaboration, and sense of belonging (Koh et al. 2019; Lee 2019). In this work we present one kind of partnership between a library and a local university. Further, we examine the ways in which such a partnership can help promote positive computing experiences for youth, including female and racially minoritized groups, in the context of an after-school program called the Scratch Technology Club (STC). For the purpose of this work, we first present the design principles guiding the STC and subsequently address the following research questions:

1. In what ways did youth participation in the STC at a local library support peer learning and collaboration?
2. How did participating youth progress in their development and understanding of CT (cognitive) as well as their sense of belonging (affective) within computing?

Review of the Literature

Developing Computational Thinking in Informal Environments

CT was originally introduced by Papert (1980) as a new approach to learning mathematics using computational methods. CT encompasses fundamental CS concepts and cognitive processes that help students analyze and solve real-world problems (Grover and Pea 2013; Wing 2006), including: (a) decomposition (e.g., breaking a problem down into a series of manageable steps); (b) pattern recognition (e.g., looking for repetitive patterns in the problem to design an efficient solution); (c) abstraction (e.g., representing the solution in a generalized form); (d) algorithm design (e.g., designing the solution in a systematic way); and (e) evaluation (e.g., ensuring all algorithmic steps are followed through to a comprehensive solution) (Anderson 2016; Atmatzidou and Demetriadis 2014). Computer programming relies on these concepts but CT is far more than programming. According to Wing (2006), CT is a skill that leads to the ability to program but also a generally useful skill in a digital world.

The recognition of CT as an essential skill for twenty-first-century citizens has resulted in K-12 curricular changes around the world (Angeli et al. 2016). Programming, in particular, has emerged as one way of promoting CT due to the recent availability of visual programming construction tools like Scratch, which lower the floor of entering into programming and facilitate the creation of interactive media (Brady et al. 2017; Kafai and Burke 2013). Brennan and Resnick (2012) proposed a three-dimensional CT framework for K-12 education built around Scratch, that includes CT concepts, practices, and perspectives. CT concepts are organized around programming competencies including sequences, loops, events, parallelism, conditionals, operators and data. CT practices refer to how children learn about CT knowledge and skills. Finally, CT perspectives refer to children's reflections or attitudes towards computing (Brennan and Resnick 2012).

While most efforts to promote the development of CT knowledge and skills have been implemented in formal learning environments (i.e., schools), a report from the American Library Association's Office for Information Technology Policy highlighted the promising potential of informal learning spaces (e.g., libraries) in these efforts (Braun and Visser 2017). Libraries have the potential to democratize computing by providing programming to more diverse audiences in the communities they serve (Koh et al. 2019). Youth learning in libraries, however, is substantially different from school learning in that it is learner-centered and interest-driven (Penuel et al. 2019). As a result, library-based learning environments are frequently organized around a framework called "connected learning" (Ito et al. 2013), which is characterized by three key components: (a) *peer supported*: peers help each other, work together to solve a problem, or get friends to try new activities; (b) *interest powered*: supports youth in exploring new interests relevant for social and learning development; and (c) *future oriented*: helps youth develop career pathways and envision their future selves (Penuel et al. 2019).

Despite their promise, informal programs face some unique challenges that require attention to specific design principles. One challenge is related to the drop-in nature of youth, which makes it difficult when designing multiday or multi-week offerings (Martin 2019). According to Ito et al. (2013), this challenge requires the design of activities with multiple entry points that allow equitable participation as well as activities with "high ceilings"—opportunities for youth to go deeper into their interests. Another challenge is access to facilitators who can support the design and implementation of engaging programs using high-leverage pedagogical strategies (Braun and Visser 2017). In particular, opportunities to support youth in informal spaces benefit from learning-by-doing activities—activities that engage youth in making, coding, or creating (Martin 2019; Penuel et al. 2019). As noted, one approach to address these challenges is through partnerships that connect researchers and mentors with librarians to leverage the design of informal environments that built on learning theory.

To date, a number of studies have emerged focusing on CT learning in libraries and other informal spaces. These studies focus on the design of workshops and their impact on youth's interest in coding and STEM careers (Martin 2019), the role of libraries in delivering technology-enhanced family storytelling experiences (Tzou et al. 2019), youth's sense of personal STEM identity as a result of participation in media production activities (Ahn et al. 2014), and the role of mentors in facilitating creative computing experiences (Roque and Jain 2018). Despite that, more research is needed on how to support ongoing improvement of informal learning designs, in ways that help libraries modify their offerings to support outcomes for all youth (Penuel et al. 2019).

Conceptual Framework

I. Lee et al. (2011) advanced a conceptual framework, called Use-Modify-Create, to represent youth's acquisition and development of CT based on systematic analyses of formal and informal computing programs. This framework specifies youth's CT progression from consumers to creators of computing, placing an emphasis on illuminating cognitive development. In the "Use" stage, youth only consume products created by others. Over time, they are able to "Modify" these products based on their developing CT skills. As youth progress in their CT skills and understanding, they are able to "Create" their own products focusing on topics and issues of their own choosing.

While this framework illustrates the development of CT from a cognitive perspective, learning in informal settings is also a cultural process; individuals' experience is often shaped by the cultural practices within their communities, including opportunities to work with peers in collaborative settings (Barton and Tan 2010; Vygotsky 1978). Therefore, we argue that acquiring CT knowledge and skills in informal settings, such as libraries, should simultaneously emphasize both a cognitive and cultural progression. Based on this notion, we build on the Use-Modify-Create model and literature on culturally responsive pedagogy to introduce the *Culturally Responsive Computing Framework* (Fig. 1). This redesigned framework, which served as the foundation of this work, encompasses a cultural dimension, whereby youth are not only able to set viable goals based on their own interests (e.g., choose what project to design), but importantly they are able to pursue such goals with the support and respect of others in the informal context (Kafai et al. 2014; Peter et al. 2003).

The culturally responsive computing framework utilized culturally relevant pedagogy (Ladson-Billings 1995; Pollock 2008) as a foundation to creating a program that served underrepresented racially minoritized and female youth. Specifically, we followed three design guidelines aligned with culturally responsive frameworks: (a) research-based computing practices for teaching and engaging a diverse population of youth (e.g., pair programming where two programmers work together on a single computer); (b) practices that build on the knowledge and assets of communities (e.g., collectivism); and (c) culturally responsive interactions between facilitators and youth underrepresented in computing (e.g., relationship building, positive behavior management) (Codding et al. 2019).

Noticeably, youth's computing identities are central to these strategies as we seek to build their sense of belonging in the STC and the field of computing. Goodenow (1993) defines belonging as an individual's perception of acceptance, respect, inclusion, and support. Such perceptions are important to the development of informal computing programs as a lack of belonging impacts youth's connection to the space, their academic motivation, and their psychological well- being (Maestas et al. 2007). The culturally responsive computing framework aimed at increasing youth sense of belonging through culturally affirming interactions with peers who acknowledge, value, and incorporate their individual identities and cultural backgrounds (Pollock 2008). By developing youth's sense of belonging, we as program facilitators and researchers sought to promote a youth-led infusion of culturally relevant themes into CS projects, as well as to facilitate youth overcoming self-doubt and persistence in computing (Veilleux et al. 2012). Culturally relevant themes included collaborative game development, personal storytelling, and community-based problem solving.

By introducing a cultural dimension to the Use-Modify-Create framework, we propose that while youth advance in their learning of CT they also progressively develop a sense of belonging in computing. In the "Use" stage, when youth engage with technology as users, the programming is devoid of personally-meaningful elements and youth are less likely to develop ownership of their learning. Additionally, youth at this stage are unlikely to experience a sense of belonging or progress in their computing identity development. In the "Modify" stage when youth experiment with remixing and modifying projects, they begin to impact the learning environment by infusing their identities and cultures into their programming, thus influencing the planning and design of future STC sessions. In this way, youth at the Modify stage begin to benefit from and contribute to culturally relevant design decisions. In the "Create" stage where youth become creators of computing products, they are more likely to experience a strong sense of belonging and ownership of their learning. Additionally, they are able to take leadership in changing the learning environment to better reflect

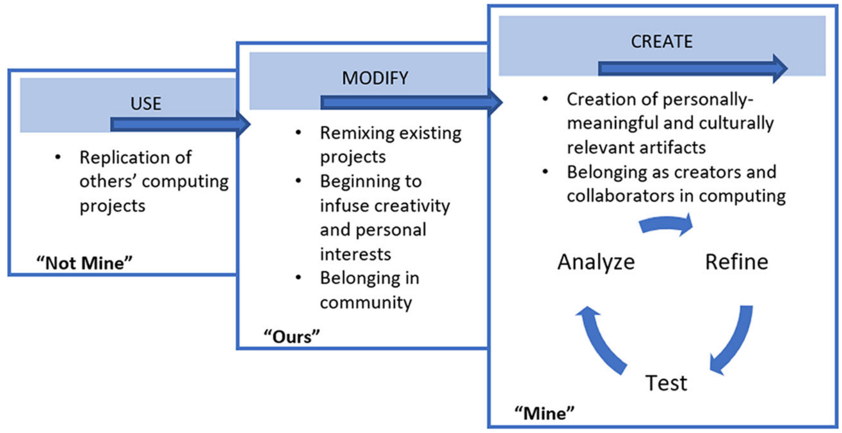


Fig. 1 Culturally Responsive Computing Framework Based on the Use-Modify-Create Progression (I. Lee et al. 2011)

their individual identities, cultures, and communities. Facilitators learn about the youth through their programming and creative engagement, subsequently using this knowledge to craft culturally responsive lesson plans. In this way, youth and facilitators become co-creators of the learning community.

In this work, we used the proposed culturally responsive computing framework in two ways: (a) as a blueprint for designing a computing program, the STC, in an informal setting (i.e., library); and (b) as an analytical lens that helped advance our investigation on youth's learning in such an environment.

Methods

Context of this Work

This work is situated in a larger effort to improve the teaching of computing in the U.S. through sustainable partnerships and a college field-experience course (Pollock et al. 2015). The course is open to undergraduates with at least one prior course in CS. Although participating undergraduates do not intend to pursue a teaching career, they enroll in the course to fulfill service-learning requirements and their desire to share their CS expertise with teachers and youth. The course combines college classroom meetings (facilitated by the authors) with field-experience in informal settings or schools.

During the college meetings, undergraduates and faculty in CS and education worked together to: (a) identify existing CT lessons, resources, and activities relevant to students' age group, interests, and prior experiences; (b) model pedagogical strategies for teaching CT that have shown promise in broadening participation in computing; (c) design computing lessons; and (d) reflect on the implementation of lessons during the field experience (Mouza et al. 2016). In addition, as part of the course undergraduates engaged in three hour-long culturally responsive training sessions throughout the semester. These sessions encouraged undergraduates to adopt a culturally validating and affirming attitude toward participating youth that promotes a sense of belonging and intentionally learn about youth's interests and cultural backgrounds (Ladson-Billings 1995; Mejias et al. 2018; Nieto 2002; Pollock 2008). In the field, undergraduates facilitated classroom activities or after-school programs with instructors at the partner sites (see Pollock et al. 2015; Mouza et al. 2016, 2020). Here, we examine *one such partnership* in an *after-school program at a local library,* which had been a partner for 3 years.

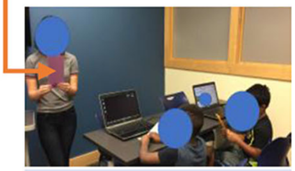
The STC was established at a local suburban library located in a Mid-Atlantic city. Each semester, for a period of 3 years, at least two college undergraduates enrolled in the field-experience course served as the STC program facilitators. The program was offered on Saturday mornings for 2 h over 11-weeks. The first two authors provided on-site support to undergraduate facilitators while serving as participant-observers during data collection. Any child (ages 6–14) interested in participating was encouraged to attend at any time of the program. Personnel at the library were responsible for advertising the program and recruiting participants, though walk-in participants were welcome at any time.

The design of the program was consistent with a design-based research (DBR) paradigm—an iterative and effortful process that requires program facilitators and researchers to consistently examine and explicitly articulate the refinement between learning designs and the designed goals (Design-Based Research Collective 2003). Each week, two university undergraduates who served as program facilitators collaborated with university faculty and researchers in education and CS (i.e., authors), as well as the local librarians, to discuss and analyze the previous weekly session and design upcoming STC activities drawing on effective learning and cultural practices (e.g., Lewis et al. 2019). Figure 2 presents an overview of key program activities. Considering the nature of the informal setting, which was characterized by unpredicted participation rates and new participants at any time point, each session started with facilitator-youth introductions or new-returning youth introductions and ended with the youth's programming project demonstrations, which their parents were invited to

Fig. 2 STC Program Overview of Key Activities: Week 1 (Introductions), Week 3 (Learning how to replicate programs), Week 6 (Robotics) and Week 11 (End of the program – connecting programming to real-life applications)

Week 1: Introduction to Computing, Programming and Programming Environment

Main Activity	• Create your name tag through Scratch. • CS-Unplugged activities related to algorithms. • Exploring coding using various resources.
Computing Tools and Materials	• Scratch • Web resources related to coding, such as Code.org • Relevant materials to support the CS-Unplugged activities.
Culturally Responsive Pedagogy	• Intentionally getting to know participants. • Creating a welcoming and engaging learning environment. • Collecting participants' feedback about what projects/goals that they intended to create/achieve from STC.
Computing Concepts	• Getting to know code and exploring Scratch (e.g., the looks and the sounds coding blocks).
CT Practices	• Using algorithmic thinking as a mean to solve problems.



The facilitator is conducting a CS-Unplugged activity related to algorithms.

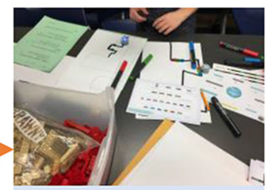
Week 3: Creating a Scratch Project: Replications and Remixing of Codes

Week 6: Programming with Difference Computational Tools -- Scratch and Ozobots

Week 11: Real-World Programming Uses -- Robotics, Design and Applications

Activities for each following sections were designed based on reflections and observations from facilitators and researchers following the DBR paradigm.

Main Activity	• Four station activities are designed as "mission tasks" to provide participants with real-life experience related to computing applications (e.g., computing your robots for a soccer game). • Each participant receives a mission package which contains detailed instructions for four station activities. • A facilitator oversees each station and facilitate participants' completion of a mission. If a participant finish the mission, he/she would be given a stamp on his/her mission package. • Participants who collect four stamps will receive a prize at the end of the section. • Participants will be given certificates to recognize their achievement during their participation at STC.
Computing Tools and Materials	• Scratch • Finch Robot • Ozobot • Makey Makey • Relevant materials to support participants' creations of computing artifacts, such as Lego blocks.
Culturally Responsive Pedagogy	• Cultivating fun and welcoming environment. • Infusing student ideas and interests into programming. • Developing sense of belonging. • Connecting CS to community and culture.
Computing Concepts	• Algorithms, Loops, Conditionals, Broadcasting, Variables
CT Practices	• Problem decomposition • Algorithmic thinking • Abstraction • Parallelization • Simulation • Pattern Recognition • Creativity



Participant is designing a path for the Ozobot to follow in order to deliver food in the context of a school cafeteria. The Lego blocks are used to build the context.

Participants are collaborating in the design of a robotic soccer game, while the facilitator is overseeing their progress.

attend. Further, participating youth were given choices to either work alone or collaborate with peers at STC.

Participants

Participants included 30 youth and two undergraduate program facilitators who participated in the study over the period of a single 11-week semester. Most youth (73%) were ages 8–10 years old. Both undergraduate program facilitators were female students with a STEM-related major/minor who have completed introductory coursework in CS. Considering that the STC program did not require mandatory participation, any youth who was interested in programming was welcome to join at any time. Therefore, participation rates for each STC session were unpredictable. In total, 30 youth attended at least one STC session throughout the 11-week period. Among these participants, 22 youth were categorized as *less frequent* participants (they participated in 1–5 sessions), while eight youth were categorized as *more frequent* participants (they participated in more than five sessions). More than 60% of participating youth did not have any prior programing experience. An overview of participating youth's demographic information is presented in Table 1. As shown in Table 1, participants were from diverse backgrounds; specifically, we grouped our African American and Hispanic participants into the underrepresented minority group ("URM") based on the URM definition by Museus et al. (2011). Ethnicity data were collected through direct observation but were verified by parents of participating youth.

Data Sources

Data for this work were collected from multiple sources over an 11-week period in one semester during the second year of the program.

Undergraduate Facilitator-Level Data Data included undergraduate facilitators' ($N = 2$) weekly lesson plans ($N = 11$) and reflective entries composed at the end of each Saturday session ($N = 21$). The two undergraduate program facilitators collaborated with the first author and the librarian in designing lessons for each session at STC. Each lesson plan included key activities that aimed to scaffold youth learning of foundational programing concepts, such as loops and variables. In addition, each lesson plan included resources (e.g., robotics, laptops) needed to implement the activities, as well as instructional strategies (e.g., pair programming, end-of-session project presentation, etc.). After each session, undergraduate facilitators submitted reflections discussing four aspects of their work: (a) a narrative description of the lesson taught; (b) a reflection of what went well and what did not go well; (c) any questions/concerns they had related to upcoming sessions, including anticipated challenges; and (d) specific needs (e.g., resources) for the upcoming session.

Program-Level Observation Data Data collected at the program-level included researchers' weekly fieldnotes ($N = 11$), videos, and pictures taken from each session (around 300 videos and pictures). The first author conducted all the fieldwork for this study as she has had "prolonged engagement in the field" (Creswell and Miller 2000, p.127) since the STC's establishment and was able to obtain pluralistic perspectives from the youth, their parents, and the librarians to establish in-depth understandings of the site and the participants. At each STC session, the observer recorded fieldnotes while also walking around the room to provide individualized assistance to youth. Although a formal observation protocol was not used, each observation documented learning activities, instructional strategies, and youth interactions.

To ensure the credibility and validity of the observational data, the observer employed member checking whereby she frequently checked with members within the community (e.g., youth, their parents/accompanied adults, and the program facilitators) to ensure the accuracy of the information recorded (Creswell and Miller 2000). Additionally, after each session the observer emailed a summary of the learning activities, youth's feedback, successes, and challenges to the librarians, university faculty members, and undergraduate facilitators, to ensure an accurate understanding of the lesson and its purpose.

Table 1 Gender and Race/Ethnicity of Participating Youth

		Total ($N = 30$)
Gender	Female	23%
	Male	77%
Race	Caucasian	37%
	Asian	37%
	Underrepresented Minority	26%

Youth-Level Data Data collected at the youth-level included programming artifacts created during each session ($N = 65$) as well as individual interviews ($N = 2$) conducted at the end of the 11-week program with two of the participants. Participants' programming artifacts were collected and stored immediately after each session. In addition to collecting artifacts, we invited high-frequent participants for interviews and two youth agreed to participate at the end of the program. These youth are presented in the analysis as case studies.

A two-phase protocol was used for the interviews with the two participating youth. Phase I focused on probing participants' learning experience at STC (Mouza et al. 2016, 2020) while phase II presented a scenario that asked participants to demonstrate knowledge of CT through a think-aloud approach (Pan et al. 2019). Sample questions for phase I included "What did you like the most about STC?" and "Why did you decide to come to STC?" The phase II interview procedure first asked youth to play a pre-made Scratch game called Bee Maze (Sullivan et al. 2012; see Fig. 3) without seeing any code behind this project. Subsequently, the researchers asked participants follow-up questions about Bee Maze while revealing the code at particular key points in the interview (Sullivan et al. 2012). These follow-up questions addressed both specific programming concepts and CT practices. For instance, participants were asked to describe what Bee Maze does (abstraction and problem decomposition) and what Scratch blocks could be used to replicate this game (problem decomposition and simulation). Consent forms were collected from both youth and their parents for all data collection and only data from those parents/youth who consented are included in this work.

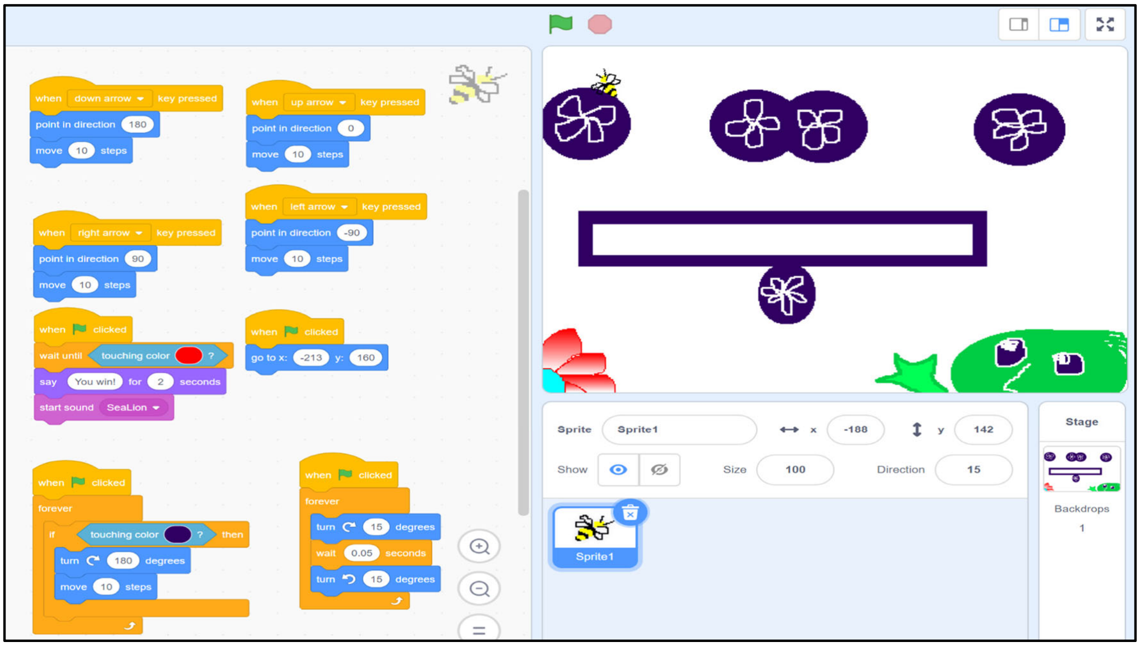


Fig. 3 Previously Developed Bee Maze Scratch Program (Sullivan et al. 2012)

Data Analysis

Program Level Data To answer the research questions, we selected to report a synthesized form of above-described data sources. First, we explored patterns of participation and peer collaboration in the STC in response to the first research question: *In what way did participation in the STC support peer learning and collaboration?* For this purpose, we performed social network analysis (SNA) with relational data which included lesson plans ($N = 11$), fieldnotes ($N = 11$), and programming artifacts ($N = 65$) to determine the degree of interactions among youth at STC.

In educational research, SNA has been employed to determine the human and social dynamics through analyzing the relational structures between entities (e.g., learners) within specific learning environments/communities (Martınez et al. 2003; Penuel et al. 2006). Specifically, SNA aims to determine how social influences contribute to explaining the outcomes of learners who shared similar network positions and patterns (Grunspan et al. 2014). In this study, we performed SNA using the igraph package in R (an open-sourced statistical computing toolkit) to understand the relationships among youth interactions (Csardi and Nepusz 2006; Grunspan et al. 2014).

SNA primarily relies on quantitative mapping of network relations; however, it does not provide a qualitative explanation underneath such relationships, including the actual interactions between the entities (Coviello 2005; Heath et al. 2009). Therefore, in addition to understanding the structures of the relational patterns among participating youth, we further examined the association between youth network positions and the frequency of their participation at STC throughout the 11-week period. We used this analysis along with the case study data (see Analysis of Case Studies) to answer the second research question: *How did participating youth progress in their development and understanding of CT (cognitive) as well as their sense of belonging (affective) within computing?*

Analysis of Case Studies After analyzing data collected from all students, we utilized collective evidence to present two cases illustrating how youth progressed in their cognitive and affective learning during their participation in the program. For each case, we followed the same analytical approach to aggregate a profile. We first analyzed the programming projects created by each case study participant, followed by an analysis of their interview data. Concurrently, we referred to the program-level data to triangulate our analyses.

Specifically, we examined the Scratch programming projects of the participants, including the project developed during their participation in the first and last session. To analyze these projects, we adopted a rubric originally developed by Denner et al. (2011) and adapted in our previous work (Mouza et al. 2016), which focused on (a) CS programming concepts represented in youth's products (i.e., loops, conditionals, parallelism, operators, data); (b) project structure (e.g., number of scripts, backdrops, types of sprites, interactivity); (c) code organization (e.g., variables named appropriately); and (d) usability. Three coders individually examined a sample of programming projects using the rubric to establish consistency in coding. The initial inter-rater reliability was 95.06%. The first author then coded all artifacts collected from case study participants.

To triangulate our scoring, youth's programming projects were analyzed through an automatic analytical system called "Dr. Scratch." Dr. Scratch automatically analyzes Scratch projects and produces scores in seven domains related to programming, which were consistent with the rubric we used for this study: Flow Control, Data Representation, Abstraction, User Interactivity, Synchronization, Parallelism and Logic. Dr. Scratch also provides an overall score, ranging from 0 to 21, and assigns the project to one of three levels: (a) Basic (scores between 0 and 7): the project uses introductory programming features; (b) Developing (scores between 8 and 14): the project includes intermediate programming functions; and (c) Master (scores between 15 and 21): the project is at the developed stage with advanced programming features (Moreno-León et al. 2015).

In addition to analyzing their products, we also analyzed interview data collected from the two case study participants. Interview data were analyzed qualitatively using a combination of a priori themes related to the study's questions (see also Yang et al. 2019) and themes that emerged during the interviews. The six themes included in the Phase I coding scheme are: (a) background information, (b) motivation for attending the STC program, (c) surprises with programming, (d) most enjoyable learning experiences, (e) challenges experienced, and (f) reflections on learning experiences. The three themes included in the Phase II coding scheme are: (a) functionalities of five coding blocks presented in the Bee Maze project, (b) most difficult code blocks to program the Bee Maze project, and (c) creativity of remixing the Bee Maze project.

Finally, results from the analysis were aggregated to establish two case studies that represent youth's profiles (Hatch 2002; Saldaña 2015). In particular, we used the culturally responsive computing framework as a lens to present youth's cognitive (e.g., use-modify-create) and affective (e.g., sense of belonging) representations.

Results

Participants' Peer Collaboration Patterns in the STC

In total, 30 youth participated in the STC program over the 11-week period. Among these youth, 25 had at least one experience collaborating with other peers. Figure 4, generated from the SNA indicates that STC served moderately diverse youth groups, including ethnicity and gender. In addition, two metrics from SNA were referred to further describe the degree of the interactions. As shown in Table 2, the degree centrality indicates the number of peers with whom each youth collaborated, while the betweenness centrality indicates the degree of belonging to different peer groups (Freeman 1977; Grunspan et al. 2014; Nieminen 1974). Pearson correlation tests were subsequently performed and the results revealed that there was a strong correlation between degree centrality and participation times ($r = 0.78$, $n = 25$, $p < 0.001$), as well as between participation times and betweenness centrality ($r = 0.71$, $n = 25$, $p < 0.001$) (Field 2013).

Results from SNA further indicate that major peer collaboration appeared during three capstone projects, which occurred in the middle and towards the end of the STC program. Among these collaborative interactions, participants were observed to form peer collaborations based on similar gender and ethnicity backgrounds. As shown in Fig. 4, most of the male and female youth were observed to include at least another peer with the same gender in collaborative work. Similar results were observed based on ethnicity backgrounds. Moreover, youth who were frequent participants or participated in the STC during a prior semester were involved with more groups compared to other participants. For instance, in Fig. 4(c) and (d) and Table 2, Andy, Morgan, Aubrey and Mason were centered in the social networks and connected to more peers compared to youth who were at the end branches, such as Damon and Lily.

In summary, SNA analyses present the patterns of how youth formed groups in the STC and reveal that several of the youth were connected to more communities than others based on their participation frequency and computing background.

Profiles of Youth's Affective and Cognitive Progression in Computing

In order to complement the quantitative analyses, we present two cases (Mason and Aubrey) to illustrate how and why youth chose to participate in STC more frequently by examining their progression in computing. Their progression, based on the culturally responsive computing framework (Fig. 1), contains two major components: cognitive and affective. The cognitive consists of youth's development and understanding of CT, while the affective reflects how youth valued their participation at STC, such as their sense of belonging. We first illustrate their affective progress focusing on collaboration and sense of belonging. Subsequently, we present their cognitive progress by focusing on the analysis of their computational products and their understanding of a pre-programmed Scratch project. Our analysis indicated that Mason reached the "Modify" stage while Aubrey reached the "Create" stage. The cases of Mason and Aubrey are presented below.

Mason's Case

Mason was an 8-year-old male from an URM group. When he entered the STC he had no prior background in programming. At the time of data collection, Mason was in his first semester of participation in the program. Mason was a regular participant in the program, attending seven out of 11 STC sessions.

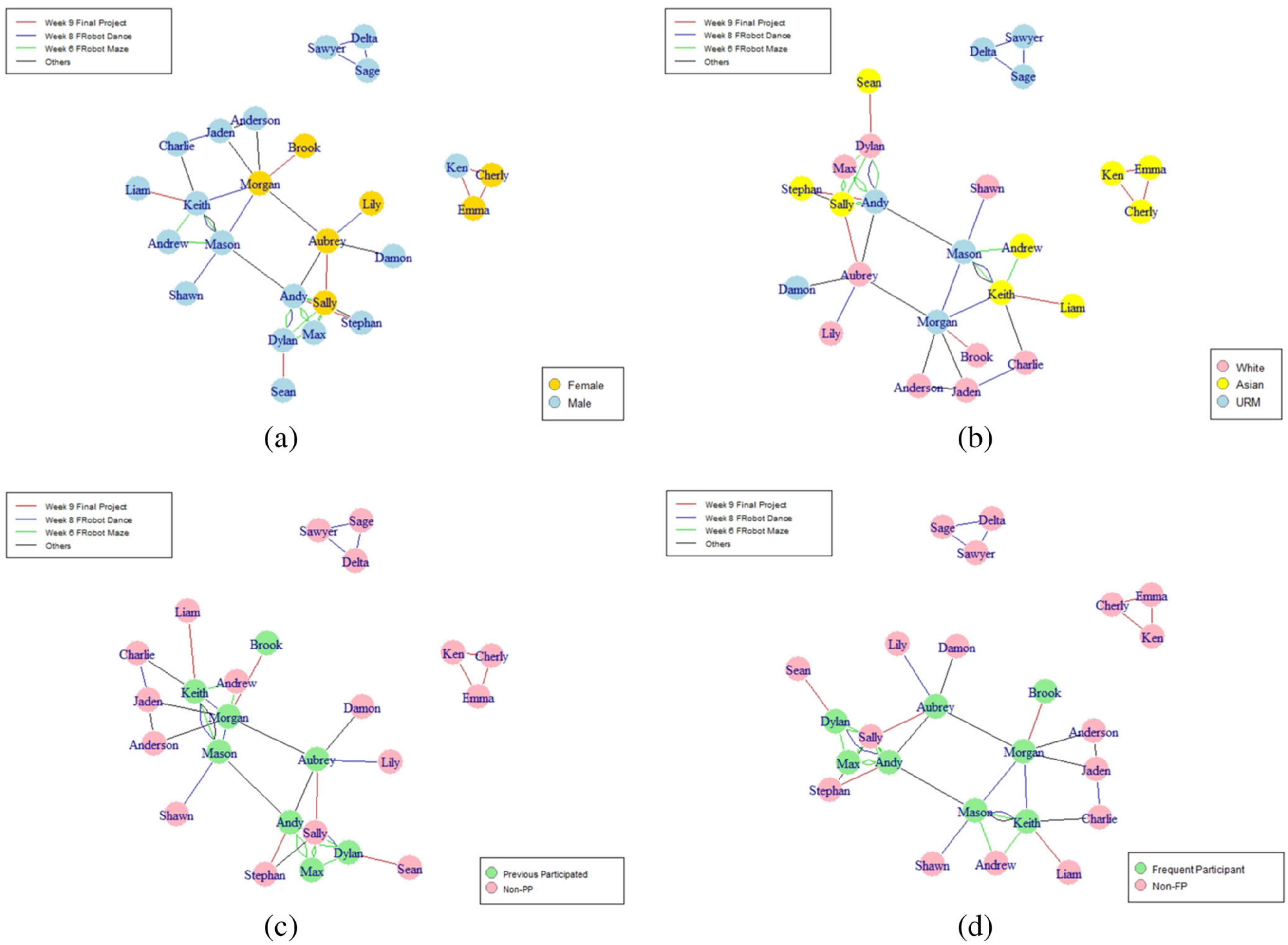


Fig. 4 Networks Developed Through Collaboration by Gender (**a**), by Ethnicity (**b**), by Participation record (**c**), and by Participation Frequency (**d**)

We present Mason's sense of belonging and CT learning through his progress from the "Use" to the "Modify" stage.

Belonging in the STC Community Mason entered the STC with some hesitation because it was his dad's idea. Without previous exposure to programming, Mason worked alone and initially just followed the exact Scratch code provided by the program facilitators, duplicating the same Scratch projects. After each replication, Mason and other youth were given time to create a new project or to add new features to their existing projects while receiving help from the program facilitators. During this time, Mason was often observed talking to other STC participants, frequently discussing his project plans or looking at their projects. Further, he started to make linkages between the programming code he was learning with what he liked to program (Fig. 5). His initial projects focused mostly on presenting his interests, including puppies and sports. In these projects, Mason mostly relied on using the built-in sprites on Scratch (Fig. 5a) but gradually tried to create sprites which were not available in the sprite library, such as images of his family members (Fig. 5b).

In Week 2, Mason remixed the project that he replicated from the program facilitators by adding some of his favorite images and colors for Halloween (see Table 3). Describing his project, he explained: "I like to pick things chasing each other so that ghost is going to chase that cat. The bat is flying around because it is scared. Red is my favorite color, so I put the background red. And dog is my favorite kind of animal".

Table 2 Degree Centrality and Betweenness Centrality from SNA *(N = 25)*

Degree Centrality	1	2	3	5	6	7	9
N	6	10	1	3	1	3	1
Betweenness Centrality	0	0.01–0.05	0.06–0.10	0.11–0.15	0.16–0.20	0.21 and above	
N	16	3	1	1	3	1	

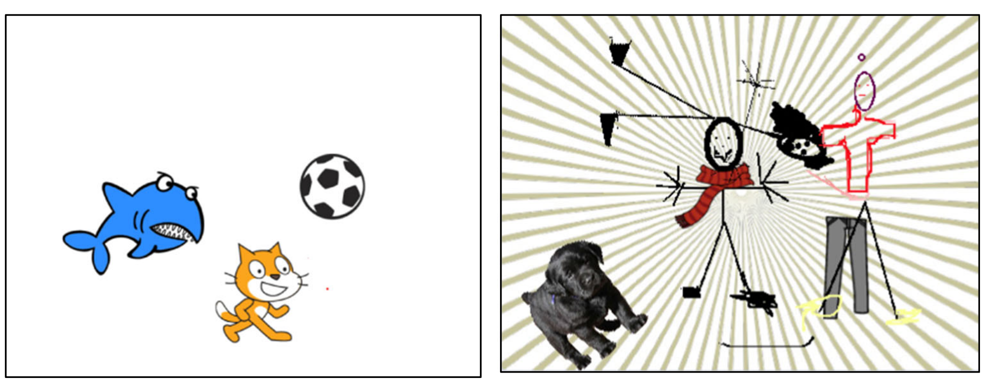
Compared with the replicated program, Mason duplicated the code then changed and tested it for his own project. This project later became the one he was "most proud of" in the entire program and acknowledged the moments of "making things move and programming them" as the "best part."

Mason's lack of previous programming experience did not prevent him from pursuing computing at STC. Instead, he started to form a sense of belonging in the STC community with the confidence of "I did things that I like!" In contrast to his participation in the early sessions at STC, which focused more on making programming projects by himself, Mason engaged in collaborative work in the three-week robotics unit. SNA results showed that Mason had worked with seven different partners (degree centrality = 7) at different groups (betweenness centrality = .19). Mason was found to actively communicate with his peers in the design and execution of the team projects. In particular, Mason joined the group which included the friends he made at STC. The group worked together to program a Finch Robot, a robot that could be programmed through Scratch, to complete different tasks such as going through mazes or participating in a robotic dance party. In one of the videos recorded during his group discussion, Mason and his group members completed the code for the Finch Robot and had a maze created for the robot to go through. However, the Finch Robot did not move as expected when they pressed certain arrow keys on the keyboard. Mason was the first one to realize that the left wheel of the robot did not move. Mason and his peers looked at their code on Scratch and the following conversation transpired:

> Mason: Let's see, this one (*pointed the left wheel of the Finch Robot*) is not working. Let's try [pressing the keys] again.
> *He then leaned himself towards the laptop which displays the code for the Finch Robot and pressed the left arrow to see the reactions of the Finch Robot.*
> Mason: Not moving.
> Group Member 1: Yes, because this is zero (*pointed to one of the numbers in under the "left arrow" code*).
> Mason held the robot and confirmed: That's a zero.
> Group Member 1: It's not supposed to move.
> Group Member 2: (*Presses other arrow keys and mumbles*) Why is it not working, it is impossible!
> Group Member 1: Because there is a zero.
> *To address Group Member 2's concerns, Mason put the Finch Robot back to the carpet and asked for his peers to press the arrow keys to test the robot again.*
> Group Member 2: (*Press the up arrow the left arrow*) Oh, wait, it is going the wrong way.
> Group Member 1 and 2 looked at the code again and Mason says, [it is time to make the robot] Turn Turn Turn!
> *However, the robot did not make a left turn.*

The video data revealed that Mason collaborated and contributed to the group work in problem-solving. In the three-week collaborative robotics tasks, Mason was found to enjoy talking and interacting with others. In fact, when Mason was asked what he liked the most about joining STC, he pointed out the collaboration and interaction with peers because "people talk, and I learn." Further, Mason was very excited to share his projects in front of the room at the end of each session while his dad was watching his presentation in the back of the room.

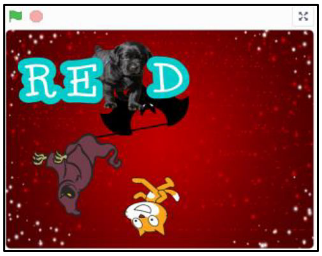
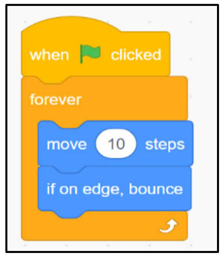
Developing towards a Remixer Throughout his participation in the STC, Mason created five individual programing projects. Three of his projects were animations while two were games. The average number of scripts in his projects were 11 lines. Figure 6 shows that Mason's projects covered most of the CS programming concepts introduced at STC. In particular, all of his projects addressed the programming concepts of "flow control," "user interactivity" and "abstraction." In contrast, "logic" and "synchronization" were not addressed in any of his own projects. Although Mason's projects covered most of the programming concepts, they were evaluated at the *beginner* level with an average score of 6. For instance,



(a) (b)

Fig. 5 Mason's Programming Projects Collected in a Sequential Order

Table 3 Mason's Remixed Coding Project at Week 2

<table>
<tr><th>Scratch Project Screen Shot</th><th>Replicated Code</th><th>Sample Remixed Features</th></tr>
<tr><td>
</td><td>
</td><td>S.1: Changed the numerical data at the motion block to test how changes of the data affected the speed of each sprite.
S.2: Added other features such as sound.</td></tr>
</table>

Mason's projects all had "Green flag(s)" and some of his later projects had "key pressed" functions. Such evidence shows that his projects demonstrated basic to developing understanding of the "user interactivity" concept. Comparing with his individual programing projects, Mason's collaborative project addressed other CS programming concepts as well, such as "synchronization" and "logic".

Mason demonstrated emerging CT practices in abstraction and problem decomposition during the second phase of his interview when he was asked to interact and interpret a pre-programmed Scratch project called Bee Maze. Mason was initially hesitant to start playing Bee Maze until he was told that he could try clicking the "green flag" to get started. After he was given one minute to explore the game without seeing its code, he was asked to explain the functionality of the game. He noted: "maybe like making the bee move around and maybe it hits one of these (purple flowers), it sometimes gets stuck a little, and you have to try to find out how to make it (bee) out." Subsequently, Mason envisioned how the game could be made based on this description. Mason was able to accurately point out what Scratch blocks could be used to make the bee move. He encountered some difficulties in addressing Scratch blocks that executed the statement of "if … [the bee] … bounce, if … [the bee] get stuck". When Mason was provided with the Scratch code, he could understand all of it; however, he experienced some challenges in interpreting the output of the code, such as which portion made the bee buzz. In terms of advancing the design of the game, Mason proposed several ideas such as adding more challenging features and giving more specific instructions. However, he did not mention how Scratch blocks would be potentially used for such redesigns.

In summary, Mason's case reflects a pathway from "Use" to "Modify". At the Use stage, Mason learned how to navigate a new programming environment and replicate the facilitators' programming projects. Based on this learning experience, he was given the opportunity to interact with his peers, which prompted his interests in remixing and redesigning projects. Throughout his participation in the STC, Mason exhibited a sense of belonging in the community reflected by his collaborative work with others and his eagerness to showcase his projects at the end of each session.

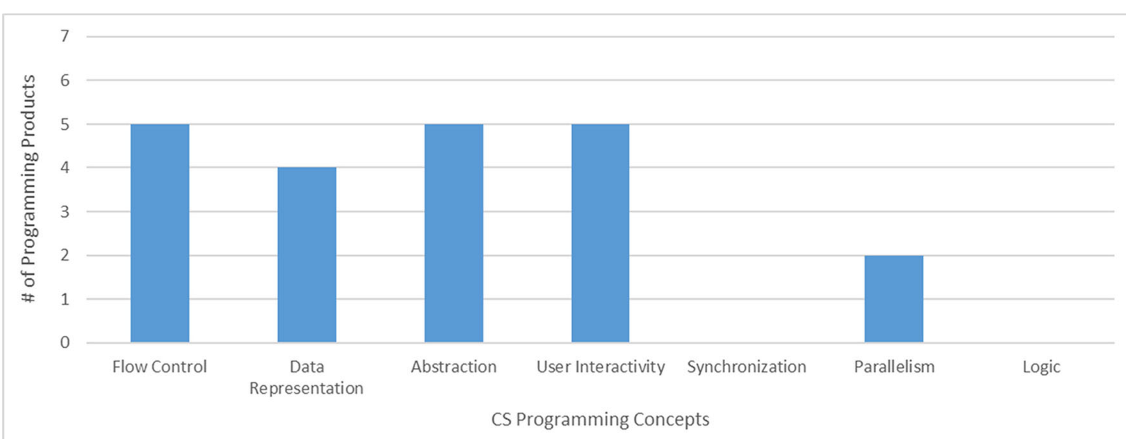


Fig. 6 CS Programming Concepts Reflected in Mason's Programming Projects

Aubrey's Case

Aubrey was a 10-year-old female Caucasian participant. Along with her sister, she had participated in STC in a previous semester. At the time of the study, she attended 6 out of 11 STC sessions. Aubrey always had ideas for computing projects using Scratch and utilized resources to program those ideas. In addition, she was observed collaborating with others very often during the program. We present Aubrey's sense of belonging and CT learning through her progress from the "Modify" to the "Create" stage.

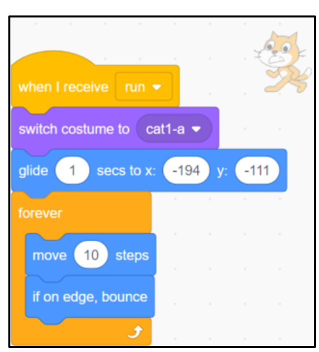
Belonging in Computing Aubrey came to STC for two semesters. She became a regular participant since then because "you just got to choose whatever you wanted to do." She had more experience with programming and she enjoyed making her own games. In her initial participation (Week 2) at STC during the current semester, she brought a project that she made at home which was about Halloween (Table 4) – one of her favorite holidays. She chose to finish this project alone but asked the facilitators questions towards the completion of the project.

Aubrey preferred to work with other peers to create new projects. The Scratch projects that she introduced during the interview were the ones she developed in collaboration with peers. One project was the Pacman game in which players could control the "cookie" sprite using the arrow keys to navigate the maze (see Fig. 7). Players have to avoid having the cookie touch the "ghosts" otherwise certain noises would play. Aubrey developed this idea on her way to STC and subsequently realized her partner was also interested in making a similar game. Their first step in designing the game was to make some ghosts with customizations: "one [ghost] is the Pikachu, and one has the mustache; we did a basketball one to represent Sally because she likes basketball and we did a green one because one of the other people in our group [table] likes the color green. Then, someone else likes the color blue, and someone else likes the color orange. And I like the cookie monster, so I am the cookie monster one. And we made our Pacman a cookie." The design of the ghosts, however, required extensive time commitment from Aubrey and her collaborators. Through their collaborative effort they developed a method to address that: "[We] tried to duplicate them and that worked a lot quicker." Later on, they also encountered difficulties in testing their code and debugging the portion of the code that did not work.

The challenges they faced on debugging code did not prevent Aubrey from becoming a regular participant at STC and she really enjoyed "…that we got a lot of free choices to do random/whatever projects we want, with the same stuff [resources and programming environment], and then everyone ['s projects] turned out different[ly]." In fact, SNA results demonstrated that Aubrey had worked with 5 different partners (degree centrality = 5) at different groups (betweenness centrality = .18). Aubrey's group projects were found to either apply more ideas in remixing existing programs or creating new projects through collaboration with peers. For instance, after the program facilitators demonstrated how to create a pong game on Scratch, Aubrey and her partner decided to create an air hockey game that they both liked to play based on the pong game idea and associated code (Fig. 8a). Later on, Aubrey worked with a new participant, who did not have any programming background, on creating an animation in Scratch and helped the new participant draw two sprites during their collaboration (Fig. 8b).

Moreover, Aubrey always liked to bring new ideas to her group. At Week 4, when Aubrey worked with a partner in developing a Scratch project which could be executed with the Makey Makey toolkit (a kit designed to connect everyday objects to computer keys), they first developed a project

Table 4 Aubrey's Coding Project at Week 2

<table>
<tr><th>Scratch Project Screen Shot</th><th>Sample Code</th><th>CS Programming Concepts</th></tr>
<tr><td></td><td>
</td><td>Flow control (e.g., using the forever function).

Synchronization (e.g., using the "when I receive (message)" function).</td></tr>
</table>

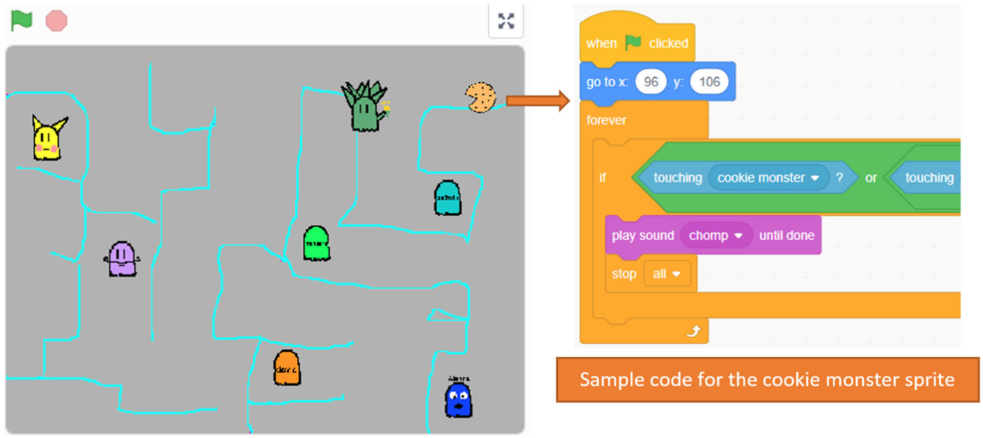


Fig. 7 Pacman Project Screenshot (left) and Sample Code (right)

similar to the one (e.g., piano) demonstrated by the program facilitators. Later on, they created a new project that included the clone feature. Their second project was praised by many of their peers and their parents at the end of the session.

Developing towards Creator Aubrey's first project demonstrated her fundamental understandings in programming. Her collaborative projects also involved a number of programming concepts and an increased number of scripts. Figure 9 shows that Aubrey's projects covered all CS programming concepts with stronger representations on "synchronization," "parallelism," and "logic". Moreover, Aubrey's programming projects were evaluated at the developing level with an average score of 9. For instance, one of her projects created clones of a Scratch cat which demonstrated a proficiency level of "abstraction" (Fig. 10).

Aubrey demonstrated a number of CT practices during the second phase of the interview, including abstraction, creativity, simulation, and problem decomposition. Aubrey jumped right ahead in exploring Bee Maze without being told where to begin. She quickly understood the rules of the game and made a specific plan in replicating it. She pinpointed specific Scratch blocks for making the bee "buzz," explaining: "I will make the bee to be able to move, by saying, like, when the up arrow is pressed, then change x by like 5, or y by 5. If the down arrow is pressed, change y by -5, and if the right arrow is pressed, then you would change the x by -5, and if the left arrow is pressed, change x by 5." When she was shown the code inside the game, she was fluent in interpreting the functionality of each code block until she encountered the portion of the code that aimed to make the bee "buzz." This code block was different than what she previously assumed would make the bee buzz. She looked at those specific Scratch blocks without immediately saying their output. She thought a while then recognized that "this is making the bee turn back and forth." When she was asked to add more functionality to the game, she suggested adding a scoreboard using the variable Scratch block and adding a timer to make the game more challenging.

In summary, Aubrey's case reflects a computing pathway from "Modify" to "Create". At the Modify stage, Aubrey learned code blocks and developed ideas that could be recreated in other programming projects. Based on this new knowledge, she was able to become a collaborator and a creator of new original projects. Throughout her participation in STC, Aubrey demonstrated a sense of belonging in computing

(a) pong game (b) sprite design

Fig. 8 The Programming Project Created by Aubrey and her Collaborators

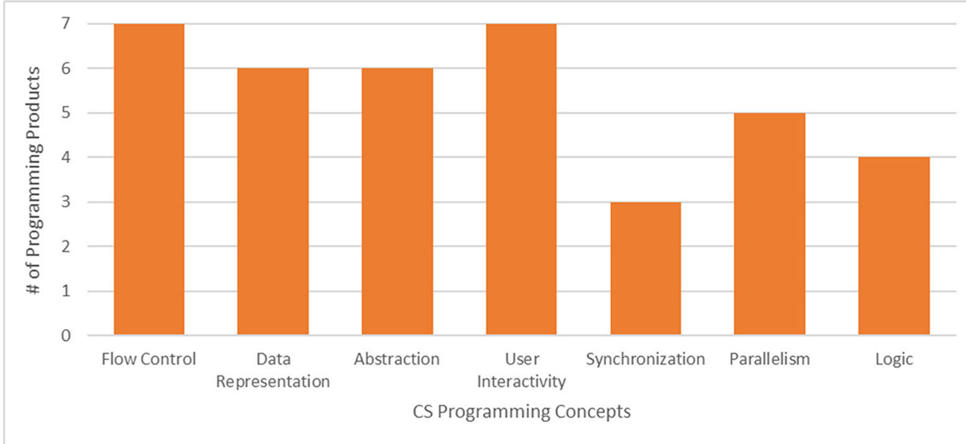


Fig. 9 CS Programming Concepts Reflected in Aubrey's Programming Projects

based on her projects and her identity as the one who can code whatever she wants.

Limitations

There are two limitations associated with this work. First, data were collected over the period of one semester. Therefore, they may not be representative of continuous and sustained trends in youth peer collaboration, including peers and communities youth chose over time. Second, interview data on youth trajectories were only collected from two participants who attended the STC frequently. Therefore, the affective and cognitive growth they exhibited may not reflect generalizable trends. Future studies should collect data over a longer period to examine the trajectory of participating youth as well as their identity development over time.

Discussion and Conclusion

In this paper, we examined youth's learning in an informal computing program through a library-university partnership. In particular, we introduced and illustrated a culturally responsive computing framework which served as a foundation for the design of the STC in a library. Furthermore, we showcased how youth who attended STC frequently progressed in their CT understanding and sense of belonging in computing. Supporting youth's sense of belonging both in the field of programming and their affiliated community is important for improving their attitudes and identities towards computing (Veilleux et al. 2013).

Findings of this work revealed that youth formed a variety of learning communities during the collaborative development of computing artifacts. Frequent participants, such as Aubrey and Mason, worked with a greater number of peers compared to less frequent participants. Such evidence revealed that youth's progression in computing often involved both cognitive (e.g., growth in CT knowledge and skills) and affective (e.g., peer collaborations, sense of belonging) development as shown in our culturally responsive computing framework. Moreover, the SNA results indicate that youth preferred to form groups with shared gender and ethnic background. As discussed, female youth tended to form groups with other female peers, while minority and Asian youth tended to work with peers from similar backgrounds. This finding is similar to previous research in group work, which indicates that learners tend to form social relationships with those who have shared experience or culture (e.g., Rienties et al. 2013). Additionally, similar to prior work, results of this study emphasize the importance for youth to construct personal meaningful artifacts in a culturally responsive learning context – youth's sense of belonging to STC progressed through their interactions and programming with peers from their supportive communities regardless of their computing background (Csizmadia et al. 2019; Feurzeig et al. 2011).

Findings from the two case studies illustrated how youth's sense of belonging and CT development evolved through

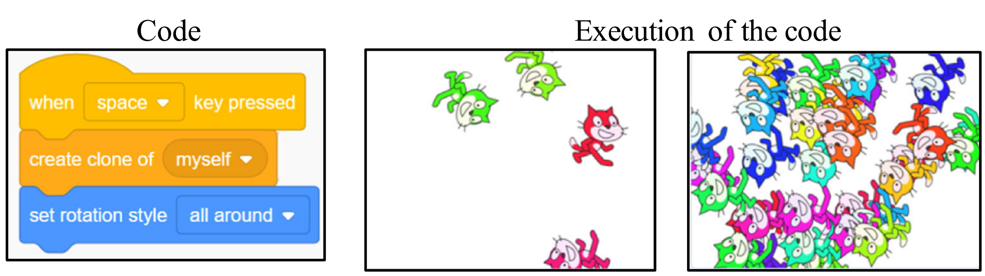


Fig. 10 The Execution of a Sample Code Block from Aubrey's Programming Project

collaborative work at different stages. Mason, who was novice and new to STC, initially focused on developing CT skills then progressively engaging with collaborative work. Aubrey, who was more knowledgeable among other youth at STC, was involved in more collaborative work and demonstrated advanced CT skills. In fact, the cases of Mason and Aubrey reflected that effective collaboration involved certain CT skills and engagement with the learning community. Further, findings indicated that Mason and Aubrey obtained new knowledge in terms of programming skills or ideas through collaborative work, which encouraged them to develop a sense of belonging in computing (Kafai et al. 2008). The cases of Mason and Aubrey highlight the importance of a culturally responsive computing framework, which illustrates how youth simultaneously progress cognitively and affectively towards computing. These cases also reveal the role of informal environments which provide youth with choices and autonomy in constructing their work with peers.

Moreover, synthesized evidence from this study reflects the potential of using public libraries as educational hubs and social places to establish computing opportunities that help broaden participation in computing. Specifically, we argued that using the culturally responsive computing framework helps researchers and practitioners unpack “learning” in informal settings, placing an emphasis on fostering youth’s sense of belonging (to the community and the field of computing) by actively engaging them into collaborative work and the design of personally-meaningful artifacts.

Finally, our work highlights the importance of instructional practices rooted in sociocultural frameworks. Yet many of the culturally relevant efforts in STEM education focus on addressing actions within specific classroom settings and have not been able to provide a practical model that could be widely adopted in other settings (Kraemer et al. 2019; Lachney et al. 2019). The *Culturally Responsive Computing Framework*, introduced in this work, could potentially contribute to providing a practical model for practitioners that promotes equitable and inclusive culture within formal and informal settings. Specifically, the framework bridges learning of CT (the cognitive aspect) with sense of belonging (the affective aspect) into a three-stage inter-related continuum. Students’ sense of belonging is a key predictor for students’ academic success and persistence in STEM, especially for females and minorities (e.g., Lachney and Yadav 2020; Nakajima and Goode 2020). Yet measuring student sense of belonging has been challenging (e.g., Johnson and Elliott 2020). In this work, we illustrate participants’ sense of belonging through patterns in their collaborative networks as they simultaneously acquired CT knowledge and skills. Therefore, our framework and study design has promising implications for capturing students’ sense of belonging and expanding participation in computing.

In terms of future work, we seek to ground the culturally responsive computing framework into the design of informal experiences in settings that dominantly serve underrepresented populations (e.g., Boys & Girls clubs) to reach and broaden diversity in computing. Ultimately, we aim to provide empirical guidance and professional development models for libraries and other informal spaces on how to develop effective programs that help broaden participation and diversity in computing.

Acknowledgements We acknowledge Dr. Florence Sullivan of the University of Massachusetts, Amherst for providing us with the Bee Maze Scratch project and accompanied interview guide. This work is supported by the National Science Foundation (Awards # 1639649 and # 1649224). All opinions are the authors and do not necessarily represent those of the funding agency.

Compliance with Ethical Standards

All procedures performed in this study were conducted in accordance with the ethical standards of the institutional review board (IRB) and with the 1964 Helsinki declaration and its later amendments or comparable ethical standards. The methodology and research instruments for this study were approved by the IRB at the University of Delaware.

Conflict of Interest Research reported in this article was supported by National Science Foundation under award numbers 1,649,224 and 1,639,649 to Lori Pollock (PI) and Chrystalla Mouza (co-PI).

References

Ahn, J., Subramaniam, M., Bonsignore, E., Pellicone, A., Waugh, A., & Yip, J. (2014). I Want to be a game designer or scientist: Connected learning and developing identities with urban, African-American youth. In J. L. Polman, E. A. Kyza, D. K. O’Neill, I. Tabak, W. R. Penuel, A. S. Jurow, et al. (Eds.), *Learning and becoming in practice: The International Conference of the Learning Sciences (ICLS) 2014* (pp. 657–664). Colorado, CO: International Society of the Learning Sciences.

Anderson, N. D. (2016). A call for computational thinking in undergraduate psychology. *Psychology Learning & Teaching, 15*, 226–234.

Angeli, C., Voogt, J., Fluck, A., Webb, M., Cox, M., Malyn-Smith, J., & Zagami, J. (2016). A K-6 computational thinking curriculum framework: Implications for teacher knowledge. *Educational Technology & Society, 19*(3), 47–57.

Atmatzidou, S., Demetriadis, S. (2014). How to support students’ computational thinking skills in educational robotics activities. *Proceedings of 4th International Workshop “Teaching Robotics & Teaching with Robotics” & 5th International Conference “Robotics in Education”*, Padova (Italy) July 18, 2014.

Barr, V., & Stephenson, C. (2011). Bringing computational thinking to K-12: What is involved and what is the role of the computer science education community? *ACM Inroads, 2*(1), 48–54. https://doi.org/10.1145/1929887.1929905.

Barton, A. C., & Tan, E. (2010). We be burnin’! Agency, identity, and science learning. *The Journal of the Learning Sciences, 19*(2), 187–229.

Bers, M. U., Flannery, L., Kazakoff, E. R., & Sullivan, A. (2014). Computational thinking and tinkering: Exploration of an early childhood robotics curriculum. *Computers & Education, 72*, 145–157. https://doi.org/10.1016/j.compedu.2013.10.020.

Bliksten, P., & Moghadam, S. H. (2019). Computing education: Literature review and voices from the field. In S. A. Fincher & A. V. Robin (Eds.), *The Cambridge handbook of computing education research* (pp. 56–78). Cambridge University Press.

Brady, C., Orton, K., Weintrop, D., Anton, G., Rodriguez, S., & Wilensky, U. (2017). All roads lead to computing: Making, participatory simulations, and social computing as pathways to computer science. *IEEE Transactions on Education, 60*(1), 59–66.

Braun, L., & Visser, M. (2017). *Read to code: Connecting youth to CS opportunity through libraries*. American library association: Office for information technology policy. Retrieved from http://www.ala.org/advocacy/sites/ala.org.advocacy/files/content/pp/Ready_To_Code_Report_FINAL.pdf.

Brennan, K., & Resnick, M. (2012). Using artifact-based interviews to study thedevelopment of computational thinking in interactive media design. Paper presented at annual *American Educational Research Association* meeting, April 13–17. BC, Canada: Vancouver.

Codding, D., Mouza, C., Rolon-Dow, R., & Pollock, L. (2019). Positionality and belonging: Analyzing an informally situated and culturally responsive computer science program. In Proceedings of *8th Annual Conference on Maker Education* (FabLearn '19). Mar 9–10. New York: ACM.

Coviello, N. E. (2005). Integrating qualitative and quantitative techniques in network analysis. *Qualitative Market Research: An International Journal, 8*(1), 39–60.

Creswell, J. W., & Miller, D. L. (2000). Determining validity in qualitative inquiry. *Theory Into Practice, 39*(3), 124–130.

Csardi, G., & Nepusz, T. (2006). The igraph software package for complex network research. *InterJournal, Complex Systems, 1695*(5), 1–9.

Csizmadia, A., Standl, B., & Waite, J. (2019). Integrating the constructionist learning theory with computational thinking classroom activities. *Informatics in Education, 18*(1), 41–67.

CSTA & ISTE (2011). *Computational thinking: Teacher resources.* http://csta.acm.org/Curriculum/sub/CurrFiles/472.11CTTeacherResources_2ed-SP-vF.pdf.

Denner, J., Werner, L., & Ortiz, E. (2011). Computer games created by middle school girls: Can they be used to measure understanding of computer science concepts? *Computers and Education, 58*, 240–259.

Design-Based Research Collective (2003). Design-based research: An emerging paradigm for educational inquiry. *Educational Researcher, 32*(1), 5–8.

Eglash, R., Gilbert, J. E., & Foster, E. (2013). Toward culturally responsive computing education. *Communications of the ACM, 56*(7), 33–36.

Estrada, M., Hernandez, P. R., & Schultz, P. W. (2018). A longitudinal study of how quality mentorship and research experience integrate underrepresented minorities into STEM careers. *CBE—Life Sciences Education, 17*(1), ar9.

Feurzeig, W., Papert, S. A., & Lawler, B. (2011). Programming-languages as a conceptual framework for teaching mathematics. *Interactive Learning Environments, 19*(5), 487–501.

Field, A. (2013). *Discovering statistics* (4th ed.). SAGE Publications: Sage.

Freeman, L. C. (1977). A set of measures of centrality based on betweenness. *Sociometry, 40*(1), 35–41.

Goodenow, C. (1993). The psychological sense of school membership among adolescents: Scale development and educational correlates. *Psychology in the Schools, 30*(1), 79–90.

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

Grunspan, D. Z., Wiggins, B. L., & Goodreau, S. M. (2014). Understanding classrooms through social network analysis: A primer for social network analysis in education research. *CBE—Life Sciences Education, 13*(2), 167–178.

Hatch, J. A. (2002). *Doing qualitative research in education settings*. New York: State University of New York Press.

Heath, S., Fuller, A., & Johnston, B. (2009). Chasing shadows: Defining network boundaries in qualitative social network analysis. *Qualitative Research, 9*(5), 645–661.

Ito, M., Gutierrez, K. D., Livingstone, S., Penuel, W. R., Rhodes, J. E., Salen, K., et al. (2013). *Connected learning: An agenda for research and design, research, and practice*. Irvine: Digital Media and Learning Research Hub.

Johnson, A., & Elliott, S. (2020). Culturally relevant pedagogy: A model to guide cultural transformation in STEM departments. *Journal of Microbiology & Biology Education, 21*(1), 1–12.

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

Kafai, Y., Desai, S., Peppler, K. A., Chiu, G. M., & Moya, J. (2008). Mentoring partnership in a community technology Centre: A constructionist approach for fostering equitable service learning. *Mentoring & Tutoring: Partnership in Learning, 16*(2), 191–205.

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

Koh, K., Abbas, J., Willett, R. (2019). Makerspaces in libraries: Social roles and community engagement. In V.R. Lee and A.L. Phillips (Eds), *Reconceptualizing libraries: Perspectives form the Information and Learning sciences* (pp. 17–36). Taylor & Francis.

Kraemer, E. T., Sitaraman, M., Che, S. M. (2019). How can we engage in inclusive, culturally responsive computer science?. In *Proceedings of the 50th New York: ACM Technical Symposium on Computer Science Education* (pp. 1251–1251). ACM. https://doi.org/10.1145/3287324.3293743.

Lachney, M., & Yadav, A. (2020). Computing and community in formal education. *Communications of the ACM, 63*(3), 18–21.

Lachney, M., Babbitt, W., Bennett, A., & Eglash, R. (2019). Generative computing: African-American cosmetology as a link between computing education and community wealth. *Interactive Learning Environments*. https://doi.org/10.1080/10494820.2019.1636087.

Ladson-Billings, G. (1995). But that's just good teaching! The case for culturally relevant pedagogy. *Theory Into Practice, 34*(3), 159–165.

Lee, V.R. (2019). Libraries will be essential to the smart and connected communities of the future. In V.R. Lee and A.L. Phillips (Eds), *Reconceptualizing libraries: Perspectives form the Information and Learning sciences* (pp. 9–16). Taylor & Francis.

Lee, V.R., & Phillips, A.L. (2019). Introduction. In V.R. Lee and A.L. Phillips (Eds), *Reconceptualizing libraries: Perspectives form the Information and Learning sciences* (pp. 1–6). Taylor & Francis.

Lee, I., Martin, F., Denner, J., Coulter, B., Allan, W., Erickson, J., et al. (2011). Computational thinking for youth in practice. *ACM Inroads, 2*(1), 32–37.

Lee, V. R., Tzou, C., Bang, M., Bell, P., Stromholt, S., Price, N., et al. (2017). Libraries as emerging spaces for computer-supported collaborative learning in schools and communities. In B. K. Smith, M. Borge, K. Y. Lim, & E. Mercier (Eds.), *Making a difference: Prioritizing equity and access in CSCL*, 12th Internnational Conferencce on Computer Supported Collaborative Learning (CSCL), Volume 2. Philadelphia: ISLS.

Lewis, C., Shah, N., Falkner, K. (2019). Equity and diversity. In S. Fincher and A. V Robins (Eds), *The Cambridge handbook of computing education research* (pp. 481–510). Cambridge University Press.

Maestas, R., Vaquera, G. S., & Zehr, L. M. (2007). Factors impacting sense of belonging at a Hispanic-serving institution. *Journal of Hispanic Higher Education, 6*(3), 237–256.

Margolis, J., Estrella, R., Goode, J., Holme, J. J., & Nao, K. (2017). *Stuck in the shallow end: Education, race, and computing*. Cambridge MA: MIT Press.

Martin, C. (2019). Designing for STEM in libraries serving underserved communities. In V.R. Lee and A.L. Phillips (Eds), *Reconceptualizing libraries: Perspectives form the Information and Learning sciences* (pp. 123–139). Taylor & Francis.

Martınez, A., Dimitriadis, Y., Rubia, B., Gómez, E., & De La Fuente, P. (2003). Combining qualitative evaluation and social network analysis for the study of classroom social interactions. *Computers & Education, 41*(4), 353–368.

Mejias, M., Jean-Pierre, K., Burge, L., Washington, G. (2018). Culturally relevant CS pedagogy theory & practice. In *2018 Research on Equity and Sustained Participation in Engineering, Computing, and Technology (RESPECT)* (pp. 1–5). IEEE.

Moreno-León, J., Robles, G., & Román-González, M. (2015). Dr. scratch: Automatic analysis of scratch projects to assess and foster computational thinking. *RED. Revista de Educación a Distancia, 46*, 1–23.

Mouza, C., Marzocchi, A., Pan, Y. C., & Pollock, L. (2016). Development, implementation, and outcomes of an equitable computer science after-school program: Findings from middle-school students. *Journal of Research on Technology in Education, 48*(2), 84–104.

Mouza, C., Yadav, A., & Leftwich, A. (2018). Developing computationally literate teachers: Current perspectives and future directions for teacher preparation in computing education. *Journal of Technology and Teacher Education, 26*(3), 333–352.

Mouza, C., Pan, Y. C., Yang, H., & Pollock, L. (2020). A multiyear investigation of student computational thinking concepts, practices, and perspectives in an after-school computing program. *Journal of Educational Computing Research, 58*(5), 1029–1056.

Museus, S., Palmer, R., Davis, R., & Maramba, D. (2011). Special issue: Racial and ethnic minority students' success in STEM. *ASHE Higher Education Report, 36*(6), 1–140.

Nakajima, T. M., & Goode, J. (2020). Lighting up learning: Teachers' pedagogical approaches for mak (e) ing computing culturally responsive in electronic-textiles classrooms. *Computing in Science & Engineering, 22*(5), 41–50.

Nieminen, J. (1974). On the centrality in a graph. *Scandinavian Journal of Psychology, 15*(1), 332–336.

Nieto, S. (2002). *Language, culture, and teaching: Critical perspectives for a new century*. Mahwah: Erlbaum.

Pan, Y., Yang, H., Mouza, C., Pollock, L. (2019). Employing a think-aloud approach to assess young learners' computational thinking. In *Annual Meeting of the American Educational Research Association*.

Papert, S. (1980). *Mindstorms*. New York: Basic Books.

Penuel, W. R., Sussex, W., Korbak, C., & Hoadley, C. (2006). Investigating the potential of using social network analysis in educational evaluation. *American Journal of Evaluation, 27*(4), 437–451.

Penuel, W.R., Chang-Order, J., Michalchik, V. (2019). Using research-practice partnerships to support interest-related learning in libraries. In V.R. Lee and A.L. Phillips (Eds), *Reconceptualizing libraries: Perspectives form the Information and Learning sciences* (pp. 239–256). Taylor & Francis.

Peter, L., Christie, E., Cochran, M., Dunn, D., Elk, L., Fields, E., et al. (2003). Assessing the impact of total immersion on Cherokee language revitalization: A culturally responsive, participatory approach. In J. Reyhner, O. Trujillo, R. L. Carrasco, & L. Lockard (Eds.), *Nurturing native languages* (pp. 7–23). Flagstaff: Northern Arizona University.

Pollock, M. (2008). From shallow to deep: Toward a thorough cultural analysis of school achievement patterns. *Anthropology & Education Quarterly, 39*(4), 369–380.

Pollock, L., Mouza, C., Atlas, J., Harvey, T. (2015). Field experience in teaching computer science: Course organization and reflections. Proceedings of *Special Interest Group in Computer Science Education*, march 4–7, Kansas City, MO.

Rienties, B., Héliot, Y., & Jindal-Snape, D. (2013). Understanding social learning relations of international students in a large classroom using social network analysis. *Higher Education, 66*(4), 489–504.

Rogowski, A., Recker, M., Lee, V. R. (2018). Designing online support guides for librarians managing STEM maker activities. International Journal on Innovations in Online Education, 2(4).

Roque, R. & Jain, R. (2018). Becoming Facilitators of Creative Computing in Out-of-School Settings. In Kay, J. and Luckin, R. (Eds.) *Rethinking Learning in the Digital Age: Making the Learning Sciences Count. Proceedings of International Conference of the Learning Sciences (ICLS) 2018,* volume 1. London: International Society of the Learning Sciences.

Ryoo, J. J. (2019). Pedagogy that supports computer science for all. *ACM Transactions on Computing Education, 19*(4) article 36, 23.

Saldaña, J. (2015). *The coding manual for qualitative researchers*. Sage.

Shute, V. J., Sun, C., & Asbell-Clarke, J. (2017). Demystifying computational thinking. *Educational Research Review, 22*, 142–158.

Sullivan, F. R., Hamilton, C. E., & Foley, A. (2012). *Scratch: An entrée into computational literacy and learning in young adolescents. Roundtable presentation at the annual meeting of the American Educational Research Association*. Canada: Vancouver.

Tzou, C., Bell, P., Bang, M., Kuver, R., Twito, A., & Braun, A. (2019). Building expansive family STEAM programming through participatory design research. In V.R. Lee and A.L. Phillips (Eds), *Reconceptualizing libraries: Perspectives form the Information and Learning sciences* (pp. 56–77). Taylor & Francis.

Veilleux, N., Bates, R., Jones, D., Allendoerfer, C., Crawford, J. (2012). The role of belonging in engagement, retention and persistence in computer science. In *Proceedings of the 43rd ACM technical symposium on Computer Science Education* (pp. 707–707). ACM.

Veilleux, N., Bates, R., Allendoerfer, C., Jones, D., Crawford, J., Floyd Smith, T. (2013). The relationship between belonging and ability in computer science. In *Proceeding of the 44th ACM technical symposium on Computer science education* (pp. 65-70). ACM.

Vygotsky, L. S. (1978). *Mind in society: The development of higher psychological processes*. Cambridge: Harvard University Press.

Wing, J.M. (2006). Computational thinking. Communications of the ACM.

Yadav, A., Krist, C., Good, J., & Caeli, E. N. (2018). Computational thinking in elementary classrooms: Measuring teacher understanding of computational ideas for teaching science. *Computer Science Education, 28*(4), 371–400.

Yang, H., Mouza, C., & Pollock, L. (2019). Establishing equitable computing programs in informal spaces: Program design, implementation and outcomes. *CoolThink@ JC, 197*.