AC 2011-2106: SO YOU WANT TO TEACH AN IPHONE PROGRAMMING COURSE?

Kyle D. Lutes, Purdue University, West Lafayette

Kyle Lutes is an Associate Professor for the Department of Computer & Information Technology (CIT) at Purdue University. Kyle joined the department in 1998 and is the chair of the department's software development curriculum. His teaching and scholarly interests cover a broad range of software development areas including software applications for mobile devices, data-centered application development, and software entrepreneurialism. He has authored/co-authored numerous papers and two college textbooks on various software development-related topics. Prior to his current appointment at Purdue, Kyle worked for 16 years as a software engineer and developed systems for such industries as banking, telecommunications, publishing, healthcare, athletic recruiting, retail, and pharmaceutical sales.

Teresa A. Shanklin, Purdue University

Teresa A. Shanklin has a Bachelors degree in Computer Science and graduated from Iowa State University with a Masters Degree in Information Assurance. She is currently a Ph.D. candidate at Purdue University in the College of Technology, where she is a research assistant in the Machine-to-machine (M2M) lab. Her research interests lie in the areas of indoor positioning and path planning, mobile devices and multi-agent systems.

So You Want To Teach an iPhone Programming Course?

Abstract

According to a Pew Research survey conducted in April and May 2010, an estimated 82 percent of adult Americans now own a mobile phone and nearly 25% of United States adults use mobile apps on their phones. The Apple iPhone was introduced in 2007 and to date has been a cultural phenomenon in addition to being a commercial success. According to Apple's quarterly earnings, close to 60 million iPhone's have been sold through the end of June 2010. The success of the iPhone can at least partially be attributed to the iPhone ecosystem consisting of mobile device hardware, the iOS operating system, software developer tools, and the App Store – all created and controlled by Apple. To date, over 300,000 apps are available in the App Store, and Apple has reported that over 1 billion dollars in profit has been paid to iPhone developers.

The Department of Computer and Information Technology at Purdue University strives to keep its curriculum current and to teach courses using best-of-breed technologies. For this reason, an undergraduate, upper-level course on iPhone application development was offered during the Fall 2010 semester. Our department has been teaching software development for mobile devices since 2002, but the Fall 2010 semester was the first using Apple development tools for iOS devices including the iPhone. In this paper we will discuss our experiences teaching the course. Topics will include obstacles faced, the selection and purchase Macintosh computers for our mobile computing lab, selection and purchasing of mobile devices, course pedagogy, textbook selection, student assessment, and unexpected problems are presented. Finally, conclusions and lessons learned are addressed.

Background

By design, the curriculum in the Department of Computer and Information Technology (CIT) at Purdue University is focused on the application of Information Technology as opposed to theoretical computing. To this end, CIT's courses are focused on students learning through hands-on experience. This characteristic differentiates our courses from those of the Computer Science or Computer Engineering departments at Purdue. The typical student progression through the CIT curriculum includes a minimum background of three programming courses including an introductory course using C#, a web application development course using C#, and an intermediate object-oriented programming course using Java[1].

Our first course teaching software development for mobile devices, CIT 355, was introduced in the Fall 2002 semester. This course has proven to be popular and has been offered nearly every semester since. The typical CIT 355 student is a junior or senior majoring in CIT. Occasionally students from other related disciplines such as Computer Science, Electrical and Computer Engineering, and Computer Graphics also take the course. Typical enrollment figures are between 12 and 20 students per semester. The numbers are limited by the number of workstations and mobile devices in our mobile computing laboratory.

This first offering of CIT 355 had students programming applications for the Microsoft Pocket PC PDA platform as this platform was the clear market leader in the handheld device space at

the time. Students used the C# programming language, the Visual Studio IDE, and the .NET Compact Framework class libraries when developing applications. In addition to being a popular platform, this environment proved useful for pedagogical reasons as our students had used Visual Studio IDE and C# in prerequisite courses. CIT 355 evolved along with commercial mobile device technologies and has used Windows Mobile Smartphone devices exclusively for the past several years.

Our second course in the software development for mobile device sequence, CIT 425, was offered for the first time during the Spring 2009 semester. The justification for the development of this second course was that the topics of mobile device development are so broad, they cannot all be explored in one semester to the desired depth. Additionally, CIT 425 allowed us to introduce alternative smartphone platforms and development tools, which was especially important as the Windows Mobile platform was becoming a dated technology and diminishing interest from both consumers and the CIT students.

During the Spring 2009 semester CIT 425 was taught using the RIM Blackberry smartphone platform. For pedagogical reasons, the Blackberry was a good choice. We were loaned Blackberry smartphone devices from RIM, the software development tools are free and run on Windows workstations, and the development environment is based on the Java programming language, of which our students have had prior experience. Although the class was well received, common student complaints included: the development environment being regarded as too similar to Windows Mobile devices, yet more cumbersome to use; and students were not interested in the Blackberry platform as few students owned Blackberry devices because of its perception of a business enterprise smartphone platform.

A primary reason for teaching software development courses for mobile devices is because of student interest. When polled about which smartphone platform they were most interested in, CIT students overwhelming chose the Apple iPhone over RIM BlackBerry, Google Android, and Microsoft Windows Mobile. iPhone's are considered to be at the forefront of changing technology and a portion of the draw is the ability of any programmer to develop applications to be sold through the iPhone App Store. This is particularly appealing as there is a very low investment threshold and a shortened time line in developing relevant and available applications for these devices.

Perceived Obstacles

Even though the student demand for an iPhone programming course was high, the we CIT faculty were hesitant to teach such a course because of several perceived obstacles. In the case of iPhone development, both instructors and students would be required to learn several new technologies.

Developing for the iPhone requires using Macintosh workstations. All prior development for mobile devices had been done in a Windows environment, as was done in the prerequisite three programming courses. In fact, there was no Macintosh computing laboratory within our department. Macintosh workstations use the Macintosh Operating System (Mac OS). While many students, and a few faculty, own and use Mac computers, no prerequisite courses require

any working knowledge of Mac computers but most courses do require the use of applications running on Windows PCs.

Software development for iPhones also would require instructors and students to learn new development tools – the Xcode IDE, Interface Builder, new class libraries and SDKs, and the Objective-C programming language.

Finally, obtaining handheld device hardware would be problematic. For each semester our mobile courses have been offered, we have always provided a loaner hardware device for each student to use. While the platform vendors provide excellent simulators for testing and debugging, real experience, with all the trials that this includes, can only be found using actual devices. Real devices provide better experience using smartphone specific features such as GPS, cameras, accelerometers, multi-touch screens, and tiny QWERTY keyboards.

For example, one student had been developing her semester project for many weeks using a device simulator only. Since the simulator mapped the PC's physical keyboard to the mobile device, the student had never attempted to input text using the device's on-screen keyboard. Her application was data entry intensive, and she didn't realize how unusable her application was until the final project presentation when we required all students to demonstrate their projects on real devices.

Our initial batch of 15 Pocket PC PDAs were provided by Microsoft. Other funding sources, including corporate gift funds and department equipment funds, were used in later semesters for newer model PDAs and eventually for Windows Mobile smartphones. Unlocked Windows Mobile devices could legally be obtained which limited their price to that of hardware only because a service plan was not required. We knew obtaining iPhone hardware would be expensive as iPhone hardware was not easily obtained without a two-year contract from AT&T. A common rule of thumb when pricing a smartphone is to assume $2,000 for the two-year contract plus the cost of the iPhone hardware which means to equip our lab with just 12 iPhones could cost our department over $25,000.

Laboratory Setup

Mac Hardware

For our new iPhone course to be successful, our desire was to have the necessary equipment available for student use. This equipment included supplying Mac's for the new course and keeping the Windows OS available for the other courses. One of the options explored and ultimately implemented, was the use of a Mac computer with the Apple Boot Camp dual-boot software. This allowed the lab to function both as Mac or Windows lab but required only one physical computer.

The range in price of Mac Pro desktop computers using educational discounts during Summer 2010 was $2299 to $4549. With economy in mind, we explored the option of using Mac Mini's as the workstations of choice. The cost of a Mac Mini with 2.26 GHz, 2GB memory and a 160GB HD was $549.

One major concern with implementing the lab with the Mac Mini was the small form factor of the units. Due to previous experiences with thefts from the lab, and the potential for the disappearance of easily carried small items, it was important to implement anti-theft practices for the lab. Specialized anti-theft brackets are commercially available that allow Mac Mini to be bolted and padlock to a physical desk. Such solutions such as the MacCuff Mini[2], were deemed price prohibitive with an individual cost of $59 each. In the end, a total of $40 was spent on off-the-shelf parts from a local hardware store that allowed us to secure each Mac Mini to the back of each workstation's monitor stand.

We reused existing Dell keyboards and mice, and had considered reusing dated 17-inch Dell monitors, but in the end our frugality allowed the purchase of new monitors. The monitor of choice was the Dell 24-inch LCD for $329 each. This Mac Mini and single large monitor replaced our prior configuration of two 17-inch LCD monitors connected to a single Dell workstation.

Software Development Tools

In addition to the Mac OS software, the students need access to the iOS developer tools – namely the Xcode IDE, Interface Builder, the iOS class libraries, and the Objective-C programming language, which are all freely available. Instructors and students also need to register with the Apple developer program, which is easy because Apple offers a free Developer University Program.[3] This special category of developer program is free to qualifying institutions and allows students to work under a single Apple department account, with separate student accounts. The Developer University Program accounts are able to do everything the regular accounts can, except deploy applications to Apple's App Store.

Handheld Devices

Due to carrier restrictions and requirements, iPhones are not easily obtained for teaching purposes without a valid service contract. The possibility of a limited service contract was considered, but the cost was recurring and prohibitive. Providing carrier service for student use would also open the possibility of liability due to misconduct or misuse. For example, if a student uses the phone to make threatening or illegal phone calls; or the student exceeds the usage agreement and incurs a large bill under the department account. Consideration was given to requiring students to have their own devices. However, not all students use iPhones. Often students are still using a device on a parent's account. An additional obstacle existed with the fact that the iPhone was only available from AT&T.

Given the problems with using iPhone devices, the second-generation iPod Touch was an attractive second choice. Although the iPod Touch has feature limitations compared to the iPhone, the cost for the cheapest iPod Touch in summer of 2010 was reasonable at $200 for the 8GB model. The iPad tablet device broadens the horizons of device application due to the increased screen size. The cheapest iPad available in Summer 2010 was $500. This is the most limited model that has 16GB storage capacity and Wi-Fi, but lacks 3G access.

Funding

At the time equipment was purchased, the summer of 2010, the cost to reconfigure our Mobile Computing Laboratory was approximately $15,000. This included 12 Mac Mini workstations, 12 24-inch Dell LCD widescreen monitors, 12 iPod Touch devices, and 3 iPad tablet devices. Total cost could have been reduced by approximately $3,000 by using existing computer monitors. Necessary funding was realized through two sources: an unrestricted gift from ArcelorMittal; and some funds from our department's annual Supply and Expense (S&E) fund.

Fall 2010

The first run of our iPhone course was the completed at the end of the Fall 2010 semester. The student make-up for the semester included fourteen students, all CIT seniors. Although fourteen students were two more than our lab with the 12 Mac workstations could support, additional space was created for these two additional students because they were willing and able to provide personal Mac laptops and iOS devices.

All fourteen students in the Fall 2010 semester had completed at least four prior programming courses including an introductory course using C#, a web application development course using C#, an intermediate object-oriented course using Java, and the course first software development for mobile devices course using C# and Windows Mobile (CIT 355).

The planned pedagogy of the course was considered relative to what other universities are/were offering. The de facto standard that other universities seemed to use, or at least considered is to use, is the materials from the CS 193 from Stanford.[4] The Stanford course runs every semester on location at Stanford. It is also recorded and freely available through iTunes University for anyone interested in learning more about application development. The course offers lecture material and sample code, but no book is required. Official Apple documentation is the reference material utilized. This course is designed as an undergraduate computer science course and requires object-oriented programming as a prerequisite.

Several other universities offering some version of an iPhone programming course include Madison Area Technical College[6] and the University of Maryland[7]. The Madison Area Technical College offers a two semester series of iPhone app development. The first semester introduces Objective-C, the SDK and basic development. The second semester covers more advanced topics such as core animation, core data, etc. The course is offered in the Information Technology department. The University of Maryland has offered an iPhone course a number of times as a senior level course. The prerequisites are knowledge of C, object-oriented programming as well as a class on data structures and algorithms.

Moving on to technical and extension options, there are offerings at the University of California Berkeley Extension[8] and the New Jersey Institute of Technology[9]. The offering by the UC

Berkeley extension has a ten-meeting course planned for basic iPhone development. This course is open to anyone who would like to enroll (and pay.) The website lists *Beginning iPhone 3 Development: Exploring the iPhone SDK* as the required text. The instructor is a working software engineer and has also taught programming courses at UC Berkeley extension for the last 10 years.

Available as a non-credit course to anyone online for the cost of the class, the New Jersey Institute of Technology Adult Learners course is listed as a continuing professional education course. The course runs for eight weeks and will use *Beginning iPhone 3 Development: Exploring the iPhone SDK* as the book of choice. The instructor is not listed.

Other universities that have offered some version of an iPhone application development course in the past, but are not showing a current listing include: Columbia University[5], Amherst[10], UNM[11], and MIT[12].

Given CIT's focus on application development, the book *iPhone for Programmers: An App-Driven Approach* by Deitel and Deitel[13] was selected. Another factor in selecting this book was the cost to the students. An online version is available to students at no cost via our universities subscription to Safari books online. A final assumed benefit of this book is that each chapter within the book is presented in the form of a complete and practical application.

The initial plan was the students would complete a chapter each week. After reading the chapter, they would complete the program as presented in the book. Next, a homework assignment would be given based on the book examples, with specific changes that would force them to understand and enhance the code presented in the book. The weekly lecture topics would discuss each new topic that was covered in the chapter to reinforce and clarify the objective.

Additionally, the students would be given a list of extra reading on related topics that were available from free online sources. This would supplement and also provide a deeper level of information than the book chapters.

The course was graded based on the following criteria: quizzes; midterm and final exams; and an end of semester team project. The quizzes and exams were comprised of multiple choice and short answer questions; the quizzes being lecture topical and the exams being comprehensive and longer. The end of semester team project allowed the students to pursue a topic they were interested in and explore it in more detail with the result being a working application. This strategy is identical to ones employed in most other departmental programming courses.

Suggestions for Improvement

There are a variety of lessons learned from this initial offering as well as suggestions for improvement offered by both students and the course instructors. Monetarily, it would have been a cost savings of approximately $2,000 to require students enrolled in the course own or have access to an iOS device. In the initial semester, at least five students had a personal iPhone and one student had an iPad, resulting in many devices remaining unused and locked in a cabinet all semester long. Additionally, the low-end iPod Touch can be purchased for little more than the cost of some textbooks. Requiring students to supply their own iOS device has the additional benefit of allowing the number of students to scale larger since the lab computers can be shared.

Another important point to consider is the possibility of requiring students to have previous experience with the C programming language. Objective-C, the language required for iOS development, is based on C. While all students had at least three prior programming courses, all were based on higher-level programming languages such as C# and Java. These high-level programming languages abstract the developer from low-level programming concerns such as data structures and manual memory management which are heavily used in Objective-C. Prior exposure and understanding of the C syntax and memory management would lessen the steep Objective-C learning curve posed to our students. The book we used for the course covered only iOS application development topics and not Objective-C. Without the background knowledge of Objective-C, the complexity of each programming topic was greatly increased.

The next time the course is offered, the homework assignments will not follow our initial strategy of adding features to the examples from the book. Students suggested it would be helpful to have frequent smaller applications built from the ground up, to practice and demonstrate essential topics. The book offered a variety of sample applications, however the authors of the source code found in the book examples followed different styles and patterns. These variances in source code caused further confusion from chapter to chapter.

Another deficiency with the book, from our vantage point, is the source code is too lengthy. Most examples were too complex and required knowledge of a variety of topics. The narrative of the book included what the code did, but not why it was coded the way it was, so it proved difficult for the students to follow all the intricacies of the topic. From previous teaching experience, it is often easier for students to comprehend new topics if they can be presented in small, modular and digestible quantities.

One final critique of us using book examples as the starting point for homework assignments, is that the source code for the book examples was readily available from the book author’s web site. This allowed students to “copy and paste” code (even though they were warned not to) rather than type it on their own, which would have provided a better learning experience.

In-line with some of the difficulties encountered with the course, our department has modified its curriculum to include an optional Systems Programming course (CIT 315),which will use both Unix and the C programming language. There is consideration to making this new System Programming course a prerequisite to any iPhone programming course.

Another option being considered is to split the iPhone course into a two-course sequence. The sheer volume and complexity of the topic almost make it a necessity to be a two-course sequence. The first course would be an introduction to Objective-C as well as exposure to the different software development tools available. The second course would focus more on advanced topics in iOS development, including using features found only on smartphone devices (for example, accelerometers, GPS, and cameras) as well as a unit on smartphone app commercialization.

Conclusion

After overcoming initial obstacles, perceived or actual, our department offered an iPhone app development course during the Fall 2010 semester. For a first run course, our iPhone course proved popular and enjoyable to both students and the instructors. The course will be offered again in the Fall 2011 semester.

This next offering, however, will see at least two changes to the past deliver method. First, students will be encouraged to use their own iOS device. If a student is not able to supply his or her own device, he or she will be allowed to check out an iPod Touch or iPad from our existing stock for a short period for testing. Second, the course lecture topics and homework assignments will be based on fundamental Objective-C programming topics rather than the emphasizing creating and modifying existing complete iPhone apps.

Assuming the Apple iPhone continues to be popular, and students continue to be interested in learning how to develop apps for these smartphones, other institutions should consider offering such a course, but do need to be aware of the Macintosh developer workstation requirement and the steep learning curve associated with the iOS software development tools.

Bibliography

1. Harriger, A., Lutes, K., Purdum, J. (2007). Designing Curricula to Teach Concepts and Increase Employability. In proceedings of American Society for Engineering Education, 2007.
2. Sonnet Technologies, Inc. Retrieved January 1, 2011 from website: http://www.sonnettech.com/product/maccuffmini.html.
3. Apple Inc. iOS Developer University Program. Retrieved January 1, 2011, from website: http://developer.apple.com/programs/ios/university.
4. Stanford University. CS 193 iPhone Application Development. Retrieved January 5, 2011, from website: http:// cs193p.stanford.edu.
5. Columbia University. Mobile Computing with iPhone and Android. Retrieved January 2, 2011, from website: http://www.cs.columbia.edu/~nieh/teaching/e6998/.

6. Madison Area Technical College. IT – iPhone Applications Development Certificate. Retrieved January 4, 2011, from website: http://matcmadison.edu/program-info/it-iphone-applications-development-certificate.
7. University of Maryland. CMSC 498I. Retrieved December 15, 2010, from website: http://www.cs.umd.edu/class/fall2010/CMSC498I/CMSC498I/Home.html.
8. UC Berkeley Extension. Developing Applications for the iPhone. Retrieved January 3, 2011, from website: http://extension.berkeley.edu/cat/course2215.html.
9. New Jersey Institute of Technology. Online iPhone Application Development Course. Retrieved January 4, 2011, from website: http://adultlearner.njit.edu/admissions/prospective/iphone_online.php.
10. Amherst University. iPhone Application Development. Retrieved January 3, 2011, from website: https://www.amherst.edu/offices/it/teaching_research/projects/intertermcourses/2009/iphone_app_dev.
11. University of New Mexico. iPhone Application Development. Retrieved July 5, 2010, from website: http://www.ece.unm.edu/course/ece495.
12. Massachusetts Institute of Technology. Introduction to iPhone Application Development. Retrieved July 4, 2010, from website: http://courses.csail.mit.edu/iphonedev/.
13. Deitel, P., Deitel, H., Deitel, A., Kern, E., and Morgano, M. 2009. iPhone for Programmers: An App-Driven Approach. Prentice Hall Press, Upper Saddle River, NJ, USA.

Move The Turtle: an educational app for iPhone and iPad that teaches your kid the basics of computer programming using a visual interface. Lightbot: a puzzle game that requires programming logic to solve levels. It helps your child understand procedures, loops, and other basic coding concepts. More like this âž¡ï¸ here. Tynker: another app that teaches you coding concepts as you solve puzzles. It supports Sphero, Ollie, drones, and smart lights.Â It has 45 levels to teach programming logic. Hopscotch: lets you make your own games and publish them, so others can play them. It provides you with simple videos to teach you how to make your own videos. Have you found better apps to learn programming? Please share them here. Currently trending posts