The Challenges of Using a MOOC to Introduce "Absolute Beginners" to Programming on Specialized Hardware

Jennifer S. Kay
Computer Science Dept., Rowan University
201 Mullica Hill Road, Glassboro, NJ, USA
kay@rowan.edu

Tom McKlin
SageFox Consulting Group
1201 Clairmont Road, Decatur, GA, USA
tmcklin@sagefoxgroup.com

ABSTRACT
Educational Robotics for Absolute Beginners is a MOOC designed to introduce K-12 teachers with no prior computer science or robotics experience to the basics of LEGO NXT Robot programming. The course was developed following several successful in-person workshops on the same topic. This paper introduces some of the issues that arose as we transitioned the material to a MOOC, describes some of the unique challenges we faced by incorporating specialized hardware into a MOOC, and presents some preliminary data evaluating the success of our approach.

Author Keywords
MOOC; Educational Robotics; LEGO Mindstorms; K-12 Teacher Education.

ACM Classification Keywords
H.5.2. Information interfaces and presentation (e.g., HCI): User Interfaces – Training, help, and documentation.

K.3.2. Computers and Education: Computer and Information Science Education – CS Education.

INTRODUCTION
Educational Robotics for Absolute Beginners is a MOOC (available at https://cs4hsrobots.appspot.com/) designed to introduce K-12 teachers with no prior computer science or robotics experience to the basics of LEGO NXT Robot programming. The goal is to enable K-12 teachers to use robotics to incorporate computer science and computational thinking concepts into classes or after-school activities.

The course is designed to enable asynchronous learning. We used *Course Builder* (https://code.google.com/p/course-builder/) following the model of Google's *Power Searching* course (http://www.powersearchingwithgoogle.com/). A typical *lesson* includes a five- to ten-minute video as well as a set of self-test questions that are graded automatically. A *week* of the course consists of 5-10 lessons, and the course was five *weeks* long. The initial run of the course attempted to encourage participants to work at roughly the same pace by releasing one week's worth of lessons at a time with the hopes that this might make the course forum more relevant.

This work-in-progress paper introduces some issues that arose as we transitioned the material to a MOOC, describes some of the unique challenges we faced by incorporating specialized hardware into a MOOC, and presents some preliminary data evaluating the success of our approach.

WHAT'S THIS MOOC WORTH?
We typically offer *Professional Development* (PD) credit to teachers who complete our in-person workshops. However, the question of how to accurately estimate the number of hours that someone might spend on a MOOC seems to us to be a challenging research question in itself and so we chose not to offer formal PD to teachers for this MOOC. We did, however, want to offer a certificate of completion to those teachers who successfully completed all five of the robot programming projects. But this raised the question, how does one remotely grade a robot programming project?

REJECTING AUTOMATED GRADING
The LEGO NXT-G programming language is a graphical language in which programmers drag *blocks* onto a *sequence beam*. For example, Figure 1 shows a two-block program that says "good morning" and then drives forward for one second. Parameters such as duration of movement are specified in additional *configuration panels* which are not shown due to space limitations.

Automated grading of NXT-G programs seems difficult, if not impossible, to implement. There are, of course, a multitude of working solutions to a given problem. However the biggest challenge is that the correctness of a given program is also dependent on both the design of the robot it will run on as well as on the accuracy and precision of that robot's motors and sensors.

Figure 1: A Short NXT-G Program

L@S 2014, March 4–5, 2014, Atlanta, Georgia, USA.
ACM 978-1-4503-2669-8/14/03.
http://dx.doi.org/10.1145/2556325.2567886

DISTRIBUTING THE GRADING – BUT NOT TO PEERS

We rejected implementing a peer-grading scheme because we felt that our target audience might find the uploading process too burdensome. Instead, we took advantage of the fact that most members of our target audience of K-12 teachers already had school principals who routinely evaluated them. If the principals were willing to place a value in a certificate of completion, we reasoned, perhaps they would be willing to assess robot programming assignments. Furthermore, the project demonstrations could serve not only as a means of evaluating performance, but also as a way for teachers to share what they had learned with others and to demonstrate to school leaders how CS/robotics learning might take place.

Thus, each of our robot programming projects has two sets of instructions: one for the participant and another for the reviewer. Reviewers evaluate projects solely on robot performance and do not need programming experience.

INCREASING RESEARCH SURVEY RESPONSE RATES

We routinely set aside time at our in-person workshops for our participants to fill out surveys to support our evaluation efforts. Participation is optional, but we typically have a high response rate. We hypothesized that MOOC participants would be less inclined to provide feedback. To encourage participation, we created a *lesson* in our MOOC for each survey whose video strongly encouraged participation. We also required that participants who wanted to receive a certificate *submit* a survey. The survey began by asking if they were over 18, and if so, whether they were a teacher. We then explained in a statement (that they were required to acknowledge), that answers to all subsequent questions were completely optional.

This approach seems to have been successful. For each of our pre, mid, and post surveys:

- Over 50% of those who had done at least some portion of that week's lessons *visited* the survey.
- Over 75% of those who *visited* a survey responded to at least one optional question.

EXTENDING THE TIMELINE

Given our goal of introducing as many teachers as possible to this material, we saw little benefit to requiring teachers complete the course within the official 5-week period (particularly since our course began in mid-November – a busy time for everyone). Not only did we see participants continuing to progress after week 5 "concluded," but we also continued to see new participants joining the course.

SUPPORTING PARTICIPANTS WITHOUT HARDWARE

We wanted to make at least part of our course accessible to those who did not have access to the hardware as well as to administrators who might be interested in an overview of the material. Thus, we designed (and advertised) the week 1 *lessons* so that they did not require hardware.

OFFERING EXTRA HELP

Our in-person workshops gave us significant insight into common problems that novices encountered with the LEGO NXT robotics programming environment and hardware. We created an "appendix" of "when something unexpected happens" videos designed to support participants.

VERY EARLY DATA:

Seven weeks after launching our MOOC, we have over 1100 participants enrolled. Over 725 have partially or fully completed week1, >325 week2, >200 week3, >125 week4, >75 week5, and >80 appendix.

Table 1 presents an analysis of data collected in the post-course survey. Participants are significantly more confident in their ability to learn CS/robotics and report significantly greater knowledge and skill. Gathering pre/post retrospective data (i.e. asking participants at the conclusion of the course to simultaneously rate their starting and current confidence and knowledge) proved to be beneficial since in many cases those who completed a pre survey did not also complete a post survey, or vise versa. Offering a pre/post retrospective mitigates response-shift bias, diminishes the surveying burden, and increases the usable set of paired samples that can be analyzed for significance testing.

Preliminary data also indicate that those who primarily teach CS, math, or science are much more likely to persist in the course to the end. Further, teachers with one year of teaching or less are more likely to be retained in the course than those with 2-10 years of experience (19% vs. 5%).

CONCLUSION

The course began in mid-November, and this paper is being written at the start of the New Year. It will be interesting to see whether an analysis of the data collected in a few weeks, as more participants complete the course, shows any differences. In particular, it seems plausible that non-STEM teachers may take more time to finish the course.

ACKNOWLEDGMENTS

This work was made possible thanks to a generous grant from Google's CS4HS Program.

Items (n=40)		Mean	Paired samples t-test
I would rate my confidence to learn the materials in this course as…	Start	3.86	<0.001 **
	Now	4.55	
I would rate my knowledge or skills in computing and robotics as…	Start	3.10	<0.001 **
	Now	4.24	

Table 1. Confidence and Knowledge