Reducing the Barriers for Computational Action

How Blocks-Based Programming Can Support Digital Empowerment

Mike Tissenbaum†
College of Education
University of Illinois at Urbana Champaign
Champaign IL
miketissenbaum@gmail.com

Josh Sheldon
Computer Science and Artificial Intelligence Lab
Massachusetts Institute of Technology
Cambridge, MA, USA
jsheldon@mit.edu

Hal Abelson
Computer Science and Artificial Intelligence Lab
Massachusetts Institute of Technology
Cambridge, MA, USA
hal@mit.edu

ABSTRACT

Computational action, by which learners are given opportunities to build projects with real-world applicability, is, we argue, a preferable way to engage young people in learning how to program computers. Mobile computing, as the dominant form-factor of computing currently, provides an especially meaningful opportunity for youth to create these projects. Meanwhile, platforms with powerful abstractions, that make operations that might otherwise be difficult simple, build student capability to create projects that enhance their computational action.

KEYWORDS

Computational action, computational thinking, abstractions

1 Rethinking Computation Education Through the Lens of Computational Action

If we are serious about supporting students to become empowered creators and engaged citizens in their digital futures, rather than passive consumers of technology, we need to critically reframe the role that computing education plays in students' development. We advocate for an approach to computing education that we call *computational action*. In computational action, rather than focusing on all students in a class learning the same specific programming elements (e.g., loops, functions, databases), students are given opportunities to create with computing which have a direct impact in their daily lives and communities (Tissenbaum, Sheldon & Abelson, *in press*).

Implementing a computational action curriculum requires a **context**, that supports meaningful creation, and activities that foster **design thinking** and **authentic solution implementation**. Two key outcomes of computational action are changes in students **computational identity**, and **digital empowerment**.

1.1 Computational Identity

Building on prior research that showed the importance of young people's development of scientific identity for future STEM growth [6], computational identity is a person's recognition of themselves as capable of designing and implementing computational solutions to self-identified problems or opportunities. As part of computational identity, students should also see themselves as part of a larger community of digital creators.

Supporting students in developing their computational identity requires students to engage in activities that they feel are authentic to the practices and products of the broader engineering and computing communities. Also, rather than working towards predetermined solutions, students should feel that they have agency in identifying a need, and designing and implementing its solution.

1.2 Digital Empowerment

Digital empowerment builds on the work of Freire [3], which defines empowerment as students' ability to critically engage with issues that concern them, and that of Thomas and Velthouse [7], who see empowerment connecting to the concepts of meaningfulness, competence, self-determination, and impact. As a result, digital empowerment involves instilling young learners with the belief that they can put their computational identity into action in authentic and meaningful ways related to issues that matter to them.

Curriculum that aims to support students in growing their sense of digital empowerment needs to have activities that are situated in contexts that are authentic and personally relevant. Further, students need to feel that the products of their work have the potential to make an impact in their own lives or in their communities. As a result of this work, students should also feel capable of pursuing new computational opportunities, and be able to recognize how they might achieve them.

2 Supporting Computational Action Through Mobile App Development

To date, one of the challenges in developing curricula based on a computational action framing, is that most computing education has been constrained by the physical confines of traditional computing labs and its desktop computers. Even if students developed something with the potential to have an impact in their communities, actually getting out of the classroom and into the communities was difficult, if not impossible. By harnessing the growing ubiquity of mobile devices and their associated apps, we can free the work of students from the traditional confines of the

classroom walls, opening up new opportunities for students to directly impact their communities.

3 MIT's App Inventor: A Blocks-Based Approach to Computational Action Through Mobile App Development

While providing students the opportunity to develop mobile apps that can have an impact is a laudable goal, it is not without some significant challenges. Foremost among these is that most tools and programming languages for developing mobile apps are often exceedingly complex. The need to understand the often-arcane syntax and unique grammars of traditional programming languages has long been a barrier for engaging youth in computational practices [5]. In response, MIT has developed a mobile app development environment called MIT App Inventor that uses a blocks-based interface that allows users to develop fully-functional native mobile applications. Currently, MIT App Inventor has more than 24 million registered users (with over 1.05 Million unique monthly users) spread across 195 countries, who have collectively worked on more than 24 million mobile app projects. Given the size of the MIT App Inventor user base, the platform is in a unique position to have a direct impact on the computational identity and digital empowerment of children all over the world.

In the sections below we describe specific capabilities of MIT App Inventor, that embody powerful abstractions to support students in building apps that can be deployed for impact.

3 App Inventor Functionality to Support Computational Action

3.1 Maps

Geography plays an important part of the lives of all people, young people included [4]. Despite this, there are limited opportunities for these youth to shape their geographies or have a critical voice within them. Those opportunities they do have are often within the context of narrowly defined participation in social networks or "citizen science" projects [2]. Through the Maps component in MIT's App Inventor, we can empower students to design how they and their community move through, interact with, and share experiences within the physical spaces they inhabit.

By simply dragging the maps component into the App Inventor designer, users get access to a host of powerful tools including: geolocation, superimposing lines and polygons, and adding and annotating map markers.

With these tools, a young person could easily create an app for cataloging and sharing their favorite pieces of street art throughout a particular locale.

3.2 CloudDB

While creating applications that are locked to a single device can provide meaningful experiences, being able to share information and connect with a larger community offers exciting opportunities to virtually connect one's communities.

The CloudDB component in App Inventor allows users quickly and easily add cloud-based data storage capabilities to their apps. Moreover, the blocks that make up the component's functionality allow for easy push notifications whenever data is changed.

With this, the catalog of street art can be crowd sourced, with many users contributing to a database that lives in the cloud.

3.2 Internet of Things

Similar to the smartphone revolution making computers truly personal and mobile, the Internet of Things (IoT) is changing how we relate to the world around us and the objects within it [1]. By connecting IoT with mobile computing, we can create new avenues for students to develop computational interventions that extend into all aspects of students' lives.

While the promise of youth developing impactful IoT designs is exciting, similar to mobile app development, the usual technical complexity of IoT solutions remains a barrier. In response, MIT's App Inventor team has developed a set of extensions (http://iot.appinventor.mit.edu/) that allow students to create mobile applications that can send and receive information over Bluetooth between an Arduino or a Micro:Bit, popular and modular computing platforms for IoT.

Imagine, now, that a young person is to build a piece of IoT enabled street art that uses a phone connected to the database to sense when someone with the database app approaches and says hello.

3 Conclusion

This paper outlines the need to critically rethink our motivations for getting students engaged with and empowered by mobile computing. We believe there's a need to reframe computing education through the lens of *computational action*, by situating it in contexts that allow students to develop personally meaningful and impactful solutions in their lives and communities. In order to achieve this, we need to reduce the barriers that prevent them from putting their ideas into action. MIT's App Inventor is a platform that strives to make this possible. We have outlined three aspects of MIT App Inventor – Maps, CloudDB, and IoT – that hold particular promise for helping students realize impact in their lives, and in the process, help foster their computational identities and digital empowerment.

REFERENCES

[1] Ashton, K. (2009). That 'internet of things' thing. *RFiD Journal*, *22*(7), 97-114.

[2] Foth, M., Forlano, L., Satchell, C., & Gibbs, M. (Eds.). (2011). *From social butterfly to engaged citizen: urban informatics, social media, ubiquitous computing, and mobile technology to support citizen engagement*. MIT Press.

[3] Freire, P. (1993). *Pedagogy of the oppressed (20th anniversary ed.)*. New York, NY: Continuum.

[4] Knox, P., & Pinch, S. (2014). *Urban social geography: an introduction*. Routledge.

[5] Maloney, J., Burd, L., Kafai, Y., Rusk, N., Silverman, B., & Resnick, M. (2004, January). Scratch: a sneak preview [education]. In *Creating, connecting and collaborating through computing, 2004. Proceedings. Second International Conference on* (pp. 104-109). IEEE.

[6] Maltese, A. V., & Tai, R. H. (2010). Eyeballs in the fridge: Sources of early interest in science. *International Journal of Science Education*, 32(5), 669-685.

[7] Thomas, K. W., & Velthouse, B. A. (1990). Cognitive elements of empowerment: An "interpretive" model of intrinsic task motivation. *Academy of management review*, 15(4), 666-681.