Using App Inventor in Introductory CS Courses

Meimei Gao
Mercer County Community College
Business & Technology Division
1200 Old Trenton Road
West Windsor, NJ 08550
gaom@mccc.edu

Julie Johnson
Vanderbilt University
Elec. Eng. & Comp. Sci. Dept.
2301 Vanderbilt Place
Nashville, TN 37235
Julie.l.johnson@vanderbilt.edu

Dale Reed
University of Illinois at Chicago
Dept. of Computer Science
851 S. Morgan St. (MC 152)
Chicago, IL 60607-7053
reed@uic.edu

Cate Sheller
Kirkwood Community College
Math/Science Dept.
6301 Kirkwood Blvd SW
Cedar Rapids, IA 52402 USA
Cate.Sheller@kirkwood.edu

Franklyn Turbak (Moderator)
Wellesley College
Computer Science Dept.
136 Central St.
Wellesley MA, 02481
fturbak@wellesley.edu

Categories and Subject Descriptors

K.3.2 [**Computers and Education**]: Computer and Information Science Education—*Computer science education*; D.2.6 [**Programming Environments**]: Graphical environments

General Terms

Human Factors, Languages

Keywords

CS0, CS1, mobile application programming, visual programming, blocks programming

1. SUMMARY

MIT App Inventor [3] is a visual programming environment targeted at novices for creating mobile apps for Android devices. It lowers barriers to programming and app development through a browser-based visual blocks language that reduces syntactic errors and encapsulates mobile device features into high-level abstractions that are easy to incorporate into apps. This panel explores how App Inventor can be integrated into undergraduate computer science and engineering courses to introduce computational thinking [5, 6] in the context of building mobile apps.

The moderator will begin with an overview of the NSF TUES *Computational Thinking through Mobile Computing* project [1], which has developed numerous materials to support the teaching of computational thinking with App Inventor and has sponsored several workshops for undergraduate faculty interested in using App Inventor in their courses.

The panelists were participants in a 2.5-day Summer 2013 workshop [4] who are now teaching with App Inventor in their courses. The panelists will describe how they use App Inventor in their courses. The panelists represent many different perspectives, including the type of institution (community college, liberal arts college, university), type of course (CS0, CS1), and level of App Inventor integration in the course (short module, short course, full-semester course).

The presentations by the moderator and the panelists will be limited to 8 minutes each, leaving over 30 minutes for discussion based on questions from the audience. We expect discussion topics to include student diversity, whether students take additional CS courses, and Android devices vs. emulators.

2. FRANKLYN TURBAK (MODERATOR)

In addition to teaching a full-semester CS0 course, *Inventing Mobile Apps*, based on App Inventor, I am the lead Principal Investigator of the *Computational Thinking through Mobile Computing* project. This NSF TUES project is a collaboration between Wellesley College, MIT, Trinity College, University of Massachusetts Lowell, and the University of San Francisco.

Making the situated nature of mobile devices (including sensors, location awareness, social networks, and Internet connectivity) accessible to introductory students opens up compelling opportunities for computer science education. Our team has identified big ideas of *mobile computational thinking* that augment traditional problem solving and programming concepts: mobile computing is event-based, leverages device features, emphasizes useful programs embedded in social contexts, takes advantage of a larger informational ecosystem, and involves design, engineering, and entrepreneurship.

We have developed online curricular materials (web-based tutorials, video lectures, screencasts, programming exercises, blocks-based quizzes, and maker/concept cards) that use App Inventor to teach mobile computational thinking. We have also created and are evaluating techniques (including surveys and project rubrics) for assessing students' computational thinking knowledge in the context of our courses and materials. Finally, we have launched an App Inventor educators website [2], where members share resources and discuss how to teach with App Inventor.

SIGCSE '15 Kansas City, Missouri USA
ACM 978-1-4503-2966-8/15/03.
http://dx.doi.org/10.1145/2676723.2677335 .

3. MEIMEI GAO

I started teaching App Inventor in Fall 2013, soon after I completed the *Computational Thinking Through Mobile Computing* workshop held at University of Massachusetts Lowell that summer. Since then, I have been teaching this CS0 course every semester. The course is called *Introduction to Programming with Mobile App Development.* It is a new course that is designed to help students with little or no programming experience to learn programming concepts; to help students, regardless of their major, to explore the world of computing and new technologies; and to increase students' interest and confidence in programming. App Inventor serves these purposes very well. The graphical drag-and-drop development environment helps students to focus on problem solving and logic thinking without worrying about syntax. Students learn programming concepts and explore computing technologies by building interesting and useful mobile apps.

I teach the course in hybrid format — 50% face-to-face meeting and 50% online activities. Students are asked to follow the online tutorials to create an app before the class. In class, we discuss the problems and learn the concepts from the app and students then continue working on their app by adding more functionality or create a new app using the similar concepts. This course has attracted students from 11 different majors.

4. JULIE JOHNSON

I have just completed my third year of teaching App Inventor to first year engineering students. The class is unique in several ways. It is a 14-lecture module that is offered three times each semester. The students are not necessarily CS majors. Most of them have little to no programming experience and many do not anticipate studying computer science beyond a CS1 course as required by their major. This short-course model necessitates that a lot of information is packed into a course that is meant to be fun and engaging, introducing students to the creativity that CS can spark. A secondary goal is to get students to the point where they can begin completing mobile apps on their own as early as possible in the module. I have seen first hand how quickly students can grasp the basic constructs needed to start building working apps, and how App Inventor facilitates the discussion of complex programming structures at a level that most students can grasp. Using App Inventor, I have been able to engage students of varying interest levels and experience in a single class, providing them with the resources and freedom to create and discover how fun and natural programming can be.

5. DALE REED

I've used App Inventor in my CS0 introductory computer science course for non-majors (CS 100) for the last two years. I struggled trying to use App Inventor briefly during the Spring 2013 semester, but it wasn't until I participated in the Summer 2013 App Inventor workshop that I was equipped to implement it as a significant component of my CS 100 course. App Inventor provides the ease of use of a visual programming environment coupled with the caché of creating software that runs on students' phones.

We use App Inventor to introduce the ideas of variables, assignments, mathematical and relational operators and lists, all within the context of creating games and utilities that run on students' phones. While they have not pursued it, several students each semester create apps that would be attractive to a wider audience. Creating apps with App Inventor has the added bonus of being able to use location and motion sensors, as well as the ability to store shared information such as high scores for games in the Cloud. Students readily see the ability to write programs for their phones as a relevant skill for many different majors.

6. CATE SHELLER

I use App Inventor in my Engineering Problem Solving course as a gentle introduction to programming for engineering majors. This CS1 course covers basic programming concepts (flow of control, procedural abstraction, data handling) in the context of engineering problems (specifically in statics, circuits and engineering economy), and requires that students use C and MATLAB. I have incorporated App Inventor into the first four weeks of the course, enabling students to learn programming concepts without being burdened with syntax, and to think about solving problems in their everyday lives using a device (Android phone) that most of them carry around. My experience thus far has been positive; students like the idea of writing programs they can use and share with others, and they make the transition into C already knowing the basic principles underlying the programs they will write in a less friendly language. Decoupling the conceptual hurdles from the syntax frustration seems to make programming less intimidating for these students, most of whom have no prior programming experience.

I also use App Inventor as the basis for a faculty development workshop I entitled *Programming for Everyone.* I am trying to get my colleagues to start thinking about computer science, or programming at any rate, as a core skill (a p to go with the three r's) that all students (and their teachers) should have some aptitude in. In my 2-hour workshop, I had my colleagues develop a simple app (Magic 8-ball) and brainstorm about how they might introduce programming as a supplementary skill in their own disciplines. I plan to repeat the workshop this November.

7. ACKNOWLEDGMENTS

This work was supported by the National Science Foundation under grants DUE-1225680, DUE-1225719, DUE-1225745, DUE-1225976, and DUE-1226216.

8. REFERENCES

[1] Computational Thinking through Mobile Computing. NSF TUES project website, `http://nsfmobilect.wordpress.com`, accessed Sep. 5, 2014.

[2] MIT App Inventor for Educators website. `http://educators.appinventor.mit.edu`, accessed Sep. 5, 2014.

[3] MIT App Inventor website. `http://appinventor.mit.edu`, accessed Sep. 5, 2014.

[4] Mobile CT Workshop June 2013 website. `https://sites.google.com/site/mobilectworkshopjune2013/`, accessed Sep. 5, 2014.

[5] T. Raja. We can code it: Why computer literacy is key to winning the 21st century. *Mother Jones*, June 2014.

[6] J. Wing. Computational thinking. *CACM*, 49(3), Mar. 2006.