Promoting Active Learning & Leveraging Dashboards for Curriculum Assessment in an OpenEdX Introductory CS Course for Middle School

Shuchi Grover
Graduate School of Education
Stanford University
shuchig@stanford.edu

Roy Pea
Graduate School of Education/
H-STAR Institute
Stanford University
roypea@stanford.edu

Stephen Cooper
Computer Science Department
Stanford University
coopers@stanford.edu

ABSTRACT

Lack of teachers to teach computer science (CS) and pedagogically sound introductory CS curricula remain a significant challenge facing secondary schools attempting to teach CS. This paper describes our efforts to design and pilot an online 6-week middle/high school course using Stanford's OpenEdX platform. The pedagogy, curriculum and assessment are guided by learning theory. The course leverages OpenEdX features for contextual discussions and multiple-choice assessments that promote student learning and provide feedback. The paper reports on experiences in using instructor dashboards to identify targets of student difficulty and to aid curriculum redesign.

Author Keywords

K-12 CS Education; Blended Learning; Instructional Design; Learning Theory; Instructor Dashboards; Assessment; Analytics; OpenEdX; MOOC.

ACM Classification Keywords

H.5.m. Information interfaces and presentation (e.g., HCI): Miscellaneous.

INTRODUCTION

As educators and policymakers attempt to take computer science education to scale in K-12, a severe shortage of trained teachers and the lack of curricula with sound pedagogical underpinnings and assessment measures remain significant challenges [3]. Any large-scale effort today that must reach thousands of classrooms and teachers must look to state-of-the-art online teaching mechanisms. MOOCs currently represent a possible massive shift in the higher education landscape. Given the shortage of CS teachers and curricula, well-designed online courses deployed on MOOC platforms could serve a crucial need in K-12 as well. Such courses could be used for blended learning in addition to purely online use by students in and out of classrooms, and as a means to prepare teachers as well. Online venues like Khan Academy, CodeAcademy and Code.org currently cater largely to the development of skills in programming and their success as structured CS curricula for K-12 classrooms has not been empirically tested. These efforts also lack robust measures for assessing student learning which limits their use in K-12 learning settings where issues of assessment are paramount.

L@S 2014, Mar 04-05 2014, Atlanta, GA, USA
ACM 978-1-4503-2669-8/14/03.
http://dx.doi.org/10.1145/2556325.2567883

OUR RESEARCH ON A CS MOOC FOR K-12

This paper describes our work in progress around a six-week **online** introductory CS mini-course for middle school, titled ***"Foundations for Advancing Computational Thinking" (FACT),*** and created and deployed on the Stanford OpenEdX platform. Based on Exploring CS (www.exploringcs.org/), it is designed to build awareness of computing as a discipline while engaging students in foundational computational concepts such as algorithmic flow of control- sequence, looping, and conditional logic using Scratch. In Fall 2013, we piloted this MOOC as a SPOC [5] to study its effectiveness in a local public middle school "Computers" elective class that met four times a week for 55-minute periods.

Promoting Active Learning in a Video-Based Course

This section presents some of the pedagogical approaches adopted in the curriculum:

- Builds on the rich body of prior research involving children and novice programmers to guide the pedagogy and assessments for the content being taught. These include: using of worked examples for conceptual learning [6], using pseudo-code, teaching reading/code-tracing [2], and using frequent multiple-choice "quizzes" to reinforce concepts [1] including innovative ones like "Parson's puzzles" [4].
- Promotes active, constructivist learning in Scratch through several hands-on activities and assignments.
- As design research [7], this study actively solicits feedback from students on various aspects of the course through short online surveys inserted inconspicuously throughout the course.

Day 5	Comps-Dumb+powerful (3m24s)	Think & Write- What can computers not do (5 mins)	types of instructions (1m53s)	Patterns in human instructions (1m55s)	Algorithms (3 m 16s)	Share a Recipe			
Day 6	Characteristics of Algorithms (3m 43s)	Quiz 3 questions (3 mins)	Sequence; Repetition; Selection (7 mins)	Programs & programming Languages (4m 09s)	Quiz (Algorithms & programs) 5 ques; 5 min	First Program Example (4m 38s)	Scratch Assignment: Science Life Cycle		
Day 7	Vid #20 Intro to loops - 4m	Quiz 7 questions (5 mins)	Make a polygon in Scratch (5 mins)	Intro Spirograph activity (5 mins)	Nested Loops vid - 2m51s	Quiz - 5 mins	Rings Demo video - 4m02s	Scratch Assignment:Make a Spirograph	Optional Quiz (2 questions)
Day 8	Initialization - 1min	Quiz - 4 ques (5 min)	Scratch Assignment - 4 part problem	Optional Scratch Assignment - olympic rings	Kids helping other kids catch-up				

Figure 1: Learning sequences planned for a week's worth of classroom time (over four 55 minute periods) using OpenEdX FACT

The course comprises roughly 60 short lecture and demonstration videos varying between 1.5 to ~6 minutes in length interspersed with quizzes with automated grading and explanations (which the students found to be very useful based on post-course feedback). The video length was based on student feedback following a short online unit that ran as a pilot in Spring 2013. Additionally, the modular nature of the OpenEdX platform design that allows various types of "elements" to be added to a page aided the use of contextual discussion prompts below instructional videos, and HTML code to "iframe" http://scratch.mit.edu below the video so students could try out the ideas taught in the video. Qualtrics surveys were similarly "iframe"-d in to obtain student feedback as well as responses to open-ended "thought questions" which students completed at the end of a video lecture. Figure 1 above shows a sample learning "sequence" covered by students in a 55-minute classroom period. The different colors indicate the type of learning activity. Videos are in the lightest color- lemon, videos with activities below them in light peach, thought questions in dark peach, Scratch assignments in orange, quizzes in dark orange, and lastly, extension assignments are shown in red.

Leveraging Dashboards for Analyzing Student Learning and Promoting Curriculum Assessment

While a primary goal of automated assessments is to assess student learning, data and metrics generated by MOOC platforms are also enormously useful as aids to curriculum evaluation as well. We actively used instructor dashboards on quiz data provided by OpenEdX to ascertain targets of difficulty and prompt course revision, where necessary. For example, on a quiz that tested variable manipulation in loops, OpenEdX metrics suggested huge gaps in understanding on even simple questions. Figure 2 shows a screenshot of the dashboard. This prompted additional explanations and Scratch assignments to be uploaded to aid further understanding and reinforce the concept.

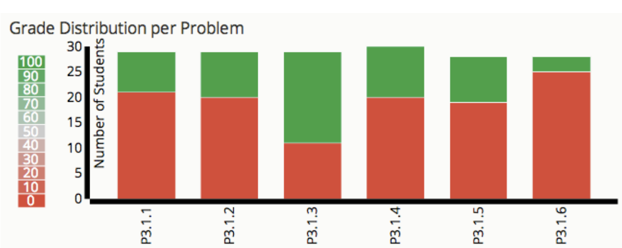

Figure 2. Screenshot of OpenEdX instructor dashboard showing useful feedback on conceptual gaps in student understanding

CONCLUSION & FUTURE WORK

Our instruments measuring computational learning showed gains for all students using the FACT SPOC. More importantly, students on OpenEdX FACT performed as well or moderately better on all questions of the post-tests than students in an earlier face-to-face version of the course Post-course student feedback has been encouraging as well.

Following successful completion of the 6-week course, we now plan to make the material available as a MOOC for large-scale use and for teachers to use in various classroom settings. We are developing accompanying online teacher guides as well. FACT exemplifies the application of existing learning theory and design-based research methods to MOOC course design. More importantly, this course represents crucial first steps to leverage MOOC platforms in K-12 school settings to address a huge need in providing effective CS education at significant scale.

ACKNOWLEDGMENTS

We gratefully acknowledge grant support from the National Science Foundation (NSF #1343227).

REFERENCES

1. Glass, A. L., & Sinha, N. Multiple-choice questioning is an efficient instructional methodology that may be widely implemented in academic courses to improve exam performance. *Current Directions in Psychological Science*, *22*, 6 (2013), 471-477.
2. Lopez, M., Whalley, J., Robbins, P. & Lister, R. Relationships between reading, tracing and writing skills in introductory programming. *In Proc. ICER 2008*, ACM Press (2008).
3. NSF, Increasing accessibility to computer science education across the US. http://www.nsf.gov/news/news_summ.jsp?cntn_id=129882
4. Parsons, D. and Haden, P. Parson's programming puzzles: a fun and effective learning tool for first programming courses. *In Proceedings of the 8th Australasian Conference on Computing Education-Volume 52* (2006), 157-163.
5. Rivard, R. Three's Company. *Inside Higher Ed* (2013). http://www.insidehighered.com/news/2013/04/03/stanford-teams-edx
6. Sweller, J., & Cooper, G. A. The use of worked examples as a substitute for problem solving in learning algebra. *Cognition and Instruction*, *2*, 1 (1985), 59-89.
7. The Design-Based Research Collective. Design-based research: An emerging paradigm for educational inquiry. *Educational Researcher* (2003), 5-8.