Learning to Program with Android

Ciarán Brien
B.A.(Mod.) Computer Science
Final Year Project  April 2011
Supervisor: Dr. Jeremy Jones

School of Computer Science and Statistics
O’Reilly Institute, Trinity College, Dublin 2, Ireland
DECLARATION

I hereby declare that this project is entirely my own work and that it has not been submitted as an exercise for a degree at this or any other university

______________________________________________        ________________________
Name            Date
Permission to Lend

I agree that the Library and other agents of the College may lend or copy this thesis upon request.
I would like to thank my project supervisor Dr. Jeremy Jones for his patience and understanding during this project.
Abstract

Concerns over high dropout rates and the quality of learning outcomes in Computer Science has led to much research into why this is, and into how topics such as programming can be better taught. A number of approaches and tools have been developed to aid teaching, such as Robots or Software Visualisation programs.

This project proposes to show that a smartphone can be used as a tool to make the teaching of programming more engaging and enjoyable for students.
Contents

1 Introduction ................................................. 4
  1.1 Motivation .............................................. 4
  1.2 Objectives ............................................. 5
  1.3 Report Layout ......................................... 5

2 Background ................................................. 6
  2.1 Issues in Computer Science Education .................. 6
  2.2 Approaches already taken ................................. 7
    2.2.1 Software Visualization ................................. 8
    2.2.2 Robotics ............................................. 9
  2.3 Smartphones ........................................... 10
  2.4 Android ............................................... 11
  2.5 Android SDK ........................................... 12

3 Design ..................................................... 13
  3.1 Target Audience ........................................ 13
  3.2 Phone features .......................................... 13
  3.3 Structure of an Android Program ......................... 14
  3.4 Topics to be covered in a course ......................... 15
  3.5 Lab influences ......................................... 16

4 Implementation ............................................ 17
  4.1 Labs .................................................. 17
    4.1.1 Lab 1 ............................................... 18
    4.1.2 Lab 2 ............................................... 19
    4.1.3 Lab 3 ............................................... 20
    4.1.4 Lab 4 ............................................... 21
    4.1.5 Lab 5 ............................................... 22
    4.1.6 Lab 6 ............................................... 23
    4.1.7 Lab 7 ............................................... 24
    4.1.8 Final Lab ........................................... 25
5 Conclusions
  5.1 Results ................................................. 26
  5.2 Learning Outcomes ................................. 27

A Accompanying CD ................................. 30
# List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>3rd level drop out rates Ireland</td>
<td>6</td>
</tr>
<tr>
<td>2.2</td>
<td>Alice</td>
<td>8</td>
</tr>
<tr>
<td>2.3</td>
<td>Alice</td>
<td>9</td>
</tr>
<tr>
<td>2.4</td>
<td>Smartphone picture</td>
<td>10</td>
</tr>
<tr>
<td>2.5</td>
<td>Android logo</td>
<td>11</td>
</tr>
<tr>
<td>2.6</td>
<td>Mobile Operating System Predictions</td>
<td>11</td>
</tr>
<tr>
<td>2.7</td>
<td>Android Mobile Device Emulator</td>
<td>12</td>
</tr>
<tr>
<td>3.1</td>
<td>Google Latitude</td>
<td>14</td>
</tr>
<tr>
<td>3.2</td>
<td>Google Latitude</td>
<td>16</td>
</tr>
<tr>
<td>4.1</td>
<td>Lab 1</td>
<td>18</td>
</tr>
<tr>
<td>4.2</td>
<td>Lab 2</td>
<td>19</td>
</tr>
<tr>
<td>4.3</td>
<td>Lab 3</td>
<td>20</td>
</tr>
<tr>
<td>4.4</td>
<td>Lab 4</td>
<td>21</td>
</tr>
<tr>
<td>4.5</td>
<td>Lab 5</td>
<td>22</td>
</tr>
<tr>
<td>4.6</td>
<td>Lab 6</td>
<td>23</td>
</tr>
<tr>
<td>4.7</td>
<td>Lab 7</td>
<td>24</td>
</tr>
<tr>
<td>4.8</td>
<td>Final lab</td>
<td>25</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

Concerns over high dropout rates and the quality of learning outcomes in Computer Science has led to much research into why this is, and into how topics such as programming can be better taught. A number of approaches and tools have been developed to aid teaching, such as Robots or Software Visualisation programs. This project proposes to show that a smartphone running Android can be used as a tool to make the teaching of programming more engaging and enjoyable for students.

A smartphone is a mobile phone that offers more advanced computing ability and connectivity than a contemporary feature phone. Smartphones and feature phones may be thought of as handheld computers integrated with a mobile telephone, but while most feature phones are able to run applications based on platforms such as Java ME, a smartphone allows the user to run and multitask applications that are native to the underlying hardware. Smartphones run complete operating system software providing a platform for application developers.[19]

Android is a software stack for mobile devices that includes an operating system, middleware and key applications. Applications can be developed using the Java programming language.[4] It is currently the bestselling mobile operating system.[18]

1.1 Motivation

The motivation behind this project was to create something useful with the knowledge acquired during the computer science course. As someone who had recently acquired a smartphone themselves, learning how to create ap-
1.2 Objectives

The objective of the project was to create a series of labs for a course in object orientated programming that uses an Android powered smart phone as a medium for teaching.

1.3 Report Layout

This report includes the following sections:

1. Introduction - This chapter describes what the project was about, the motivation for it, its objectives and the reports layout

2. Background - This chapter discusses the current state of teaching in Computer Science, its problems and its relevance to this project and also looks at previous attempts to teach programming through use of new mediums. It also looks at the state of the smartphone industry and the Android operating system.

3. Design - This chapter describes the decisions made before implementing the labs, and the intended goals that would influence them.

4. Implementation - This chapter goes through each lab that was created, and talks about what each lab would hope to achieve in terms of learning outcomes, and focuses on how a smartphone has been made use of.

5. Conclusions - This chapter looks at how successful the project was and what was learned during the project.
Chapter 2

Background

2.1 Issues in Computer Science Education

The past decade has seen a decline in the number of people taking technology based courses, and worse still, high dropout and failure rates among those who do choose to take them. In Ireland and the United Kingdom, Computer Science has the highest rate of any third level course.[16][17]

Figure 2.1: HEA Report on Dropout Rates
There are many reasons people fail to progress in third level education in general, social reasons being chief among them. Motivational problems, lack of preparedness and perceived difficulty are seen as problems that particularly affect Computer Science students. In many countries, including Ireland, there are no Computer Science courses offered at second level, so many students will enter the course with only vague ideas of what it actually entails.[12]

2.2 Approaches already taken

Traditional approaches to teaching programming involve a mix of lectures, reading, tutorials and labs. This usually involves a lecturer covering a particular topic, and that topic being put into practice in the tutorials or labs. Within the lecture hall students are normally passive recipients of the information. This, it is claimed, is not an effective method of teaching under certain circumstances. These include situations where memory retention is important, the tasks are complex and learner participation is important.[10] It has thus been argued that it is an unsuitable approach to teaching programming.[8] There have been many efforts to move away from this style of teaching, or at least to enhance it.
2.2.1 Software Visualization

Software visualization is the static or animated 2-D or 3-D visual representation of information and ideas represented in code and algorithms. Spatial intelligence is the ability to form a mental model of the world or pattern, and has shown to be a good indicator of programming success. [6] These visualizations can attract students’ attention and help clarify complex concepts. Program visualization focuses on the graphical representation of an executing program, rather than say, an algorithm. An example of this is Alice. Alice is a 3D interactive animation environment that provides a graphic visualization of a program’s state in an animated world. [7]

![Figure 2.2: Alice](image)

The benefits of this style of learning is that it engages students as they are able to quickly see animated results of their work. [14]
2.2.2 Robotics

Research has shown that learning prompted by interacting with ones environment is highly effective. [13] While simulated robot environments have existed for some time, it is only recently that technology for inexpensively supplying real robots to students has become available[8] LEGO MINDSTORMS is a line of programmable robotics, which are programmable in languages such as C. They originated from a project in MIT’s Media Lab.[20] Among those who have used them to teach programming are the US Military and the University of Canterbury.

![Figure 2.3: Alice](image)

The use of robots in courses has been shown in increase creativity, promote good design and planning and foster autonomous learning[8]
2.3 Smartphones

Mobile phones and computers can be found in most households in the developed world. Smartphones are an all in one, portable device that combines the functions of a mobile phone with the functions of a computer. The first phone to be regarded as a smartphone was the IBM Simon, released in 1992. It wasn’t until 2002, with the introduction of the Blackberry smartphone, that they began to get widespread usage, mainly among business customers. Apple’s iPhone was released in 2007, which opened up the smartphone market to the average mobile phone user. Smartphones with the Android operating system began appearing in 2008. A key feature of modern smartphones is the applications that are available for them. There are estimated 350,000 applications available on the Android market. Anyone is able to create and upload an application to the market, which is then available for download by market users. Most applications are available for free. Smartphone ownership is increasing rapidly. As of Q1 2011, almost half of all new phones sold in Europe were smartphones.

Figure 2.4: A picture of an android, blackberry and iphone
2.4 Android

The Android operating system is based on the Linux Kernel.[4] It was unveiled in 2007 by the Open Handset Alliance, a consortium of companies from the mobile operator, software, hardware, semiconductor and handset manufacturing sector. The Android Open Source Project is "tasked with the maintenance and further development" and led by Google.[2]

Currently, its main competitors in the mobile operating system market are Apple’s iOS, Microsoft’s Windows Phone 7, Research In Motion’s Blackberry OS and Nokia’s Symbian. Unlike iOS or the Blackberry OS, Android is not tied to any brand and can be used by any mobile manufacturer. As of the last Q4 2010, Android was the best selling mobile operating system in the world, with 33% of new smartphone sales using the platform.[18] It is predicted to have almost 50% market share of new sales by 2015.

| Worldwide Mobile Communications Device Open OS Sales to End Users by OS (Thousands of Units) |
|-----------------------------------------------|-------------|-------------|-------------|-------------|
| OS                                                | 2010       | 2011       | 2012       | 2013       |
| Symbian                                           | 111,577    | 89,930     | 32,006     | 661        |
| Market Share (%)                                 | 27.5       | 19.2       | 5.2        | 0.1        |
| Android                                           | 67,225     | 179,873    | 310,088    | 529,218    |
| Market Share (%)                                 | 22.7       | 26.5       | 49.2       | 48.8       |
| Research In Motion                                | 47,452     | 62,600     | 79,375     | 122,664    |
| Market Share (%)                                 | 16.0       | 13.4       | 12.6       | 11.1       |
| iOS                                               | 46,598     | 90,560     | 118,848    | 189,924    |
| Market Share (%)                                 | 15.7       | 19.4       | 18.9       | 17.2       |
| Microsoft                                         | 12,878     | 26,346     | 68,156     | 215,598    |
| Market Share (%)                                 | 4.2        | 5.6        | 16.8       | 19.5       |
| Other Operating Systems                           | 11,417     | 16,392     | 21,383     | 36,133     |
| Market Share (%)                                 | 3.8        | 3.9        | 3.4        | 3.3        |
| Total Market                                      | 206,647    | 701,630    | 476,1      | 104,808    |

Source: Gartner (April 2011)
2.5 Android SDK

The Android Software Development Kit (SDK) enables Android developers to create applications for the Android platform. It includes development tools, the Android libraries, debugger, documentation, sample code and tutorials. It also includes a virtual mobile device emulator that runs on your computer. The emulator lets you prototype, develop, and test Android applications without using a physical device. [3]

![Figure 2.7: Android Mobile Device Emulator](image)

Figure 2.7: Android Mobile Device Emulator
Chapter 3

Design

3.1 Target Audience

The course was originally designed with second year Engineering students in mind. At this stage they would have completed a programming course in first year. The idea was to use a more exciting environment that would appeal to the modern student. An environment that would appeal to students from a broad range of disciplines. Smartphones are such an environment. All computer engineering students from 2010/11 will be taught ARM assembly language programming, so this also ties in well with the smartphones, as most have ARM processors.

3.2 Phone features

The reason a smartphone was chosen as a medium with which to teach was because they are a new and exciting technology that is already widely used, and will be ubiquitous in the coming years. Learning how to create applications on a device many will use daily will be exciting and the thought of being able to do that after completing the course will be a great motivator. They lend themselves to programming well as it does not take long to create a program and demonstrate its usage on the phone. The idea was to utilise the features of the phone (e.g., touch screen, GPS, SMS) and be both interesting and achievable for relatively novice programmers. Developing a good set of programming/lab exercises is the key to delivering an interesting module that achieves its learning outcomes.
3.3 Structure of an Android Program

Android applications are composed of one or more application components (activities, services, content providers, and broadcast receivers). The activity is the most visible and prominent form of an Android application. An application may contain one or more activities. They are typically on a one-to-one relationship with the screens found in an application. A service is a component that runs in the background to perform long-running operations or to perform work for remote processes. A service does not provide a user interface. A content provider manages a shared set of application data. You can store the data in the file system, an SQLite database, on the web, or any other persistent storage location your application can access. Through the content provider, other applications can query or modify the data. A broadcast receiver is a component that responds to system-wide broadcast announcements. Many broadcasts originate from the system, for example, a broadcast announcing that the screen has turned off, the battery is low, or a picture was captured.

Three of the core components of an application—activities, services, and broadcast receivers—are activated through messages, called intents. Intent messaging is a facility for late run-time binding between components in the same or different applications. To inform the system which implicit intents they can handle, activities, services, and broadcast receivers can have one or more intent filters. The IntentFilter is either registered at runtime or is enumerated in the AndroidManifest.xml file.[1]

Figure 3.1: Google Latitude running on a smartphone
Every application must have an AndroidManifest.xml file. The manifest presents essential information, including

- The package name
- It describes the components of the application the activities, services, broadcast receivers, and content providers that the application is composed of
- It declares which permissions the application must have (e.g., Internet access)
- It lists the libraries that the application must be linked against. (e.g., Google maps library)

We will mostly be dealing with activities in the labs. An activity will be the main entry point of a program, somewhat similar to a main function. If a program has multiple activities, which one launches first is listed in the manifest file in an intent filter.

3.4 Topics to be covered in a course

The topics to be covered in a standard introductory course in object orientated programming include objects, classes, methods, data structures, sorting algorithms and class hierarchies. These all had to be incorporated into the labs that would be created.
3.5 Lab influences

The great thing about many of the most popular applications on smartphones at the moment is that they aren’t necessarily complicated pieces of software. With this in mind, it was decided that the labs would build up to a final lab that would resemble a fully fledged application. I settled on an application that would recreate the core functionality of the Google Latitude application. Google Latitude is a popular application that’s core function is to allow you to track the location of friends on a map using GPS. The decision to go with what would likely be familiar application was based on the belief that, if a student saw that they were able to create something themselves that they know is useful, it would create a great sense of satisfaction.

![Google Latitude running on a smartphone](image)

Figure 3.2: Google Latitude running on a smartphone

It was decided that the final lab would be created first, and that the preceding labs would be created from the final labs constituent parts. The preceding labs would therefore include labs with a focus on using GPS, overlaying images on google maps and use of a SQLite database.


Chapter 4

Implementation

This chapter goes through each lab that was created, and talks about what each lab would hope to achieve in terms of learning outcomes, and focuses on how a smartphone has been made use of. It starts off with a simple "Hello World!" type lab, and finishes with a lab that could be considered a full application.

4.1 Labs

There were 8 labs. Each overview will discuss the lab, and the intended learning outcomes.
4.1.1 Lab 1

The first lab is an introduction to using the Android SDK in Eclipse, and creating a ”Hello, Android!” program to run on the emulator. Detailed installation instructions are available at http://developer.android.com/sdk/installing.html

![Figure 4.1: Lab 1 output](image)

**Learning Outcomes**

- Become familiar with Eclipse
- Become familiar with Android SDK
- Learn how to make an Android application
- Learn about the various components that make up an Android application
4.1.2 Lab 2

A custom class is created and made textually displayable on the screen.

Figure 4.2: Lab 2 output

Learning Outcomes

- Create your first class
- Learn how to instantiate an object
- Override a method
- Display your own messages on screen
4.1.3 Lab 3

Create a currency converter class. It should have two methods, one to convert from dollars to euros and one to convert from euros to dollars.

Figure 4.3: Lab 3 output

Learning Outcomes

- Create a more complicated class
- Create methods for class
- Learn how to create an Android UI
- Learn how to program the functionality behind a UI
4.1.4 Lab 4

Make use of the phones GPS capabilities to show your GPS coordinates on the screen.

![Lab 4 output](image)

Figure 4.4: Lab 4 output

**Learning Outcomes**

- Become familiar with how GPS is used with the Android library
4.1.5 Lab 5

A lab that makes use of an overlay item and your GPS position to show where you are on Google Maps.

![Lab 5 output](image)

**Figure 4.5: Lab 5 output**

**Learning Outcomes**

- Create a subclass that extends a superclass
- Learn how to use Google Maps in an application
- Learn how to overlay your own pictures on a map
4.1.6 Lab 6
Implement the Quicksort algorithm, and display a sorted list of integers on screen

![Lab 6 output]

Figure 4.6: Lab 6 output

**Learning Outcomes**
- Implement a sorting algorithm
- Use it to sort a data structure
4.1.7 Lab 7

Implement a login screen that stores login details in a local SQLite database.

![Lab 7 output](image)

Figure 4.7: Lab 7 output

**Learning Outcomes**

- Learn how information is made persistent in an application
- Learn some SQL commands to interact with a database
4.1.8 Final Lab

The final lab was designed to bring together the knowledge gained throughout
the previous labs, and to show that a worth while application could be created
from what had been learned.

The application involves users being able to keep track of peoples locations
on a map, uploading your own location, and the ability to see the distance
between you and your friends. A 'Person' consists of a name, an email address
as a unique identifier, an age and a location. A 'Location' class which consists
of a longitude and latitude. There is a "SendToServer" class that consists
of a method to update a database with an id and location, and a method to
send a request to a server to retrieve all ids and locations currently stored
in the database. The user should be able to view, on a separate screen, the
distance they are away from others. The opening screen should allow a user
to login, and if it is their first time logging in, store their login details in a
SQLite database.

![Output showing friends location on map](image)

Figure 4.8: Output showing friends location on map
Chapter 5

Conclusions

This chapter looks at how successful the project was and what was learned during the project

5.1 Results

The main objective of creating a series of labs for an introductory course in object orientated programming, that utilizes a smartphone has been completed. How successful these labs would prove in practise is unclear, as no testing was completed on them. It is the authors belief that were students to successfully complete the labs, seeing a useful application at the end would come with a great sense of accomplishment. I suspect however that they may prove too difficult for people struggling to get to grips with a new programming paradigm.
5.2 Learning Outcomes

- Gained knowledge of what factors affect learning and motivation from research
- Learned about the state of teaching in computer science
- Learned about the smartphone market
- Android development was a new technology to the author, and the much experience was gained on the Android SDK and how applications work
- Future android application development would come much easier
- Java programming skills were improved as a result of application development
Bibliography


Appendix A

Accompanying CD

The source code for all the labs should be accompanying this report in a CD. Some php scripts for dealing with the server in the final lab are also included.