Hybrid Environments: A Bridge from Blocks to Text

Jeremiah Blanchard
Department of CISE
University of Florida
Gainesville, FL, USA
jblanch@cise.ufl.edu

ABSTRACT

Hybrid, dual-modality programming environments provide both blocks-based and text-based interfaces for programming. While previous research investigated the transition from visual to textual environments, few studies considered these hybrid environments. The purpose of this dissertation is to explore how hybrid programming environments impact computer science competency, confidence, and interest in computer science among students when moving from blocks-based environments to text-based languages. Exploring these questions will help us understand which hybrid environments are effective, in which contexts they are effective, and if they can improve on current approaches to CS instruction.

CCS CONCEPTS

• K.3.2 [**Computers and Education**]: Computer and Information Science Education – *computer science education.*

KEYWORDS

Computer Science Education, programming environments, programming languages, novice programming, hybrid programming environments.

1 PROGRAM CONTEXT

I have passed the University of Florida CISE department's Qualifying Exam. I recently conducted a study investigating the effects of hybrid environments on student competency and attitudes. The study compared students in text-based and blocks-based instructional conditions. I plan to propose in fall 2017.

2 CONTEXT AND MOTIVATION

Computer science is increasingly taught in K-12 education, making it important to find approaches that are accessible to younger audiences. Learning CS can be challenging as it requires developing several skills, including computational thinking, programming, and algorithm use [1]. These abilities are distinct and carry unique cognitive loads [2]. Visual languages and environments have been developed to address syntax challenges and accessibility. However, interviews with students suggest some learners have difficulty moving from blocks-based environments to text-based languages [3]. Learners may also perceive the programming experience as inauthentic [4].

Hybrid blocks-text programming environments, having representations in text and blocks, allow quick transition between blocks and text, these environments may help learners overcome actual and/or perceived difficulties of text-based languages. As hybrid environments are relatively new, studies of their effectiveness are limited. If we establish that hybrid environments provide an effective bridge between blocks-based environments and text languages, they can serve as a scaffolding when moving from basic CS principles introduced in blocks to more advanced CS topics that require text languages in practice.

3 BACKGROUND AND RELATED WORK

Several environments will translate from blocks to text, such as Alice 3's translation of blocks to Java code [5]. Others also translate from text to blocks. One such environment is Pencil Code and its Droplet Editor, which let users go between blocks- and text-representations at any time [6]. In doing so, Pencil Code lets users move from blocks to text at their own pace. Studies of bi-directional hybrid environments are limited but promising. Bau et al. showed that students use text representation more often over time, suggesting they use the text mode more often with experience [6]. Weintrop studied students starting in blocks, text, or hybrid versions of Pencil Code's JavaScript variant before moving to text-based Java instruction [7]. His work showed that students who began in a blocks-based environment or worked in the bidirectional hybrid environment scored similarly [7]. However, unlike blocks-based environment students, hybrid environment students had an increase in interest in taking computer science courses in the future after switching to text.

Weintrop's work provides a foundation for future work. One open question is whether changing languages (from JavaScript to Java) impacts competency learning and motivations to engage in CS in the future. Changing languages requires learning not just new syntax, but also control structures, potentially increasing cognitive load on learners and impacting perceptions and learning.

4 QUESTIONS, GOALS, & METHODS

4.1 Goals & Questions

Goals

My goals include measuring comparative differences in attitudes and competency regarding computer science among learners who learn computer science via different conditions (text, blocks-to-text, and blocks-to-hybrid-to-text). In addition, I seek to measure how students' attitudes regarding CS, especially confidence in their ability and future interest, are impacted by learning environments.

Questions

When used as a bridge between blocks and text, how do hybrid environments impact student competency and perception as compared to unbridged transitions and learning purely in text?

1.0 How do hybrid environments impact student learning of CS competencies?

1.1 How do conditions (blocks, text, and hybrid) impact student learning in CS generally and on specific topics?

1.2 What challenges do students encounter when programming in blocks and text and when transitioning?

2.0 How do hybrid environments impact student perceptions of computer science and their abilities?

2.1 How do hybrid environments impact student confidence?

2.2 How do hybrid environments impact interest in CS?

4.2 Methods

4.2.1 Environment

Pencil Code actively developed, but its language support was limited to JavaScript, whose syntax can be very complex, and CoffeeScript, which has limited use in industry and academia. By comparison, Python has been recognized as an effective learning language [8] which is in common production use. My initial work involved integration of a Python runtime into Pencil Code.

4.2.2 Instruments

To measure student attitudes and competency, I developed both (1) a survey to identify changes to and differences in attitudes regarding computer science, and (2) a series of assessments in multiple, comparable formats (blocks, text, and mixed-mode variants). The assessment drew from question styles and competencies outlined in the SCS1 [9] and Computer Science Principles manual [10], while the survey instrument was based on prior research on gauging attitudes about computing [11,12].

Study Design

I conducted an initial study at a middle school over five weeks. Six classes of about 30 students each were taught the fundamentals of computer science and the Python language in customized versions of Pencil Code. Students were divided into three conditions of about 60 students (two classes) each. Students in the text-condition, the control, learned exclusively in text. Those in the blocks-condition experimental group began in blocks before transitioning directly to text, while a hybrid-condition group began in blocks but moved into the hybrid environment before transitioning to text. Both experimental groups transitioned to text at the midpoint. Students in all groups were assessed and surveyed at the beginning, midpoint (just before the transition in experimental groups), and end of the study. The pre-assessment used only blocks-based questions, the mid-assessment varied according to group condition (blocks, mixed, and text respectively), and the post-assessment used only text. Based on the outcomes of this study, a follow-up study is also planned next year that will follow a similar design but over a longer instructional period.

5 DISSERTATION STATUS

The dissertation proposal is in its early stages of development. I am outlining and drafting the first version. Results from the initial study will also be incorporated. In late 2017, I will my dissertation. In spring of 2018, I am planning a follow-up study to build on the initial study's results. A complete draft of the dissertation will be done in the summer/fall of 2018, followed by several months of iteration. I expect to defend my dissertation in the spring or summer of 2019 to complete my PhD studies.

6 CONSORTIUM GOALS

My goals at the consortium are to connect with other CS education researchers in order to learn about and understand strands of research currently underway. It is also an opportunity to receive constructive feedback on my dissertation's topic (hybrid environments), specificity (focus on bridging), and study designs. Knowing about ongoing research and hearing feedback will aid exploration of research questions and the design of future studies.

7 EXPECTED CONTRIBUTIONS

Research to show impacts of hybrid environments on learners of CS would be a valuable contribution to CS education. This research will identify benefits and drawbacks of using hybrid environments as a bridge from blocks and text. It will provide a statistical comparison of learning outcomes and attitudes in students who transition from blocks to text via a hybrid environment versus those who do not. By identifying contexts in which hybrid environments are effective, this dissertation will provide a key foundation for further research into hybrid environments.

REFERENCES

[1] JTF on Computing Curricula - ACM/IEEE-CS, 2013. *Computer Science Curricula 2013: Curriculum Guidelines for Undergraduate Degree Programs in Computer Science*, 13-14. Retrieved from http://www.acm.org.

[2] J. Sweller, "Cognitive Load During Problem Solving : Effects on Learning," *Cogn. Sci.*, vol. 12, no. 2, pp. 257–285, 1988.

[3] D. J. Malan and H. H. Leitner, "Scratch for Budding Computer Scientists," in *Proc. SIGCSE2007*, (2007), 223–227.

[4] D. Weintrop and U. Wilensky, "To block or not to block? That is the question," In *Proc. IDC2015*, (2015), 199–208.

[5] W. Dann, D. Cosgrove, and D. Slater, "Mediated transfer: Alice 3 to Java," in *Proc. SIGCSE2012*, (2012), 141–146.

[6] D. Bau, D. A. Bau, C. S. Pickens, and M. Dawson, "Pencil Code: Block Code for a Text World," in *Proc. IDC2015*, (2015), 445–448.

[7] D. Weintrop, "Modality Matters: Understanding the Effects of Programming Language Representation in High School Computer Science Classrooms," Ph.D. dissertation, Northwestern University, 2016.

[8] R. J. Enbody, W. F. Punch, and M. McCullen, "Python CS1 as preparation for C++ CS2," *ACM SIGCSE Bull.*, vol. 41, no. 1, pp. 116–120, 2009.

[9] M. C. Parker and M. Guzdial, "Replication, Validation, and Use of a Language Independent CS1 Knowledge Assessment," *Proc.ITiCSE2016*, (2016), 93-101.

[10] College Board, 2017. "AP Computer Science Principles Course and Exam Descrption," 85-124. Retrieved from http://www.collegeboard.org.

[11] Marcu, Gabriela, et al. "Design and evaluation of a computer science and engineering course for middle school girls." In *Proc. SIGCSE2010*, (2010).

[12] B. Dorn and A. E. Tew. "Empirical Validation and Application of the Computing Attitudes Survey," *Computer Science Education*, 25(1):1-36, 2015.