Critical Computational Empowerment: Engaging Youth as Shapers of the Digital Future

Mike Tissenbaum  
Massachusetts Institute of Technology  
Cambridge, MA  
mtissen@mit.edu

Josh Sheldon  
Massachusetts Institute of Technology  
Cambridge, MA  
jsheldon@mit.edu

Lissa Seop  
Youth Radio  
Oakland, CA  
lissa@youthradio.org

Clifford H. Lee  
St. Mary’s College of California  
Moraga, CA  
chl3@stmarys-ca.edu

Natalie Lao  
Massachusetts Institute of Technology  
Cambridge, MA  
manting@mit.edu

Abstract—While there is growing recognition of the need to support young learners as they develop computational thinking (CT) skills, this paper advocates for an increased focus on computational identity (CI) and digital empowerment (DE) in particular, as we posit that these CT skills will help young learners become self-motivated, innovative creators. When nurturing identity and empowerment, it is a persistent challenge to provide youth with real-world experiences and suitable development tools. This paper advances two key approaches to address these issues: 1) A critical computational literacy approach to engage youth in developing personally meaningful applications that have impact in the real world; 2) the use of a tool, like MIT App Inventor, that lowers barriers to creating useful, impactful technology. Using two case-based studies, we show how these two approaches have supported youth in developing applications that respond to meaningful challenges in their communities and helped them to establish their computational identities and digital empowerment.

Keywords—computational thinking; digital empowerment; computational identity

I. MOVING FROM DIGITAL LITERACY TO DIGITAL EMPOWERMENT

While there has been general agreement that we need to rethink the social and technical skills today’s youth will require to become productive members of a society that is increasingly mediated by digital technologies [9][4], there has been little consensus on what these skills are or how to support their development. Many educational researchers have advocated for teaching "digital literacies." In this case, by literacy we mean the ability to read and write in the shared language of a culture. Thereby, we define digital literacy as the ability to share ideas through digital mediums. For example, digitally literate people might, upload posts and videos to social networking sites, or contribute to shared knowledge through the use of collaborative tools such as wikis [6][13][11].

A challenge with much of the work around supporting students’ digital literacy is that it is often detached from the context of engagement and empowerment for the participants. I.e., blogs and wikis may allow participants to discuss an event, but lack immediacy. They are often viewed or read on a screen far removed, in one or more dimensions, from the topic itself. Further participation is often fragmented and bound by the limitations of using off-the-shelf technologies (e.g., Facebook, YouTube).

Another challenge with this view of "digital literacy" is that it largely separates the computational aspects from the social aspects. That is, students develop a critical understanding of the unique affordances and capabilities of computation divorced from any active participation in their own learning. Supporting learners in developing deeper understandings of how to think "computationally" offers them opportunities to not only create digital products, but more critically, to formulate problems and their solutions so that the solutions can be solved computationally [1][2][5]. Brennan and Resnick describe three core aspects of computational thinking: concepts, the key constructs and ideas that are central to most forms of computing; practices, the activities people engage in when creating computational projects; and perspectives, the ways in which individuals see themselves as computational thinkers [5].

For us, the development of computational thinking perspectives is particularly critical in supporting individual growth. Our recent work centers on two key elements within CT perspectives: digital empowerment (DE), which describes the recognition of their capacity to have active, positive roles creating tools and artifacts with computers to enhance their lives and the lives of people around them; and computational identity, the formation of their identities as people who can think computationally and their membership in the computational community more broadly. This definition of computational identity draws from the corresponding concept in science education, namely scientific identity (i.e., learners see themselves as scientists and feel that they are engaging in the authentic practices of scientists [15]).

In 2006, Scardamalia and Bereiter argued the need to develop technologies that allowed students to engage in the authentic practices of a knowledge community, rather than subvert the practices to fit existing off-the-shelf tools. Similarly, if our aim is to support youth in developing their digital empowerment, we need to provide them with programming environments that offer the creative freedom to design and develop computational approaches for interacting with the world around them. Expressing this kind of creative freedom has traditionally required individuals to learn a text-based programming language (e.g., Java, or Javascript). These languages create significant barriers for novices and non-programmers to thinking computationally and building digital products. In the past, to start building computational artifacts,
novice learners would have to learn opaque, sometimes arcane, programming languages in tandem with developing critical thinking skills, i.e. computational thinking. Now, while there is little agreement on the precise definition of "computational thinking" [1], there is general consensus that it is distinct from pure "coding." As a result, many are advocating for platforms that reduce the barriers for learners to design, build, and implement their computational ideas. Such platforms would reduce the cognitive load of coding, allowing learners to invest more effort in becoming computational thinkers.

We believe that true empowerment comes from creating interventions that move into the physical world - interventions that allow learners to analyze problems in their communities and devise real solutions to these problems. As such, we have been working with a platform called MIT App Inventor that allows learners to quickly design, build, and share mobile applications. MIT App Inventor is a blocks-based programming language that allows users to quickly build, refine, and share fully functional mobile applications without the need to understand or wrestle with complicated syntax (See Fig. 1 and Fig. 2).

II. USING CRITICAL COMPUTATIONAL LITERACY TO DRIVE DIGITAL EMPOWERMENT

To support youth as they develop their computational identities and become more digitally empowered, there is a need to make meaningful connections between their designs and the world around them - their designs should allow them to have real impact on issues that affect their daily lives. In response, a team from Youth Radio, a non-profit media organization that supports journalism by young people, has been using MIT App Inventor to develop a new pedagogical...
and conceptual framework they call critical computational literacy [8]. Critical computational literacy (CCL) combines computational thinking and critical pedagogy. The view of computational thinking that informs CCL accounts for complex analytical and interpretive practices that go well beyond the mechanics of learning to code [2][5]. Indebted to Paolo Freire [3], as well as more contemporary work in literacy studies, the notion of critical pedagogy that frames CCL centers marginalized points of view and creates conditions for young people to break silences, reveal obscured truths, and challenge unjust systems and conditions. Young makers carry out this important work through collaborative digital authorship. Together with peers and colleagues, they envision, design, and develop a range of digital projects and release their work to real audiences. These products include downloadable apps, web-based interactives, and transmedia stories through which young people acquire technical skills, participate in civic life, and engage their communities as co-creators.

Taken together, digital empowerment and computational identity are enhanced by the development of critical computational literacy. By grounding learning and activity in a relevant, real-world frame, youth who are becoming critically, computationally literate can also develop their computational identities towards becoming digitally empowered.

III. EXAMPLES OF CRITICAL COMPUTATIONAL EMPOWERMENT

A. Example 1: Engaging Youth Around Mental Health

Critical computational literacy at Youth Radio can emerge at every stage in the creation of a digital product, from ideation to research, design, development, testing, publication, and community engagement sparked by the finished work. In framing one mobile app created using MIT App Inventor, Mood Ring, a group of teens knew they wanted to build a tool that promoted youth health. They pinpointed mental health as a particular interest. After rounds and rounds of brainstorming, user research, and wireframing, they landed on the specific goal of their app: to enable users to track how they’re feeling using any combination of nine emojis so that they could better understand their own mental health and receive appropriate emotional support from trusted friends (Fig. 3).

While the app would have been a lot easier to build if users could only pick one emoji, the designers knew that a marker of their complex young-adult lives was the simultaneous experience of multiple, sometimes conflicting emotions. The development of Mood Ring needed to be careful not to pathologize sadness or narrow young people’s understandings of their own emotional states. So, the coding task became harder—they needed to build functionality that allowed users to select several emojis at one time. These youths overcame that technical challenge in order to create a project that accomplished the ambitious social goals they sought.

This is an example of how young people came to see themselves as competent developers capable of taking on daunting computational tasks, cultivating empowerment not only in themselves but also among their target users.

B. Example 2: Providing Relief to Disaster Stricken Areas

Developing solutions for an underrepresented or marginalized group can also spring from responses to unexpected real-time events. Ananya, a 10-year-old from India, was inspired by discussions in her classroom and decided that she wanted to find a solution to help flood victims in the Tamil Nadu region of India, and in the city of Chennai in particular. Many residents of Chennai had lost their homes and belongings, and relief efforts were struggling to coordinate resources and to reach those who needed help the most. Many of those affected were poor and lacked the means to seek help on their own. In response, Ananya developed CalamityRelief, an app that helped coordinate the dropping off and picking up of relief supplies (Fig. 4). Users of her app can see a list of collection centers to find the nearest one, as well as guidelines for what they should or should not donate [14]. The app also helps coordinate distributors and social workers, informing them of what materials are at which donation centers, so they know where and when to pick up relief supplies. While developing CalamityRelief, Ananya needed to carefully consider what the needs of the community were and how her contribution could reduce many of the challenges facing the donation process (i.e., what is really needed, where, and when).

Using a critical computational literacy approach, we see that Ananya was able to provide a voice to a group that was both marginalized traditionally and as a result of the extraordinary circumstances of the floods. The development of the CalamityRelief app enabled Ananya to develop her computational identity by recognizing a problem in the world that had personal connections (she has relatives living in the Chennai area) that could be solved computationally. Her ability to actually develop and deploy the app using MIT App Inventor instilled in her a sense of digital empowerment. As a result, Ananya now feels empowered to design computational solutions to new challenges in her everyday life. Ananya stated that next she wants "to develop more social apps – like for recycling garbage. Many people don’t know where recycling units are located, so they simply throw all their garbage wherever they want and it goes into landfills. [She] wants to develop an app to help people find where these recycling units are located." [14]. She has also recently developed an app that uses remote sensors to measure leaks in plumbing fixtures. At a
very young age, she has come to see herself as someone who can impact her world through computing.

In the cases presented above, youth were able to recognize and give a voice to communities that they perceived as being underrepresented: in Mood Ring, the youth recognized the need to address mental health issues affecting their friends and peers; and in CalamityRelief, Ananya responded to the unfortunate conditions that beset the inhabitants of Chennai, many of whom lacked the ability to seek help before the disaster struck. As a result, in both scenarios, these creators awakened their personal identities as empowered computational thinkers. We strongly believe that developing digital empowerment and computational identity will serve youth better than learning the syntax of a particular programming language in today’s increasingly digitally mediated society.

REFERENCES

[8] Blinded for review