Space for All: Learner Self-Construct While Learning in the Digital Age

Tutaleni I. Asino, Penny Thompson, and Kathy Essmiller
tutaleni.asino@okstate.edu, penny.thompson@okstate.edu, kathy.essmiller@okstate.edu
Oklahoma State University

Abstract: While there are numerous benefits to teaching coding in schools, current practices seem too focused on the vocational aspects of computer science, with computer programming as the only route to a successful future. The danger is that this approach excludes some students who may not be drawn to coding, leaving them to think that our technological society has no place for them or their unique talents, skills, and passions. In this paper, we present our work with third and fourth graders to use an instructional design framework to involve children in a design and development project that includes coding but also other non-coding activities. We argue that the instructional design process can be a way of engaging children in our technological society in meaningful ways while still making a place for a wide variety of skills and interests.

Introduction

Numerous scholars have written on the importance of having realistic expectations towards innovations, especially when it comes to the impact on education. For example, Cuban (1986, 2009) has written on innovations that were supposed to revolutionize education but have not delivered. Mayer (2005) emphasized the need to be learner-centered and focused on helping people learn rather than simply giving access to the newest tools. In our contribution to the *rethinking learning in the digital age* dialogue, we argue for a critical examination of a new innovation, teaching coding in school, particularly in the early grades.

The focus on coding in schools is based on the premise “that by learning to think like a computer scientist, students can solve everyday problems, design systems that we all use in daily life, and progress and innovate in other disciplines” (Kafai & Burke, 2014, p. 4). Our concern is that currently, teaching coding in schools overemphasizes the vocational aspects of computer science, with computer programming as the only route to success in the digital economy. We believe that programming is not the only option in a digital age. Contrary to Rushkoff’s (2010) dire warning to “program or be programmed” (pg. 8), we believe that this instrumental view excludes some students who may not be drawn to coding, leaving them to think that our technological society has no place for them or their unique talents, skills, and passions. We present our work with third and fourth graders that used an instructional design framework to involve children in a design and development project, which includes coding but also non-coding activities. We argue that the instructional design process can be a way of engaging children in our technological society in meaningful ways while still making a place for a wide variety of skills and interests.

Conceptual lens

Our study is guided by the concept of *possible selves* (Markus and Nurius, 1986) which is anchored in “the dynamic properties of self-concept” (p. 954). While self-concept has been described in various ways, what underlies most definitions is that an individual’s perception of the self “is continually reinforced by evaluative inferences and that it reflects both cognitive and affective responses” (Bong & Clark, 1999, p. 140). Possible selves are particularly concerned with the image individuals hold regarding what or who they would like to become. As Markus and Nurius (1986) put it, “Possible selves represent individuals' ideas of what they might become, what they would like to become, and what they are afraid of becoming, and thus provide a conceptual link between cognition and motivation (p 954).

What we are arguing in our work is that if a student develops a possible self as someone whose career has no place in our technology-driven society, this may have harmful impact on goals they set for themselves. For this reason, we feel there is a need to introduce the teaching of coding, particularly in the early grades, within a context that acknowledges and values a broad range of skills and interests.

Method

Our study began as a request from a teacher looking for more innovative ways to use the technology at her school. Over the course of three weeks, we engaged in design activities with students at a private school in the

Midwest United States. Students in the third and fourth grades (ages 8 to 10) worked in teams to design a mobile application that met a need identified by the children themselves.

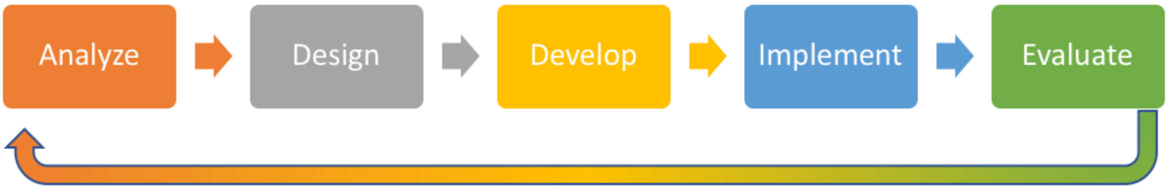

Figure 1. The ADDIE Framework.

Our study used the ADDIE framework (Branch, 2009) (figure 1) to guide the process and provide opportunities for children to apply a diverse set of skills toward accomplishing a common goal. The designed activities consisted of the following five stages: (1) students formed teams, analyzed their potential audience, and identified needs for their potential app; (2) teams designed and documented a detailed plan for the mobile app; (3) teams developed a working prototype of their mobile app using the MIT APP Inventor (http://appinventor.mit.edu/explore/) platform; (4) teams implemented the app by sharing it with their classmates for usability testing; (5) the intention at this stage was to have students evaluate the app and reflect on what they would like to change to improve it.

Data for this study was gathered through surveys, observations, video recordings and collection of artifacts. A brief pre and post survey was given to students to obtain their self-perceptions of their abilities to work with computers as well as the types of jobs they envisioned themselves having in the future.

Findings and discussion

Using the ADDIE framework, all students were able to contribute to the design of a mobile application using a range of skills such as coding, drawing, writing, or communicating orally. Preliminary analysis suggests that students ended the project feeling confident in their skills but still holding misconceptions that "working with computers" equaled computer programming. As we continue to analyze the data further we are also planning to enact a second version of this study at a different school, where we expect the demographic and socioeconomic status will be different.

Teaching programming in schools has many benefits. Luminaries such as Pappert (1980) long argued for the transference of skills from programming to literacies and math. Our research aim was not to minimize the importance of coding but rather to argue against the seemingly emerging perspective that the only way that students of the future can succeed and have a fulfilling life is if they are coders. Contextualizing a coding project within a broader design project provides young students with a chance to envision possible selves that include their diverse talents and interests and are valued in the digital age.

References

Bong, M., & Clark, R. E. (1999). Comparison between self-concept and self-efficacy in academic motivation research. *Educational psychologist*, *34*(3), 139-153.

Branch, R. M. (2009). *Instructional design: The ADDIE approach* (Vol. 722). Springer Science & Business Media.

Cuban, L. (1986). *Teachers and machines: The classroom use of technology since 1920*. Teachers College Press.

Cuban, L. (2009). *Oversold and underused*. Harvard University Press.

Kafai, Y. B., & Burke, Q. (2014). *Connected code: Why children need to learn programming*. MIT Press.

Markus, H., & Nurius, P. (1986). Possible selves. *American psychologist*, *41*(9), 954.

Mayer, R. E. (Ed.). (2005). *The Cambridge handbook of multimedia learning*. Cambridge university press.

Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. Basic Books, Inc., 1980.

Rushkoff, D. (2010). *Program or be programmed: Ten commands for a digital age*. Or Books.

Stevenson, J., & Clegg, S. (2011). Possible selves: Students orientating themselves towards the future through extracurricular activity. *British Educational Research Journal*, *37*(2), 231-246.