Enhancing CS1 with Mobile Apps

Ivaylo Ilinkin
Gettysburg College
Gettysburg, PA, USA
iilinkin@gettysburg.edu

Sunghee Kim
Gettysburg College
Gettysburg, PA, USA
skim@gettysburg.edu

ABSTRACT

This workshop will introduce an Android library that is designed to fit seamlessly in a traditional CS1 course and to provide an opportunity for instructors to introduce the fundamental computer science concepts in fun and creative ways. During the workshop the participants will build apps of various levels of complexity ranging from Hello World-like apps that can be used in the first week to simple board games that incorporate advanced programming constructs at the end of the semester. No knowledge of Android and event-driven programming is required to use the library, yet students are able to create apps that run on mobile devices using only CS1 constructs. The intended audience is college and high school instructors teaching CS1. Support code, documentation, and examples are at:

http://www.cs.gettysburg.edu/~ilinkin/cs1android .

CCS CONCEPTS

• **Social and professional topics → Professional topics → Computing education → Computing education programs → Computer science education** → CS1

KEYWORDS

CS1, mobile devices, objects-first, objects-late

1 BACKGROUND / RELATED WORK

This project aims to contribute to the growing body of computer science education research by investigating opportunities to enhance instruction at the CS1 level using mobile devices to create an engaging learning environment. The opportunities for creativity and open-ended projects afforded by mobile devices have the potential to stimulate interest from underrepresented groups, including women, who are active users of the technology and can see the immediate relevance of their CS1 experience by developing applications for their devices.

The popularity of Java opens the possibility to augment instruction in the CS curriculum through the development of apps that run on Android devices. Recent work includes papers, posters, and workshops on using App Inventor and Touch Develop in CS0 [1, 2], teaching software design through Android projects in a third-semester course [3], and integrating mobile devices in junior/senior level courses [4, 5]. Discussion of related work from previous years is included in a paper that reported preliminary results on the ideas proposed for this workshop [6].

2 CONTRIBUTIONS

There is a notable gap in the published work on successful integration of mobile devices in CS1. This is understandable, since approaches that work at the CS0 level using App Inventor aim for a gentler, high-level introduction to programming with a focus on app design. However, the visual programming environment of App Inventor is no longer a strength in CS1 where one of the goals is to introduce the students to the details of syntax and structure of a programming language. It is easy to see the potential for using Android in CS2 without significant disruption to the course structure, since instructors can assume a certain level of proficiency. Unfortunately, introducing mobile devices in CS1 without appropriate scaffolding is unrealistic due to the complexity of the programming model and the required background.

This project aims to make progress in closing this gap by introducing an Android library that is designed to fit seamlessly in a traditional CS1 course and to provide an opportunity for instructors to introduce the fundamental computer science concepts in fun and creative ways.

The library is designed to be used in both objects-first and objects-late courses. No knowledge of Android, event-driven, and object-oriented programming (classes, inheritance, polymorphism) is required to use the library, yet, it offers a sufficiently rich interface that makes it possible to develop engaging interactive games in CS1. The approach differs from using App Inventor, since the goal is to support a traditional introductory course using Java by offering a framework that can enhance the appeal of the programming assignments.

3 PROGRAMMING PROJECTS

Figure 1 shows screenshots from student submissions for programming projects we have used in the past.

MyPetApp is our first assignment and its intent is to familiarize the students with Java and the library (Fig. 1, top-left). The students write a sequence of `drawShape` statements to draw a figure, which illustrates the process of transferring a design on graph paper into Java code.

ACE'20, February 3 – 7, 2020, Melbourne, VIC, Australia
© 2020 Copyright held by the owner/author(s). 978-1-4503-7686-0/20/02.
https://doi.org/10.1145/3373165.3373189

The *NimApp* is used to introduce methods, control structures, and boolean expressions (Fig. 1, center). The students build an app in which two players take turns removing coins from a pile. The *NimApp* offers a number of pedagogical possibilities. For example, students could add an extra pile, write a two-player version, or implement a rudimentary computer player.

The *Hangman* and *MineSweeper* games are popular CS1 assignments (Fig. 1, bottom and top-right). They require no lengthy introduction and provide an opportunity to exercise in a fun and engaging way a number of fundamental computer science concepts (e.g. methods, control structures, and linear structures/strings, 2D array processing and nested loops). Creating *Hangman* and *MineSweeper* apps provides a sense of accomplishment for the students, who are just starting to learn programming; it is all the more rewarding to be able to play them on a personal mobile device and share them with friends.

4 OUTCOMES

The goal of the workshop is to introduce this work to a wider audience, share our experience using the library in CS1, and forge collaborations for the continued development and assessment of the project.

The workshop will lead the participants through a series of activities that demonstrate the features of the library and conclude with a brainstorming session and open call for contributions of lab examples and assignments that will be shared through the project webpage.

The workshop could lead to collaborations on the following aspects of the project:

- API refinements and feature extensions
- support for iOS to allow Java apps written by students to run on both iOS and Android devices (currently in prototype form through Multi-OS Engine [7])
- multi-institutional pilots for gathering feedback and assessment data

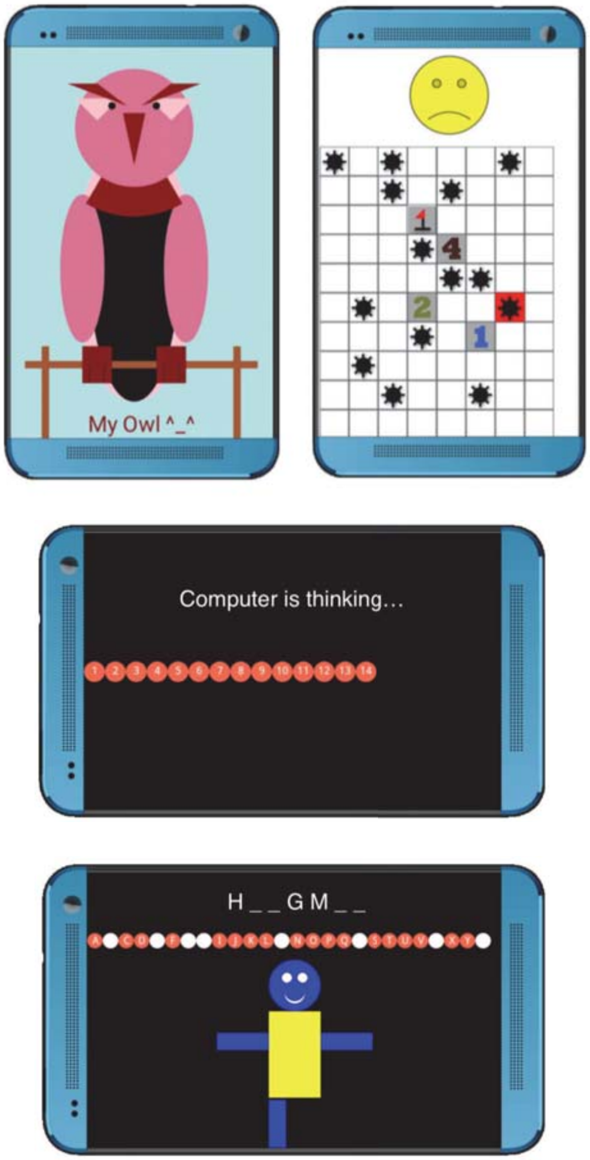

Figure 1. Screenshots of student submissions

REFERENCES

[1] Morelli, R., Uche, C., Lake, P., and Baldwin, L. 2015. Analyzing Year One of a CS Principles PD Project. *In Proceedings of the 46th ACM Technical Symposium on Computer Science Education (SIGCSE '15).* ACM, New York, NY, USA, 368-373. DOI: https://doi.org/10.1145/2676723.2677265.

[2] Catete, V., Peddycord III, B., and Barnes, T. 2015. Augmenting introductory Computer Science Classes with GameMaker and Mobile Apps (Abstract Only). *In Proceedings of the 46th ACM Technical Symposium on Computer Science Education (SIGCSE '15).* ACM, New York, NY, USA, 709-709. DOI: https://doi.org/10.1145/2676723.2678307.

[3] Campbell, J. and Tafliovich, A. 2015. An Experience Report: Using Mobile Development To Teach Software Design. *In Proceedings of the 46th ACM Technical Symposium on Computer Science Education (SIGCSE '15).* ACM, New York, NY, USA, 506-511. DOI: https://doi.org/10.1145/2676723.2677307

[4] Yuan, X., Bryant, K. S., Williams, K., and Xu, J. 2015. Integrating Mobile Computing and Security into a Computer Science Curriculum (Abstract Only). *In Proceedings of the 46th ACM Technical Symposium on Computer Science Education (SIGCSE '15).* ACM, New York, NY, USA, 686-686. DOI: https://doi.org/10.1145/2676723.2691902.

[5] Niu, S., Esakia, A., and McCrickard, S. 2015. Exploring Computer Science Topics with Programmable Smartwatches. *In Proceedings of the 46th ACM Technical Symposium on Computer Science Education (SIGCSE '15).* ACM, New York, NY, USA, 440-440. DOI: https://doi.org/10.1145/2676723.2691874.

[6] Ilinkin, I. 2014. Opportunities for android projects in a CS1 course. *In Proceedings of the 45th ACM technical symposium on Computer science education (SIGCSE '14).* ACM, New York, NY, USA, 615-620. DOI: https://doi.org/10.1145/2538862.2538983.

[7] Multi-OS Engine: https://multi-os-engine.org.