

# Three Approaches to Developing Educational Apps for Mobile Devices

Dr Matt Melchert

CPIT

130 Madras St

Central City, Christchurch 8011

(64) 022-050-3671

melchertm@cpit.ac.nz

## ABSTRACT

This paper describes three approaches to developing tutorial applications for mobile devices. Each of them has advantages and disadvantages regarding ease of development, flexibility, and feasibility.

## Categories and Subject Descriptors

Educational technology

## Keywords

Flexible delivery, mobile devices.

## 1. INTRODUCTION

Mobile devices, i.e. smartphones and tablets, are becoming ubiquitous among students. More and more, their lives revolve around their mobile devices for communication, organisation, entertainment, and the like. It's a part of their "Digital Native" nature [7], and increasingly a promising avenue for education.

We hope to tap into this rich vein by providing teaching resources which are accessible on students' mobile devices and fun to use. The aim of this ongoing project is to allow teachers to easily create and distribute tutorial apps for study and review of the subject material. The desired features are:

- Ease of development for teachers
- Ease of distribution to students
- Flexible range of question types, e.g. true/false, multi-choice, drag-and-drop, and sorting.
- Ease of creating tutorials for multiple platforms, particularly iOS and Android.

Over the course of two years, a sequence of three types of 3<sup>rd</sup>-year capstone projects for the Bachelor of Information Technology was undertaken at Wintec. Each of these took a different approach:

- Custom-made apps written from scratch. Two concurrent projects were involved: one for iPhone and one for Android [1, 2, 3].
- A web app deployed on a server [4].
- A tutorial app generator which ran on a PC [5].

Each of these approaches had its advantages and disadvantages.

## 2. THREE APPROACHES

### 2.1 Apps written from scratch

This is the most obvious approach, and also the most pedestrian.

Two concurrent projects were undertaken, one for iOS using Apple's XCode® development environment [2] and the other for Android using the Android SDK® and ADK® environments [3].

#### 2.1.1 The iOS app [2]

Two students worked together as a team to develop this app. It included a set of questions relating to the material covered in the course NCC2797, *Principles of Computer Networks*. Question types included true/false, multiple choice, sorting, and drag-and-drop according to categories. There was also an attractive help screen done in a classroom blackboard style.

This app was tested on a group of volunteers from the 2797 class, who were very enthusiastic. They said they would definitely use such a tool if it were available [1, 2].

#### 2.1.2 The Android app [3]

One student worked alone on this project. Because fewer man-hours were available, the app perforce didn't have the same polish as the iOS app, but a similar functionality was produced [1]. In addition, an exercise where students were required to align microwave dishes using touch and slide was included which was not implemented in the iOS version.

#### 2.1.3 Custom coding conclusion

Although the results were very appealing and gave great promise, the skills and amount of work required to generate these apps are well beyond what the average teacher could be expected to handle. This could be eased somewhat by producing a template, but would still require a lot of programming skills.

### 2.2 The Web app [4]

#### 2.2.1 The promise

Another approach was to implement the tutorial app on a web server. This would have several advantages over a custom-coded app:

- The tutorial could be deployed on a range of devices, rather than for just one.
- The issue of distributing the app would be solved by using a web browser on the device.

#### 2.2.2 The reality

HTML 5 was expected to be used to implement the tutorial over a range of devices. Unfortunately, at the time HTML 5 was not sufficiently mature to be able to implement the more sophisticated features such as drag-and-drop for more than one type of device at a time. This rather defeated the purpose of using a Web app in the

---

This poster paper appeared at the 3<sup>rd</sup> annual conference of Computing and Information Technology Research and Education New Zealand (CITREZN2012) incorporating the 25<sup>th</sup> Annual Conference of the National Advisory Committee on Computing Qualifications, Christchurch, New Zealand, October 8-10, 2012. Mike Lopez and Michael Verhaart, (Eds).

first place. As a result, the project was only a very qualified success [6].

### 2.2.3 *Web app conclusion*

At this time, either the technology is not sufficiently mature to provide the desired functionality across a range of devices, or the types of questions need to be limited to, say, true/false and multiple choice in order for this approach to be feasible.

## 2.3 The tutorial app generator

Another approach altogether was to write a program that would run on a desktop or laptop that would allow a teacher to define a set of questions and then generate the mobile app automatically.

### 2.3.1 *Implementation*

Android was selected as the platform of choice because of the student's familiarity with Java and the relative lack of bureaucratic strings required for development as compared to iOS and Windows Phone. The program was written in Java.

The teacher could create question sets for different courses and topics within the course, save them, and edit them later if necessary. When the tutorial was ready, the teacher selected a menu item and an .apk (Android app) file would be "automagically" generated. This could then be deployed onto an Android device via USB, installed, and run.

### 2.3.2 *App generator conclusion*

This approach holds tremendous promise, as it eliminates the need for the teacher to deal with the intricacies of creating an Android app, which is a fantastically fiddly and complicated business.

Unfortunately, setting up the application itself is still a very complicated business, requiring that the Java JDK, Android SDK, and Ant<sup>1</sup> are all installed correctly and in the proper directories, with proper file names, etc. This is still well beyond most teachers' capabilities.

## 3. CONCLUSION

Each of the three approaches had its own problems, and (potentially at least) its own advantages:

- Developing from scratch produces a good result (depending on the skills of the programmer), but requires a large skill set and lots of time, which most teachers lack. Also, it produces an app for a single platform. This could be alleviated somewhat by producing tutorial app template projects.
- A Web app promises platform independence and ease of distribution, but the tools available did not deliver the required functionality.
- The app generator is by far the most promising approach. However, further work needs to be done to make the application easy to install as well as easy to use, and presently it is also platform-dependent.

## 4. ACKNOWLEDGMENTS

Our thanks to Wintec and CPIT for allowing this joint-affiliated project to go forward.

NOTE: although the author is currently affiliated with CPIT, this poster describes work that was done at the Waikato Institute of Technology (Wintec).

---

<sup>1</sup> Ant is a program which allows you to easily build and maintain complicated Java applications, much like Unix's make.

## 5. REFERENCES

- [1] Melchert, M. 2011. A Custom-made Tutorial App for iPhone and Android. Poster paper. Proceedings, CITRENZ 2, Rotorua, 2011.
- [2] Taylor, S. and Sami, A. 2010. Student project report. School of Information Technology, Waikato Institute of Technology.
- [3] Mazzola, D. 2010. Student project report. School of Information Technology, Waikato Institute of Technology.
- [4] Stone, M. 2011. Student project report. School of Information Technology, Waikato Institute of Technology.
- [5] Slade, J. 2011. Student project report. School of Information Technology, Waikato Institute of Technology.
- [6] Potgeiter, C. and Stone, M. Successful RAD Prototyping on a Capstone Project. Poster paper. Proceedings, CITRENZ Rotorua, 2011.
- [7] Prensky, Marc. Digital Natives, Digital Immigrants. On the Horizon. MCB University Press, Vol. 9 No. 5, October 2002.