A Mobile Application Development Approach to Teaching Introductory Programming

Qusay H. Mahmoud and Pawel Popowicz
Centre for Mobile Education and Research, Department of CIS, University of Guelph, Canada

Abstract – Mobile devices such as smartphones are becoming widely used on university campuses, and as the shape of computing is evolving more into a mobile environment, the programmer of the future will need to be aware of special considerations that need to be taken into account when developing applications for mobile devices. These unique considerations will also assist the programmer to look at traditional application development on desktop platforms from a different perspective and apply some of the strategies in mobile application development to this area. This paper introduces a new approach for using mobile devices and mobile application development as a mechanism to teaching introductory programming to computer science, information technology, and computer engineering students. We will explore how the mobile device approach to teaching application development could help students to look at special considerations that must be taken into account when dealing with mobile devices while keeping them interested and excited by being on the forefront of technological changes. We provide sample applications that instructors could use as assignments to integrate into their courses.

Index terms – CS1, Mobile devices, Smartphone programming.

INTRODUCTION

There are many discussions and thoughts on how best to teach introductory programming to students in computer science courses. These discussions vary from introducing computer programming using a games first approach [7], to using special development environments [5], to using progressive graphics [2]. While there are inevitably benefits and drawbacks to any teaching approach, we feel that the use of mobile devices in teaching introductory programming could provide far more benefits to the student than other techniques as we discuss in this paper. There are examples available where the use of mobile devices to teach computer programming to students have been successful [8,9] along with different strategies for integrating mobile devices into the computer science curricula.

The emergence of smartphones such as the BlackBerry and the iPhone, which are capable of not only voice communication and simple text messaging communication but of processor-intensive activities such as multimedia playback, document editing, and audio/video streaming, are becoming a replacement for the laptop computer. These new powerful devices, with their new capabilities, have also required a new type of programmer to create mobile applications, one that is mindful of the special challenges a mobile device brings to the development environment. With billions of mobile subscribers worldwide there is a great need for programmers to have the qualifications needed to develop software in the mobile landscape. This is why this paper will push forward with the argument for using mobile devices and teaching mobile device oriented coding practices to introductory computer programming students. Not only will the use of mobile devices increase the interest of students by being on the forefront of technology, we will show that by using mobile devices in these courses a student will be able to transfer techniques used to create mobile applications to creating better and more efficient traditional desktop applications.

The objective of this paper is a smaller component in the broader vision of the Centre for Mobile Education and Research [1]: to create a multidisciplinary academic kit for integrating mobile devices into computing education. This paper will provide an outline of an introductory CS1 or CS2 course using mobile devices as the focus of application development. Ideas for assignments will be provided that will require a gradual progression of coding experience to complete culminating into a final finished product. We will also show how knowledge and experience of mobile application programming techniques can be transferred to a traditional application development environment and the benefits it will offer to the developer and a potential consumer. Information about the CMER Academic Kit for integrating mobile devices into Computing Education can be found at [1].

To this end, the rest of this paper is organized as follows. Section 2 discusses some of the challenges in mobile application development and how they can be used as opportunities for motivating students. Section 3 provides a comparison with other approaches for teaching computer programming such as robots and game development. Section 4 discusses the software tools that can be utilized in such a course. Section 5 presents the assignments and sample solutions that can be integrated in an introductory
practical solutions to making a “flexible” graphical user interface are provided in [6], such as paying particular attention to how menu items appear on the screen when using high-level APIs of the Java ME platform, and the use of low-level APIs will require spending more time testing the UI on various devices. This difference will allow students to see the pros and cons of both choices. In addition, screen contrast and its importance in the mobile environment on usability are discussed providing tips on using different contrasts effectively. Ensuring that the developer makes every pixel count such as making clear distinction between objects, or using different landscapes to display the information will help overcome some screen issues.

Students will also be able to see the effect of processing resources on the battery life of the mobile device. As this would be an introductory course, resolving processing and battery life issues may be out of scope for the course but would give students the insights into these issues.

Along with the technical challenges that will present themselves to students, providing them with a unique approach to traditional application development, the use of mobile devices may cause students to have a greater excitement for the course by using the newest and best technology available. This can help students keep focused on the content being taught.

**COMPARISON WITH OTHER APPROACHES**

There are also other approaches that have been proposed to teach introductory computer programming. In this section we will look at the differences between using mobile devices as opposed to methods such as using robotics or gaming to teach introductory computer programming.

**Robotics in Introductory Programming Courses**

Many approaches that have been proposed to teach computer programming are designed to keep the student excited about the topic by linking the subject with products that students find entertaining. Examples of this can be seen with robotics and gaming. Using robotics to teach introductory programming can definitely keep a student’s interest with the hardware side of the equation [11]. The ability to see one’s program to physically control and move a device can be very rewarding and exciting; however the usage of robotics has several drawbacks and fails to address some issues when teaching computer science to students. While using robots is fun and exciting, the hardware for establishing a course can be very costly. Outside of the start-up costs, when we look at the skills that using robots teach in programming we encounter some problems. The amount of functionality that can be constructed by students for use in robots can be quite limited. While students can create scenarios that make the robot move or perform tasks such as navigating a maze, these scenarios are only a small percentage of what is...
possible, students will not be able to create usable and functional applications when only programming for robots in mind. The choice of hardware and language may also hinder students, as robots can be programmed in a wide variety of languages, and the introductory language that allows students to quickly jump into robotics programming may only suite introductory courses and be unable to extend to other meaningful work.

Game Design in Introductory Programming Courses

Another approach to teaching introductory computer programming is through the use of game design [7]. Game design is an approach based on the excitement generated through games. Game development is a tricky area however, as students are not necessarily excited through creating simple games such as Tic-Tac-Toe while the more advanced games are out of scope in an introductory computer science course. Due to this, many approaches use a development environment to design games. Examples of such development environments include Game Maker [13] and Greenfoot [3]. These development environments allow students to move objects into a space and create movements and actions. While this type of approach can generate quick results for students, it can miss some of the key purposes in teaching programming concepts. Students should learn the fundamental principles of programming prior to taking on challenges of game design and graphics. The use of development environments hides many of the issues that a computer science student will need to overcome to become a competent programmer. Finally, we believe that in using gaming in CS1 or CS2 requires a very good understanding of various topics in Mathematics and Physics in addition to Computer Graphics.

LANGUAGE & SOFTWARE TOOLS

In any practical component of a computing course there are software tools needed to help exemplify the theoretical concepts learned in the classroom. This section will discuss the software tools utilized to aid in integrating mobile devices and mobile application development into an introductory programming course that we have found useful throughout our experience. The items that we outline are favorable because of their effectiveness and cost. The software tools include mobile platforms, programming languages, integrated development environments (IDEs), and simulation and emulation tools. We have also used physical mobile devices.

Mobile Platforms

Similar to how there are several platforms for desktop environments such as Windows, MacOSX, Linux/Unix, there are also many platforms for mobile devices. Some of these mobile platforms include Java ME, RIM’s BlackBerry, Windows Mobile, Palm OS, Symbian, iPhone, Android, and others. This heterogeneous mobile

environment makes computing education more challenging and introduces unique concepts for any programming course. We have used the Java ME platform and the BlackBerry smartphone and associated software tools [8,9] as explained later.

Java ME

In an introductory programming course we used Java and hence we have adopted the Java Micro Edition (Java ME) platform [8,9]. We have found that Java is an excellent language for developing mobile applications because it is a relatively easy to learn and also because it is well supported by devices and the Java community. The Java ME platform is a viable choice for mobile application development in CS education for a variety of reasons:

- It is based on Java, and it is simpler than Java SE. If your students are already familiar with Java they can start developing with Java ME within a few hours.
- Java ME is open-source, and it is the most ubiquitous application platform for mobile devices.
- It is widely supported and available on phones from Nokia, Motorola, Sony-Ericsson, and many others. It is also a popular platform for many portable devices such as handheld gaming systems and set-top boxes.
- There are many open-source tools available for developing mobile applications with Java ME.

Sun Java Wireless Toolkit for CLDC

The Sun Java Wireless Toolkit for CLDC or the Connected Limited Device Configuration [12] is an easy to use development environment designed to develop mobile applications based on Java ME CLDC and MIDP. We recommend using this toolkit as it provides an excellent environment as a ‘getting started’ tool for Java ME. It provides an emulation environment where students are able to see their applications running in an emulated mobile device environment. It also contains examples and provides tuning and optimization options for the developer.

Blackberry Java Development Environment

The BlackBerry Java Development Environment (JDE) is a development environment and simulation tool for building Java ME and BlackBerry based applications. It comes in the form of a standalone IDE and also a plug-in for the Eclipse IDE. The BlackBerry JDE allows for the usage of BlackBerry features such as camera and media, touch screen support, and accelerometer to name a few. The JDE comes with the necessary tools for converting existing Java ME MIDlets into a format (.cod) for the BlackBerry platform.
Blackberry Smartphone Simulator

The BlackBerry platform is built on Java ME which is why we utilize BlackBerry devices for development in our programming courses [8,9]. While physical devices are useful for testing GPS-enabled applications, the BlackBerry simulator is an excellent teaching tool for labs. It is an excellent tool for emulating the functionality of the BlackBerry device. The advantage of this tool is that you can use it to test both, basic Java ME applications as well as BlackBerry applications for a variety of BlackBerry smartphones and it can be downloaded separately or as part of a development environment such as the JDE. Students enjoy the simplicity of this tool and are excited to work with BlackBerry devices.

BlackBerry Devices

While running applications in an emulated or simulated environment provides a good framework for testing, we believe that allowing students to experiment with the physical devices themselves provides a totally different and better experience. In our case, we have provided the students with active BlackBerry devices to experiment with during the labs. In the labs, we have paired the students and provided each pair with a device. The students check out the labeled devices at the start of the lab and return them at the end of the lab. If students happen to have their own Java-enabled devices, we encourage them to use their devices. Other Java capable devices would also be suitable but would not necessarily work with some of the previously discussed software and tools.

SAMPLE ASSIGNMENTS

This section outlines the assignments that could be used in a computer science course taking a mobile device approach. The assignments are designed to build in difficulty and to allow a steady progression of skills in order to complete. Each assignment also focuses on specific areas of mobile application development to assist the students.

Four assignments are provided here to develop a student’s programming skills. The assignments call to develop a quiz, a calculator, an address book, and a text editor. The assignments are outlined here along with screenshots from sample solutions.

Mobile Quiz Application

Developing a quiz application would provide the first exposure to students in the graphical user interface components of a mobile application. It would expose them to commonly used elements such as labels, textfields, editfields, choicegroups, etc. They would also gain knowledge in working with user choices and results, and processing user input to provide scores to users. Screen navigation can also be shown to the students so they can learn how to pass information between screens in a mobile application. We have created a small quiz application which allows new students to gain a grasp of some user interface components and basic logic. The students could reuse some of the code and add additional features. Figure 1 shows a sample quiz.

```
1. Come up with a word for the following letters: D D E F N Q U
2. What is (10 + 20) * 37
3. What is the capital of Canada?
   - Toronto
   - Montreal
   - Vancouver
   - Ottawa
4. A tonne of bricks weighs more than a tonne of feathers.
   - True
   - False
```

The quiz was generated using editfields, radiobuttongroups, radiobuttons, gaugefields, and other user interface components. Once the user answers the questions they will submit the quiz. The answers are then checked and the user is taken to a results screen as shown in Figure 2. A progress bar shows the number of correct answers out of the total number of questions and informs the user of the questions they must review as they were answered incorrectly.

```
Four results:
3/4  75%
Question 4, must be reviewed.
```

Figure 1: Sample Quiz

Figure 2: Sample Results

Calculator

Developing a calculator application for a mobile device could provide a good learning experience. It would allow students to gain knowledge in the mobile application graphical user interface and allow them to work with data storage structures such as vectors. Figure 3 below shows a screen capture for the graphical user interface of a sample calculator application.

```
 Calculator

9   /  0  +/-
8   x  3  CE
7   *  2  C
4   +  1  BCKSP
5   -  6
6   =  7
```

Figure 3: Calculator running on BlackBerry 8330

Figure 3 shows the calculator application running on a BlackBerry 8330 simulator. To demonstrate issues
surrounding screen size, we also loaded the same application on a Blackberry 8220. The Blackberry 8220 also known as the Blackberry Pearl Flip has a smaller screen size compared to other Blackberry devices available. Loading and running the application showed that the BCKSP button found in the lower right hand corner could no longer fit on the button due to the size. This is shown in Figure 4.

![Calculator on 8220 without modification](image1)

Figure 4: Calculator on 8220 without modification

To overcome this issue a simple solution was implemented by retrieving the device name and adjusting the label on the button accordingly. The code displaying the solution is shown in Figure 5 while the final result is displayed in Figure 6.

```java
if(DeviceInfo.getDeviceName().equals("8330")){
    wbtnBCKSP = new FixedWidthButtonField(" BCKSP", size);
} else if(DeviceInfo.getDeviceName().equals("8220")) {
    wbtnBCKSP = new FixedWidthButtonField("<-", size);
}
```

![Figure 5: Modified code for different BlackBerries](image2)

The calculator that has been developed as a sample application allows the user to enter values in the editfield area at the top of the screen. This area only accepts real numeric input through a filter in the field. Users are allowed to perform basic operations such as division, multiplication, addition, and subtraction. Users can have negative values and decimal values as well. The calculator allows for multiple values to be stored and processed. The values are stored in a vector along with the operations that are required, once the user clicks on the “=” button, the values in the vector are removed and processed by the order that the user initially placed them. The user can also clear the entire content of the storage vector or the value currently in the editfield. Once the user has a result from a set of calculations they can also perform subsequent calculations without being required to enter the last value.

In our example, the creation of the graphical user interface was made possible through the use of a Field Manager called GridFieldManager [10] to allow for the spacing and positioning of the buttons on the screen. The GridFieldManager is available free of charge under the GNU Lesser General Public License (LGPL). To extend this application for students and make it more challenging we can ask them to create more functionality with respect to the mathematical operations that could be performed. Challenging students to create memory functions could increase the difficulty of this assignment.

**Address Book**

Students could also be asked to develop an address book application for a mobile device. The key goal of developing an address book would be to provide students with experience in screen navigation within a mobile device, developing skills with graphical user interface components and local data storage, as well as integration and interaction with device functionality.

**Text Editor**

A text editor is another application that students could be asked to develop. We have implemented a text editor to provide a working example. Figure 7 below shows the text editor listing text files from a SD card inserted into a mobile device.

![Figure 7: File List](image3)

The text editor allows a user to create a new text file or open an existing file. If the user selects an existing file it will open the contents of that file and display them to the user. The user can then proceed to edit the file and choose to save it under the same name, or under a different name. Figure 8 shows the text editor view when a file is opened by the user.
The text editor application challenges students to work with storage media such as SD cards. They will learn the operations of the FileConnection class to access files and directories on a mobile device. They will also learn how to deal with dialog inputs and managing the state of the application during that time. If a more challenging version is required, the students can be asked to provide functionality such as deleting files or navigating through folders.

CONCLUSION AND FUTURE WORK

The approach of introducing mobile application development at an early stage in the computer science curriculum can improve certain skills for computer science students. The areas of user input, display considerations, memory management, and battery life, are just some aspects that a traditional application developer will face on an infrequent basis, but a mobile application developer will face daily. A computer science student with experience in mobile application development can take the knowledge they have gained in mobile development and apply it to traditional development creating a more flexible programmer. In this paper we have presented four assignments and discussed how mobile application development can be integrated in introductory programming courses. The assignments proposed help teach specific concepts in mobile application development. The completion of the assignments will give students a good base to continue their education in both mobile and traditional application development. For future work, we plan to add more sample assignments for CS1 as well as other CS courses.

ACKNOWLEDGMENTS

The authors would like to thank the anonymous referees for their valuable comments in improving the presentation of this paper. The Centre for Mobile Education and Research is funded in part by Research In Motion (RIM).

REFERENCES