Youth Civic Engagement Through Computing: Cases and Implications

By Junnan Yu, Janet Ruppert, Ricarose Roque, Ben Kirshner, *University of Colorado Boulder*

As a powerful tool for creating content, computer programming has been employed by some civic programs to engage youth in examining and challenging sociopolitical injustices. We present a case review of youth civic engagement programs (N=6) based on the lens of critical computational literacy (CCL), a conceptual framework that calls for people to challenge social injustices using computational skills. Employing Structured Cultural-Historical Activity Theory, we examine the program features and how participants were exposed to the computational and critical aspects of CCL. We further reflect on design implications for civic researchers, organizations, and computing educators to cultivate CCL in young people.

INTRODUCTION

With the exponential growth of the internet and various digital media, Information and Communication Technologies (ICTs) have created new modes of civic engagement and novel concepts such as digital citizenship [28]. While participation in digital media has generally shown a positive correlation with engagement in offline civic and political activities [3], the nature and quality of young peoples' digital civic engagement are still under investigation and debate. Research indicates that teenagers' social media use increases their political interactions and engagement with political information (e.g., viewing and sharing) but rarely results in producing their own political media or taking collective actions (e.g., creating videos and organizing protests) [8]. Such consumption-oriented civic engagement is criticized by some social scientists as an inferior mode of participation [51] or ineffective participation [38], which may further disengage young people from traditional venues of political engagement like voting.

Compared with being ICT consumers, young people seldom take on the role of ICT creator for civic engagement [31]. The 1% rule claims that in an internet community, only 1% users actively create new content, 10% will interact with the content like commenting, while 89% will simply view it without further interaction [1]. However, designing and building ICTs allows young people to develop fluency with computational tools and gain a behind-the-curtain understanding of how ICTs operate. Advocates for civic engagement through ICT creation argue that it provides a more holistic and critical understanding of a world where technology increasingly undergirds systems of power [47,48]. Computer programming, which enables us to build media projects like websites and video games, is a particularly powerful tool for ICT creation. Recently, some digital platforms have been designed for young people to learn computational thinking and skills by creating interactive media projects, such as Scratch and MIT App Inventor. When creating projects on these platforms, some young people explored socio-political issues and expressed their civic opinions through their projects (e.g., [36]).

To frame and theorize such cases that combine computational and critical literacy, Lee and Garcia [23] proposed the Critical Computational Literacy (CCL) framework. CCL is a lens through which to examine and better understand the use of computer programming to create multimedia projects with the aim of raising sociopolitical awareness and challenging social injustices. CCL emphasizes that, in addition to cultivating critical consciousness [11] in young people and providing them with content and tools for critiquing, it is important for young people to create interactive media to express their civic opinions or provide solutions [24]. However, as a newly emerged framework, there is much remaining unknown, e.g., how to cultivate CCL in young people and what kind of roles computer programming can play in supporting young people to develop

CCL. To bridge some of the gaps, we surveyed programs (N = 6) that were designed to support youth civic engagement through creating and building projects using computer programming. In this paper, we use the term "program" to refer to projects that engage participants in civic activities and coding, which can include workshops, hackathons, and internships. We identified programs that happened in and outside of academia, then examined these programs using the Cultural-Historical Activity Theory (CHAT) framework [9], namely through the dimensions of subject, object, tool, community, division of labor, and rules. Based on the analysis, we further reflect on how CCL can bring new insights into critical and computational education, as well as the implications for designing civic programs to cultivate CCL in young people. Overall, this work reveals the features and commonalities of the examined CCL programs. The frameworks we employed in the analysis also provide an option for other researchers and educators to design CCL learning experiences and examine CCL development in young people.

CONCEPTUAL FRAMEWORK: CRITICAL COMPUTATIONAL LITERACY

Critical computational literacy (CCL) comes from two traditions of work: critical literacy and computational thinking. On one hand, with critical literacy, students learn to systematically observe, analyze, and deconstruct social injustices and inequalities. Computational thinking, on the other hand, emphasizes the use of concepts and skills from computer science to solve problems, understand the world in new ways, and express ideas [30,52]. The following are more in-depth discussions of both critical literacy and computational thinking.

CRITICAL LITERACY

Growing out of Freire's social justice pedagogy [11], critical literacy emphasizes the importance of analyzing, critiquing, and transforming the norms, rules systems, and practices governing the social fields of everyday life [27]. Critical literacy is essentially the ability that enables people to read and deconstruct the inequitable systems of power in their society, such as identity, access to knowledge, and deconstructing skills [17]. One basic factor of critical literacy is critical consciousness, a reflective awareness of the differences in power, privilege, and inequities [22]. Watts et al. [49] further identify three core components of critical consciousness: critical reflection (reflecting on the culture, policy, and practices of a society), political efficacy ("the perceived ability to effect sociopolitical change"), and critical action (taking actions to initiate social change). Critical literacy enhances and upends traditional educational goals of technical literary skill mastery with a critical skill toolkit of deconstructing power, envisioning alternative futures, and organizing community action [12,27]. For example, primary school children may develop critical literacy by applying their budding literacy skills to reading about the histories of their community, discussing and drafting their own proposals for community improvement, or writing local government officials to petition them to institute alternative changes that they recommend [5,7]. Although theorized in the context of traditional reading and writing, critical literacy also applies to contemporary modes of communication and artifact production, such as computer programming. Through critically examining and constructing ICTs, young people practice (1) identifying dominant ideological values embedded in technologies, (2) reflecting on historical, social, and political phenomena that shaped them, and finally, (3) creating ICTs that reflect their values and communicate their civic ideas to a broader audience [15,47].

> Critical literacy is essentially the ability that enables people to read and deconstruct the inequitable systems of power in their society, such as identity, access to knowledge, and deconstructing skills.

COMPUTATIONAL THINKING

Computational thinking (CT) was rooted in Seymour Papert's book *Mindstorms: Children, Computers, and Powerful Ideas* [32] and popularized by Jeanette M. Wing in the late 2000s as a problem-solving process using concepts and approaches from Computer Science [52]. In the past decade, CS Researchers and educators have significantly enriched the meaning of CT. For example, Brennan and Resnick frame CT along three dimensions: concepts (e.g., sequences and parallelism), practices (e.g., debugging and remixing), and dimensions (e.g., seeing oneself as creator and collaborator) [4]; and Bers argues there are seven important ideas for teaching CT, including algorithms, modularity, control structures, representation, hardware/software, design process, and debugging [2]. Even though how to frame CT is still under debate, some computer scientists have argued that computational thinking represents universally applicable attitudes and would be a fundamental skill used by everyone in the world [30,52]. As people learn to code, they can explore computational concepts and practices as well as cultivate creative thinking and problem-solving abilities [30].

While CT remains an important concept for the growing international push for K-12 CS education, it has also faced significant criticism and competition from more holistic approaches to CS education. Researchers and educators have criticized the way that the concept of computational thinking has remained stagnant despite a changing world and shifting modes of computing, been elevated at the expense of other problem-solving approaches, as well as allowed for individualistic framing of learning [40]. For instance, its focus on individual learners' cognitive processes has not lent well to studying or addressing civic issues within the CS classroom. Recently, some researchers have highlighted the need for CT to go beyond individualist framings to broader social and political relevance (e.g., [18,43]). Adding a critical perspective to CS education is not something new and there is a long history of integrating social issues into computing curricula, known as Computer Science Education for Social Good (CSG-Ed) [13,15]. CSG-Ed specifically targets CS education at school, typically happens at the college level [13], and traditionally "concentrate[s] on computer history, codes of ethics and intellectual property, while neglecting broader issues of societal impact." [14] Situating CS education in critical literacy and social good is still a work-in-progress and continues to be an urgent research need because there is much unknown about what social issues can be included, how they can be combined with computing and presented to learners, and what pedagogical frameworks can structure such learning experience design [14], particularly in the current context where CT is rapidly expanding into K12 classrooms and various informal learning settings.

CRITICAL COMPUTATIONAL LITERACY

CCL synthesizes critical literacy and computational thinking by highlighting the ways in which explicit discussion and exploration of social, political, and economic injustices can support and complement a traditional computer-science-style problem-solving process and vice-versa. Lee and Garcia [23] developed the term "Critical Computational Literacy" in an empirical study where they guided urban youth to create video games based on community social issues in a high-school computer science course. Lee and Garcia defined CCL as "the process in formulating problems and their solutions while utilizing fundamental CS concepts and skills with the explicit purpose of creating sociopolitical awareness and ideological change by examining marginal perspectives and questions commonly accepted beliefs." [24] CCL emphasizes the importance of resisting social injustices and the dominant narratives of institutionally dispossessed communities during the processes of creating, expressing, and inventing with computational skills. Instead of being passive consumers of digital content, CCL equips young people with the computational skills to create critical multimodal and transmedia projects to communicate and improve social issues.

CCL brings new dimensions to both critical literacy and CT. Educators who observed how CT and critical literacy organically bolster one another in learning environments where both are supported, created CCL out of the need for language and framing for describing and better understanding this process [24]. On one hand, CCL provides a framework to explore the unique affordances of new and emerging digital or transmedia tools to develop critical literacy in learners compared with traditional media. On the other hand, CCL specifies the opportunity and need to enrich CT for young people from a critical perspective. As Kafai et al. [18] argued, educators should place students' learning of CT in the tradition of critical pedagogy. As such, students no longer merely learn a technical skill but have the potential to engage with the political, moral, and ethical challenges in their lives and the world. Similarly, Tissenbaum and colleagues [43] propose "computational action" and claim that "young people should have the opportunity to do computing in ways that have a direct impact on their lives and their communities." Even though these scholars have emphasized the importance of bringing a critical perspective to CT, a consistent theoretical foundation to further structure and guide future educational practices within both formal and informal CS learning is lacking. CCL has the potential to be such a theoretical foundation to help CT educators better situate their teaching practices in more authentic, meaningful, and change-initiating contexts.

> **Instead of being passive consumers of digital content, CCL equips young people with the computational skills to create critical multimodal and transmedia projects to communicate and improve social issues.**

However, CCL is still in its infancy and has not yet attracted much attention from researchers or educational practitioners, and how to cultivate CCL in young people needs more understanding and exploration in literature. For example, what are the different ways to support young people's development of CCL? How can we assess young people's development of CCL? How would young people perceive and employ CCL in their daily practices? This study analyzes existing CCL programs and provides insights into addressing some of the gaps.

METHODS

In this section, we introduce the process of finding programs, followed by the theoretical frameworks used to analyze these programs.

IDENTIFYING PROGRAMS

To find youth CCL programs, we searched keywords like "critical computational literacy," "civic engagement," and "children" in the ACM digital library and Web of Science. We also searched these keywords on Google Scholar and the webpages of the youth organizations based in the USA listed on Wikipedia page [25]. Additionally, we examined the references of the included papers. When examining civic engagement elements in a program, we used the indicators of civic engagement from the report of Civic and Political Health of the Nation [26] and Youth Participatory Politics Survey Project [6] as references, which include civic activity, electoral activity, political voice, and online participation. After a wide search, we found six programs for analysis; see Table 1.

Table 1: A Brief Introduction of the Examined Programs

Program	Brief description
True Life Remixed [23]	A seven-week video game program designed by researchers in a high school computer science course for mostly Latino students. The students analyzed sociopolitical issues and created message-driven "choose-your-own-adventure" video games based on one of their community issues using the Scratch programming language.
West Side Stories [24,50]	An app-development program focused on gentrification in West Oakland, California. Young participants, together with adult staff members, investigated various data sources, produced content about the gentrification history, and created a web-based application to present the gentrification of West Oakland.
The Street Arcade [41,42]	A collaborative new media art program where young participants from a low-income community identified and analyzed social issues within their community, then created video games using Scratch programming language based on their chosen issues. The created games were showcased in participants' neighborhoods for community dialogues.
Bay Area Youth [46]	A mobile app program where young participants analyzed socio-political issues in their life, brainstormed project ideas, and built a mobile app that provided young people with information about after-school programs in their neighborhood.
NEED2FEED [33]	A mobile-app-development program where students from a computer science course worked on the design and implementation of an app to utilize social media platforms to increase the number of volunteers at local food pantries.
Slum Innovation Project [39]	This project aims to teach girls from the slums of Dharavi in Mumbai, India, to code and create apps. A group of young women (ages 8–21 years old) built mobile apps using MIT's App Inventor with the support from local mentors to address everyday problems in their community, from sexual harassment, to access to water to lack of education.

ANALYZING PROGRAMS

We employed the Cultural-Historical Activity Theory (CHAT) [9] to structure the analysis. CHAT is a framework for analyzing social interaction systems. Activity systems are conceptualized as the dynamics between six interconnected features: (1) *subject*, the individual or subgroup whose perspective is taken in the analysis; (2) *object*, the raw material or problem space activities are addressing; (3) *tools*, material or cognitive instruments which transform the object into outcomes; (4) *community*, groups who share the same or a similar object; (5) *division of labor*, the structure of power including vertical and horizontal roles and responsibilities; and (6) *rules*, explicit and implicit norms in the activity system [10]. In this case, we use CHAT to understand the activity systems within the included programs.

We followed the approach of content analysis [21] for data analysis. For *subject*, we examined participants' demographic backgrounds and who organized these programs. Regarding *object*, we analyzed the projects that participants created and the civic topics that participants were able to explore. As to *tools*, we looked into (1) the program formats and how long each program lasted, (2) the pedagogical frameworks to structure participants' learning experiences, (3) the civic skills participants were exposed to, as well as (4) the programming tools participants employed to create their projects. Civic skills are defined as the skill sets that are required to effectively participate in civic and political life [19]. When examining civic skills participants were exposed to, we followed the civic skill framework by Kirlin [19], which includes four skill sets: critical thinking, communication, collective decision-making, and organization. For the analysis of *community*, we investigated the sociocultural contexts of the social issues that participants aimed to address. With *division of labor*, we analyzed a program's sessions or activities where participants examined sociopolitical issues and learned the needed computational skills to build interactive projects, as well as the roles of participants and facilitators during these learning experiences. The analysis of participants' and facilitators' roles is based on the framework Guided Participation in civic organizations [20,34], an apprentice-like learning process where less-experienced newcomers engage with those who are more expert in community activities. The three patterns of guided participation include facilitation (youth-centered, adults to be neutral facilitators of the youth-led process), apprenticeship (youth-centered, adults tend to be more involved by providing coaching, feedback, and decision making), and joint work (adults collaborate with youth, but the working environment is not youth-centered). Finally, the implicit and explicit space norms of these programs are difficult to ascertain from available sources. Therefore, we did not perform an analysis for the dimension of rules.

FINDINGS

Figure 1 is a summary of the examined programs. Some programs provided less information about their processes. Therefore, such programs were not included in the analysis in terms of the aspects where the information was not accessible. Following the CHAT framework, this section presents the analyzing results of the six programs.

SUBJECTS

Five programs recruited youth of color from communities underrepresented in CS. For example, most participants of West Side Stories and Plug-In Studio Street Arcade Game were youth of color from low-income communities; True Life Remixed recruited mainly Latino youth; Bay Area Youth recruited Black participants. The Slum Innovation Project recruited young women living in the slum community of Dharavi, in Mumbai, India. These programs were more uniform in terms of organizers, mainly by researchers in academia in partnership with a civic organization or outside funding support.

OBJECTS

Participants mainly produced two types of projects: video games and web/mobile-based applications. For example, the

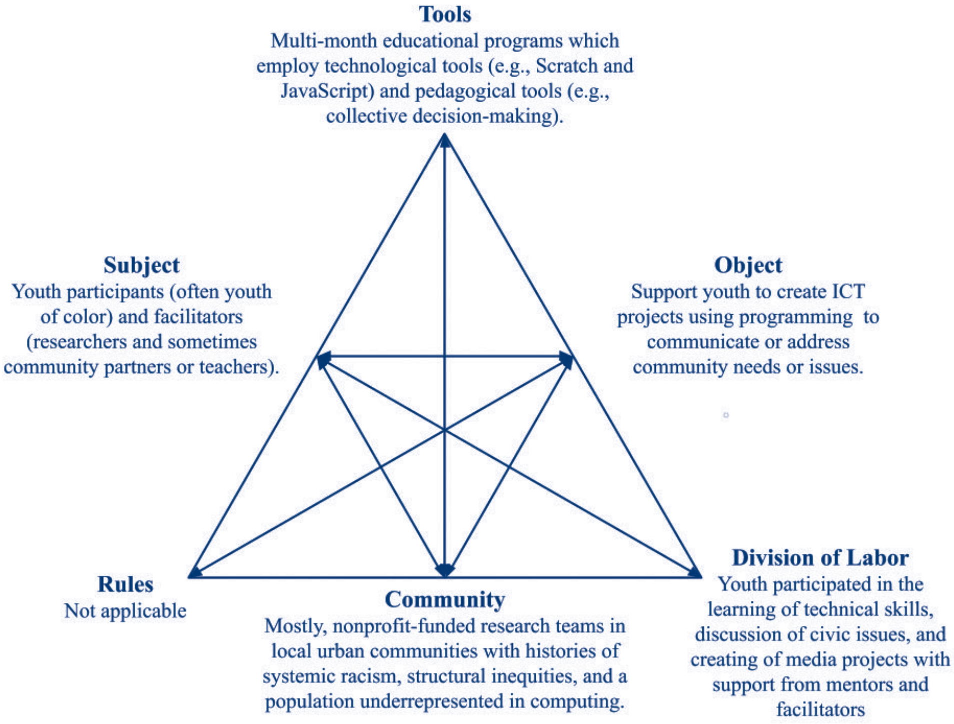
Tools
Multi-month educational programs which employ technological tools (e.g., Scratch and JavaScript) and pedagogical tools (e.g., collective decision-making).
Subject
Youth participants (often youth of color) and facilitators (researchers and sometimes community partners or teachers).
Object
Support youth to create ICT projects using programming to communicate or address community needs or issues.
Rules
Not applicable
Community
Mostly, nonprofit-funded research teams in local urban communities with histories of systemic racism, structural inequities, and a population underrepresented in computing.
Division of Labor
Youth participated in the learning of technical skills, discussion of civic issues, and creating of media projects with support from mentors and facilitators

Figure 1: The summary of the examined CCL programs following the CHAT framework.

Street Arcade Games and True Life Remixed asked participants to create video games based on one of the social issues within their community. The Slum Innovation Project, NEED2FEED, and Bay Area Youth all supported young people to develop mobile apps to address issues they contend with in their communities and can be used by fellow community members. In terms of civic topics, most programs focused on community challenges within participants' neighborhoods. Some of these challenges include white privilege, racial profiling, police brutality, peer pressure, the inadequacy of police in high-crime and materially unprivileged communities, marginalized perspectives, and women's safety.

TOOLS

Program Formats. Five out of the six programs happened outside schools as after-school programs. West Side Stories, NEED2FEED, and Bay Area Youth gathered young participants to discuss civic issues and develop interactive projects either in the evenings or weekends throughout a normal-course semester. The Slum Innovation Project was run in a community-based learning center. The Street Arcade Games happened during summer vacation. Only True Life Mixed took advantage of a course in a high school to engage students in creating civic-centered video games. Also, all programs took several months to finish.

Pedagogical Frameworks. To scaffold participants' learning, some pedagogical frameworks were employed in some programs. Two programs specifically say they employed CCL, including True Life Remixed and West Side Stories. Both programs guided students to identify and analyze social issues, then create media projects using computational skills. West Side Stories also adopted the collegial pedagogy where "young people and adults working as colleagues together produce original median and technology projects." [20] The program Bay Area Youth phrased its pedagogy as "critical pedagogy in the context of a mobile app programming project" [38], which is exactly CCL because the program guided participants to design and create a mobile app based on a community issue that participants cared about. Street Arcade, Slum Innovation Project, and NEED2FEED did not disclose their pedagogical frameworks.

Civic Skills. First, all programs exposed participants to certain critical thinking skills, especially identifying and analyzing public issues. For example, when identifying and analyzing social issues, participants of Street Arcade Games, the Slum Innovation Project, and True Life Remixed brainstormed and compiled the social issues that they thought were not well addressed or understood within their communities. Participants of West Side Stories explored the gentrification stories and marginalized perspectives by audio interviewing residents and investigating a range of resources, such as library archives, oral histories, and public databases. In addition, some participants examined and valued marginalized perspectives in their neighborhood. Second, participants also employed communication skills in the examined programs. Several programs exposed participants to civic writing. For example, participants in West Side Stories collected the gentrification stories of West Oakland and wrote these stories which were presented in their final application. Additionally, four programs supported participants to circulate their civic thoughts to a broad audience by publishing their created projects online, e.g., the video games created through Street Arcade Games and

True Life Remixed using Scratch programming environment are publicly accessible on the Scratch website. Participants of Street Arcade Games also showcased their created video games in their community and actively talked with community members about and guide them to reflect on the community issues behind these games. Third, exposure to collective decision-making skills generally happened in the process of deciding project topics and formats. For example, the Slum Innovation Project allowed participants to decide the topic and design of their apps based on the issues relevant to them in their communities. Although the project topic was decided by organizers in West Side Stories, participants actively participated in the discussion and decision-making process of who was their target audience, how these audiences received messages, and the format of the final project, where they shared their opinions and listened to others' opinions. Finally, we did not notice participants being involved in organizational skills in the examined programs.

Programming Skills. Five programs recruited participants without programming experiences. Four programs adopted block-based programming languages that are mainly designed for programming beginners and children. For example, Street Arcade Games and True Life Remixed adopted Scratch as their programming tool to create video games.

Programming Skills. Five programs recruited participants without programming experiences. Four programs adopted block-based programming languages that are mainly designed for programming beginners and children. For example, Street Arcade Games and True Life Remixed adopted Scratch as their programming tool to create video games. Two programs (NEED2FEED and West Side Stories) employed professional programming languages and environments in their projects like C#, Java, and JavaScript.

COMMUNITY

Four projects (Street Arcade, West Side Stories, Bay Area Youth, and True Life Remixed) were designed for low-income youth or youth of color from urban, historically Black, or Latino underserved communities in the US to design and create ICTs that engage with issues in their immediate contexts. Central to these neighborhoods' histories and sociopolitical context are systemic injustices around anti-Black racism, police violence, and gentrification, among others. The Slum Innovation Project was situated in the context of one of the world's largest slums, Dharavi, in Mumbai, India. This project was an initiative from Dharavi Diary, a non-profit launched by an Indian filmmaker to support the education and empowerment of young Indians, especially low-income rural or urban women. NEED2FEED involved undergraduate and graduate students to work with food pantry coordinators across the county where these young participants were located.

DIVISION OF LABOR

Youth participants in all six projects were involved in the design and creation process of their respective apps or games to various degrees, with help from facilitators. In the following, we provided details into the moments where participants learned civic and computational skills, as well as the roles of participants and facilitators in these learning moments.

Learning Civic Skills. Most programs met weekly so that participants and facilitators could gather to discuss social issues and work on their projects. Three programs (Street Arcade Games, West Side Stories, Bay Area Youth) held brainstorming sessions for participants to identify, reflect on, and discuss social issues. For example, in Bay Area Youth, group conversations were facilitated by organizers for participants to understand the social issues in their neighborhood, which were structured by activities, informal conversations around the role of technology, and open-ended discussions about the root causes of inequity and oppression. In addition to brainstorm and discussion sessions, two programs (West Sides Stories and Bay Area Youth) guided participants to investigate resources to get a better understanding of certain sociopolitical issues. For example, participants of West Sides Stories investigated library archives, oral histories, and public databases to discover the gentrification stories of West Oakland. They also interviewed residents in their neighborhood to gather more insights into the community change.

Learning Computer Programming. Five programs recruited participants who had little or no programming experience. These programs, therefore, specifically included sessions to teach participants the needed programming skills for their projects. For example, the Street Arcade Games and Bay Area Youth held programming workshops to teach participants the needed coding skills, and participants of West Sides Stories took part in a series of after-school coding classes held by organizers. Additionally, the Street Arcade Games guided participants to interact with video games created by other Scratch users and remix these projects to build their own video games.

Learning Roles. We examined the roles of participants and facilitators in the above-mentioned learning experiences using the Guide Participation (i.e., facilitation, apprenticeship, and joint work) [20,34]. We notice the facilitation approach in the qualitative narratives of two programs (Street Arcade and Bay Area Youth), mainly in the brainstorming and discussion of social issues and generating project ideas where participants were guided to brainstorm and compile social issues and generate possible themes for their projects where participants played a leading role. In the learning of computational skills and developing projects, we notice the apprenticeship approach where facilitators were more actively involved in coaching and giving feedback. For example, when participants started to code video games in Street Arcade Games, facilitators worked individually with each teen to overcome coding challenges, such as building a scoring system and to make sure the chosen social issues were sufficiently represented in his/her game. Finally, we noticed the approach of joint work in West Side Stories, where facilitators worked as co-producers of participants' final application.

DISCUSSION

The CCL programs examined in this paper mostly took the form of after-school programs. Young people participated in months-long programs where they learned the needed programming skills, analyzed and reflected on the social issues within their community, and built interactive media projects to communicate or address identified issues mainly through video games and mobile applications. These programs highlight the existing practices to develop CCL in young people. More specifically, the critical aspect of CCL was supported by guiding young people to investigate, analyze, understand, and explain civic issues, such as the brainstorm sessions in Street Arcade Games and audio-interview with residents by participants in West Side Stories. Some programs, like NEED2FEED and the Slum Innovation Project took a step further by supporting young people to create working applications that could be downloaded by and benefit real users. Regarding the computational aspect of CCL, most programs focused on helping participants learn the necessary coding skills for their projects.

The CCL programs examined in this paper mostly took the form of after-school programs. Young people participated in months-long programs where they learned the needed programming skills, analyzed and reflected on the social issues within their community, and built interactive media projects to communicate or address identified issues mainly through video games and mobile applications.

These CCL programs are a complement to existing CSG-Ed practices. First, curricula based on CSG-Ed are usually designed for CS students who have already taken some introductory programming and algorithm courses, which might not be helpful in shaping students' perceptions about the social values and impacts of computing [13]. Therefore, there has been a call recently for incorporating CSG-Ed activities into the introductory CS curriculum [13,16]. Different from CSG-Ed practices, the CCL programs in this paper expanded computing for social good to young people who had no prior programming experiences, introduced computing to them in familiar and authentic social contexts, and encouraged them to voice their civic ideas, even develop real solutions. Such learning experiences that are associated with meaningful topics for young people can help them demystify CS and see the social impact they can bring through computing, finally motivating their interest in pursuing CS and broadening participation in computing [13]. Second, CCL supports designing learning experiences in both formal and informal learning settings, which are different from CSG-Ed endeavors that are typically classroom-based. As industry leaders, policymakers, and educators have recognized the importance of CT [30], many CS learning activities for children have been developed for community-based organizations like libraries, museums, and maker spaces (e.g., [29,35]). Situating these activities in CSG-Ed is inappropriate due to the differences between formal and informal learning design, such as a lack of traditional curricula, professional CS instructors, and much shorter learning time in informal settings. However, CCL can serve as a lens to enrich such informal CS learning activities from a social-good perspective.

Finally, CCL practices can go beyond these existing programs. First, there is still much to be understood about the process of learners applying the two quite different CT and critical literacy approaches in parallel in the same project. CCL does not just explore how critical literacy can be extended to new media; CCL provides a framework for examining how critical literacy and CT complement one another and can be synthesized. Lee and Soep [24] describe in-depth two moments during the production of the West Side Stories application in which participants had to negotiate and combine considerations pulled from both CT and critical literacy. They theorize, through the language of CCL, that supporting both CT and critical literacy is essential so that "learners develop computational skills for disrupting and improving the inequitable conditions in society." [24] However, further examination of CCL is needed, specifically how CT and critical literacy intersect, how they can contribute to one other, how they can be synthesized, and how to design learning environments to support all these processes. This study contributes to the emerging line of inquiry around CCL. Second, CT in these CCL programs was mostly treated—and undervalued—simply as a tool to build projects. CCL is not just about building digital projects using computational skills, but more about developing digital empowerment (i.e., the recognition in oneself that they have the ability to create tools and artifacts with computers to enhance their and other people's lives) and computational identity (i.e., the identity one forms as a person who can think computationally in a broader computational community) [44]. In other words, it is essential to build young people's confidence in leveraging computer programming in authentic sociopolitical contexts when designing CCL learning experiences. As such, young people may be able to feel and experience the real power of computing and themselves to challenge injustice and initiate social change.

IMPLICATIONS FOR DESIGNING CCL PROGRAMS

We further discuss suggestions for civic researchers, organizations, and computing educators to design programs that support youth civic engagement through computing. The suggestions are mainly focused on program features that help youth develop CCL for disrupting and improving inequitable social conditions.

Choosing Project Topics as a Civic Practice. A common approach to deciding civic topics in the examined programs was to ask participants to reflect on the sociopolitical issues within their community, then decided on the topics they were interested in improving or addressing. We see possibilities to expose participants to sociopolitical issues in different ways, such as fieldwork and participating in relevant civic activities. When designing for issues that participants are not so familiar with, a program can organize onsite fieldwork for participants to observe and experience the issues closely or invite people who are facing the issues to be part of the program. These problem-identifying practices can help raise young people's awareness around inequalities and injustices, the first step and a key component of CCL. In addition to the issues within participants' community, a future program can be focused on solutions to a wide range of social challenges, such as climate change and the current Black Lives Matter movement.

Expanding What Participants Can Create. Participants in the examined programs created either video games to communicate community issues or mobile/web-based applications to help address social challenges. The CCL framework does not set a limitation on what kinds of projects participants can make. On one hand, video games can be applied to more programs because video games are a highly interactive and inviting format for young people and can motivate them to learn and critically reflect on the represented sociopolitical issues. On the other hand, participants can build interactive animations, arts, and music projects based on sociopolitical issues to engage more people. For example, in the 2011 Chilean higher education student movement [37], the song Shock was created and widely circulated among young protesters, which communicated the students' demand for changing Chile's market-oriented educational system. Additionally, building physically interactive projects is another meaningful direction. Multiple tools are relatively easy for people who have little programming experience to build physical computing projects, such as Arduino and Micro:bit. Such physical computing tools can support young people to work on solutions that can address social issues beyond the screen.

Combining critical and computational literacy is a promising tool to cultivate CCL in young people and help them become active creators of ICTs rather than passive consumers. ... Future research can continue to design and implement CCL programs in various forms and settings as well as explore effective ways to engage young people in CCL.

Expanding Who Can Participate. Five of the examined programs recruited youth of color from low-income communities, most of whom are underrepresented in computing and may have limited access to quality education opportunities, especially STEM education. More programs under CCL framework should be designed for them. In addition, future programs can consider people of different genders, races, social classes, and levels of expertise in programming. For example, students in computing have educational experiences that typically focus on technical issues. However, they might not pay enough attention to how their skills can be used to address sociopolitical issues, or even do it in a wrong way because of a lack of an appropriate framework to help guide the practice [45]. CCL provides an opportunity for those with a computing background to bring their attention to sociopolitical issues and the civic contributions they can make. Finally, a program can also consider intergenerational or family participation in CCL. For example, a program can recruit both young people and their parents to work as a group, which can not only broaden the audiences of critical computational engagement but also help involve parents in their children's learning process.

Learning Various Civic Skills. In addition to critical thinking, the civic skill supported by all the examined programs, a program can consider supporting young people to develop other important civic skills, including communication, collective decision making, and organization skills. As to communication skills, a program can guide participants to communicate certain social issues, civic ideas, and projects to a broader audience, such as posting them on social media and presenting them in public. For example, participants can do presentations of their projects and their chosen issues in front of their peer participants, facilitators, even community members in their neighborhood, or share their projects with more people by recording videos and publishing them online. It is also important to train participants in the language skills and vocabulary necessary to communicate social issues, particularly for youth from underrepresented groups, which would help them use the most effective narratives to tell their stories or the issues they care about. As to the skills of collective decision-making and organization, a program can involve participants more in the program's planning and administrative work. For instance, young people can participate in planning the program schedule, deciding project topics and formats, as well as organizing group activities like running group meetings.

LIMITATIONS

First, we did not talk to participants, facilitators, and organizers of the examined programs. Therefore, we might have missed some important details and participants' learning experiences, especially about the roles of facilitators and teachers and how they balanced the power structure in these programs. Second, we were unable to find many computing-based civic programs because CCL is an emergent field of study. However, we believe the findings based on the examined programs are helpful for understanding how computing can be leveraged in civic engagement programs as well as for designing new CCL programs.

CONCLUSION

Combining critical and computational literacy is a promising tool to cultivate CCL in young people and help them become active creators of ICTs rather than passive consumers. We reviewed computing-based civic programs and examined the features of these programs, as well as how young people were engaged in civic activities through building projects using computer programming. Through our analyses, we see more possibilities to support CCL through computing-based civic programs, such as expanding the civic topics, who can participate, and what young people can create. Ultimately, this work provides insights into computing-based civic programs and highlights ways for civic engagement researchers, organizations, and computing educators to design new programs to support the cultivation of CCL in young people. Future research can continue to design and implement CCL programs in various forms and settings as well as explore effective ways to engage young people in CCL. ❖

References

1. Arthur, C. What Is the 1% Rule? *The Guardian*, July 19, 2006; https://www.theguardian.com/technology/2006/jul/20/guardianweeklytechnologysection2. Accessed 2019 January 8.
2. Bers, M.U. *Coding as a Playground Programming and Computational Thinking in the Early Childhood Classroom*. (NY, Routeledge, 2017).
3. Boulianne, S. Twenty Years of Digital Media Effects on Civic and Political Participation. *Communication Research*, 47, 7 (2020), 947–966.
4. Brennan, K. and Resnick, M. New frameworks for studying and assessing the development of computational thinking. *Proceedings of the 2012 annual meeting of the American Educational Research Association* (Vancouver, BC, Canada, 2012), 1–25.
5. Campano, G., Ghiso, M.P. and Sanchez, L. "Nobody Knows the… Amount of a Person": Elementary Students Critiquing Dehumanization through Organic Critical Literacies. *Research in the Teaching of English*, 48, 1 (2013), 98–125.
6. Cohen, C.J., Kahne, J., Bowyer, B., Middaugh, E. and Rogowski, J. *Paricipatory Politics: New Media and Youth Political Action*. (YPP Research Network, 2012).
7. Comber, B., Thomson, P. and Wells, M. Critical Literacy Finds a "Place": Writing and Social Action in a Low-Income Australian Grade 2/3 Classroom. *The Elementary School Journal*, 101, 4 (2001), 451–464.
8. Ekström, M. and Shehata, A. Social media, porous boundaries, and the development of online political engagement among young citizens. *New Media & Society*, 20, 2 (2018), 740–759.
9. Engeström, Y. *Learning by expanding*. (Cambridge University Press, 2015).
10. Engeström, Y. and Sannino, A. Studies of expansive learning: Foundations, findings and future challenges. *Educational Research Review*, 5, 1 (2010), 1–24.
11. Freire, P. *Pedagogy of the Oppressed*. (New York, Herder and Herder, 1972).
12. Freire, P. and Macedo, D. Literacy: *Reading the word and the world*. (New York, Routeledge, 2005).
13. Goldweber, M., Barr, J., Clear, T., Davoli, R., Mann, S., Patitsas, E. and Portnoff, S. A framework for enhancing the social good in computing education: A values approach. *ACM Inroads*, 4, 1 (2013), 58–79.
14. Goldweber, M., Davoli, R., Little, J.C., Riedesel, C., Walker, H., Little, J.C., Riedesel, C., Walker, H., Cross, G. and Von Konsky, B.R. Enhancing the social issues components in our computing curriculum: computing for the social good. *ACM Inroad*, 2, 1 (2011), 64-82.
15. Goldweber, M., Kaczmarczyk, L. and Blumenthal, R. Computing for the social good in education. *ACM Inroads*, 10, 4 (2019), 24–29.
16. Goldweber, M., Kaczmarczyk, L., Blumenthal, R. and Clear, A. A Hands-On Tutorial on How To Incorporate Computing for Social Good in the Introductory Course Sequence. *Proceedings of the 51st ACM Technical Symposium on Computer Science Education* (2020), 954–955.
17. Janks, H. Critical literacy in teaching and research. *Education Inquiry*, 4, 2 (2013), 225–242.
18. Kafai, Y., Proctor, C. and Lui, D. From Theory Bias to Theory Dialogue: Embracing Cognitive, Situated, and Critical Framings of Computational Thinking in K-12 CS Education. *ICER '19 Proceedings of the 2019 ACM Conference on International Computing Education Research* (Toronto ON, Canada, 2019), 101–109.
19. Kirlin, M. The role of civic skills in fostering civic engagement. *Center for Information and Research on Civic Learning and Engagement (CIRCLE)*, June (2003), 512.
20. Kirshner, B. Guided participation in three youth activism organizations: Facilitation, apprenticeship, and joint work. *The Journal of the Learning Sciences*, 17, 1 (2008), 60–101.
21. Kondracki, N.L., Wellman, N.S. and Amundson, D.R. Content analysis: Review of methods and their applications in nutrition education. *Journal of Nutrition Education and Behavior*, 34, 4 (2002), 224–230.
22. Kumagai, A. and Lypson, M. Beyond cultural competence: critical consciousness, social justice, and multicultural education. *Academic Medicine*, 84, 6 (2009), 782–787.
23. Lee, C.H. and Garcia, A.D. I want them to feel the fear: Leveraging critical computational literacies for English Language Arts success. *Exploring Multimodal Composition and Digital Writing*, (2014) 364–378.
24. Lee, C.H. and Soep, E. None But Ourselves Can Free Our Minds: Critical Computational Literacy as a Pedagogy of Resistance. *Equity and Excellence in Education*, 49, 4 (2016), 480–492.
25. List of youth organizations; https://en.wikipedia.org/wiki/List_of_youth_organizations. Accessed 2018 December 07.
26. Lopez, M.H., Levine, P., Both, D., Kiesa, A., Kirby, E., Marcelo, K. and Williams, D. The 2006 civic and political health of the nation; https://www.pewtrusts.org/en/research-and-analysis/reports/2006/10/03/the-2006-civic-and-political-health-of-the-nation-survey. Accessed 2020 Oct 30.
27. Luke, A. Critical Literacy: Foundational Notes. *Theory Into Practice*, 51, 1 (2012), 4–11.
28. Manzuoli, C.H., Sánchez, A.V. and Bedoya, E.D. Digital Citizenship: A Theoretical Review of the Concept and Trends. *Turkish Online Journal of Educational Technology-TOJET*, 18, 2 (2019), 10–18.
29. Metcalf, L.E. and Anderson, J.L. Hidden No More: Using Museum Exhibits of Underrepresented Scientists to Engage Students in Computational Thinking. *Proceedings of Society for Information Technology & Teacher Education International Conference 2020* (Online, Apr. 2020), 1416–1421.
30. National Research Counci. *Report of a workshop on the pedagogical aspects of computational thinking*. (National Academies Press, 2011).
31. Ohlin, J., Heller, A., Byrne, S. and Keevy, N. How young people participate in civic activities using Internet and mobile technologies. *Report to the National Youth Affairs Research Scheme (NYARS)*. (2010).
32. Papert, S. *Mindstorms: Children, Computers, and Powerful Ideas*. (Basic Books Inc., 1980).
33. Pichardo, M., Samuels, G., O'Leary, B., Bayiokos, P., Kapiti, A., Asija, S. and Coppola, J.F. Service-learning project for computing students: Creating a mobile app for a non-profit agency: Helping community based organizations manage events & volunteers. *2016 IEEE Long Island Systems, Applications and Technology Conference*. (Farmingdale, NY, USA, 2016).
34. Rogoff, B., Mistry, J., Göncü, A., Mosier, C. and Mosier, C. Guided Participation in Cultural Activity by Toddlers and Caregivers. *Monographs of the Society for Research in Child Development*, 58, 8 (1993), 184.
35. Roque, R. Family Creative Learning. In *Makeology in K-12, Higher, and Informal Education*. (New York, Routeledge, 2016).
36. Roque, R., Dasgupta, S. and Costanza-Chock, S. Children's Civic Engagement in the Scratch Online Community. *Social Sciences*, 5, 4 (2016), 55.
37. Salinas, D. and Fraser, P. Educational Opportunity and Contentious Politics: The 2011 Chilean Student Movement. *Berkeley Review of Education*. 3, 1 (2012), 17–47.
38. Skoric, M.M. What is Slack about Slactivism? *Methodological and Conceptual Issues in Cyber Activism Research*. (2012), 77–104.

39. Slum Innovation Project: *https://mashable.com/2016/03/29/mumbai-dharavi-girls-coding-apps/;https://qz.com/india/1032018/dharavi-diary-what-happens-when-girls-in-one-of-the-worlds-largest-slums-start-coding-and-building-apps/*. Accessed 2020 August 12.

40. Tedre, M. and Denning, P.J. The long quest for computational thinking. *Proceedings of the 16th Koli Calling International Conference on Computing Education Research* (2016), 120–129.

41. The Plug-In Studio Street Arcade: *https://medium.com/scratchteam-blog/the-plug-in-studio-street-arcade-92e5d0206463*. Accessed 2018 December 06.

42. The Street Arcade Games: *http://pluginstudio.net/games.html*. Accessed 2018 December 06.

43. Tissenbaum, M., Sheldon, J. and Abelson, H. From computational thinking to computational action. *Communications of the ACM*, 62, 3 (2019), 34–36.

44. Tissenbaum, M., Sheldon, J., Seop, L., Lee, C.H. and Lao, N. Critical computational empowerment: Engaging youth as shapers of the digital future. *IEEE Global Engineering Education Conference, EDUCON* (2017), 1705–1708.

45. To Really "Disrupt," Tech Needs to Listen to Actual Researchers. WIRED, July 26, 2019; *https://www.wired.com/story/tech-needs-to-listen-to-actual-researchers/*. Accessed 2019 August 07.

46. Vakil, S. A Critical Pedagogy Approach for Engaging Urban Youth in Mobile App Development in an After-School Program. *Equity and Excellence in Education*, 47, 1 (2014), 31–45.

47. Vakil, S. Ethics, identity, and political vision: Toward a justice-centered approach to equity in computer science education. *Harvard Educational Review*, 88, 1 (2018), 26–52.

48. Vogel, S., Santo, R. and Ching, D. Visions of computer science education: Unpacking arguments for and projected impacts of CS4All initiatives. *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education* (Seattle, Washington, USA, 2017), 609–614.

49. Watts, R.J., Diemer, M.A. and Voight, A.M. Critical Consciousness: Current Status and Future Directions. *Youth Civic Development: Work at the Cutting Edge*. (2011), 43–57.

50. West Side Stories: *https://youthradio.github.io/*. Accessed 2018 December 12.

51. White, M. Clicktivism Is Ruining Leftist Activism. *The Guardian*, August 12, 2010; https://www.theguardian.com/commentisfree/2010/aug/12/clicktivism-ruining-leftist-activism. Accessed 2019, January 08.

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

Junnan Yu[1]
Department of Information Science
AUniversity of Colorado Boulder
1045 18th Street, UCB 315
Boulder, CO 80309, USA
junnan.yu@colorado.edu

Janet Ruppert[2]
Department of Information Science
University of Colorado Boulder
1045 18th Street, UCB 315
Boulder, CO 80309, USA
janet.ruppert@colorado.edu

Ricarose Roque
Department of Information Science
University of Colorado Boulder
1045 18th Street, UCB 315
Boulder, CO 80309, USA
ricarose@colorado.edu

Ben Kirshner
School of Education, Room 312
University of Colorado Boulder
249 UCB, Boulder, CO 80309, USA
Ben.Kirshner@colorado.edu

[1] Co-first author; Leader. [2] Co-first author.

DOI: 10.1145/3432727

©2020 ACM 2153-2184/20/12 $15.00

INTERACTIONS

An influential voice in the study of people, technology, and design.

EVERY ISSUE:

- Explores how and why we interact with the designed world of technologies
- Offers content to inspire and educate HCI designers
- Shares innovations and creations in the business world
- Makes engaging HCI research accessible to practitioners and makes practitioners voices heard by researchers.

To learn more about us, visit our award-winning website
http://interactions.acm.org

Follow us on
Facebook and Twitter

To subscribe:
http://www.acm.org/subscribe

Association for Computing Machinery