

Developing iPhone apps for language learning

Paul Daniels

Kochi University of Technology
daniels@kochi-tech.ac.jp

Several years back in a column about mobile learning, I discussed how mobile devices could become more ubiquitous in educational environments if they had the ability to connect to wireless networks via Wi-Fi.

As combination IP/cellular phones become more widespread, additional makers will be offering wireless handsets. Most importantly, students will be able to use their IP-based mobile phones to connect to wireless campus networks, thus allowing them to download large amounts of data at high speeds while circumventing the high cellular data transmission charges. (Daniels, 2006)

In January 2007, Steve Jobs unveiled the iPhone and in January 2010 the iPad; both devices allow consumers to shift between cellular networks and wireless Wi-Fi hot-spots. While Apple has succeeded in packaging Wi-Fi with the iPhone, other handset makers are slow to follow suit. Perhaps one reason mobile phone companies are slow to adopt Wi-Fi is that it cuts into company profits since Wi-Fi can be used to circumvent more expensive cellular network access. Because of the success of the iPhone, mobile phone companies are now waking up to the idea of supporting mobile phones that include Wi-Fi. Google has introduced the "Nexus One", which is manufactured by HTC and sold by major mobile providers around the world. Microsoft has also quietly developed "Windows Phone 7." The Apple, Google and Microsoft Smartphone rivalry is bound to yield more sophisticated and powerful Smartphones.

In addition to introducing the first successful cellular Wi-Fi combo Smartphone, **143**

Apple has succeeded in creating a media machine; the iPhone includes a remarkable 3.5 inch color screen, stunning multimedia capabilities and a full-fledged web browser. Another powerful feature of the iPhone is the software development kit or **SDK**, which allows developers to create sophisticated applications (apps) for the iPhone, iPad or iPod Touch. The iPhone **SDK** was released to third-party developers in February 2008 and since then hundreds of thousands of iPhone apps have been developed and billions of applications have been downloaded from the Apple App store.

As the number of iPhone and iPad applications available via Apple's App site continues to swell, more and more educational applications are emerging. When visiting the App Store, you can browse a list of the "top paid apps", which are the applications which have made the most money. The majority of these applications cost between **USD\$1** and **USD\$5**. Interestingly enough, some of the top selling applications include a basic Spanish course, a Japanese tutor application, an astronomy guide, a flashcard tool, a graphing calculator and a number of learning apps for children. After testing a few flash card learning applications with my two-year-old son, I was determined to design an improved application which was both flexible, as far as adding/editing content, and adequate for use with learners at different proficiency levels.

Designing verses developing applications

While I have some programming experience, I know enough about programming to realize that fairly advanced coding skills are required to develop successful iPhone applications. iPhone apps are developed using Xcode, which is the same set of tools used to develop applications for Mac **OS X**. Xcode supports a number of popular programming languages such as C, C++, Fortran, Objective-C, Objective-C++, Java, AppleScript, Python and Ruby. While many of us may not have the required skills to program in these languages, educational professionals can be involved in both the application and content design. Below are suggested ways to become involved in the design and development of iPhone applications.

Teaming up with a programmer

As an educator, you can choose to design the layout and content of an application and team up with an experienced programmer to code the actual iPhone app. With the abundance of freelance programmers around the world, connecting with a talented and reasonably priced iPhone app programmer is fairly simple. Locating programmers typically entails posting a project description to a site such as <www.getafreelancer.com>, <www.Elance.com>, <www.project4hire.com>, <www.rentacoder.com>, or <www.scriptlance.com>. While there are a number of legitimate programmers, care must be taken to ensure the developer is able to complete the anticipated project. When working with programmers online, remember that lowest bidders are typically not the best choice and good communication is of utmost importance. After several unsuccessful attempts, I was able to finally locate a reliable programmer whom I have been working with for the last five years. Once a reliable connection is made, the design process begins. The best method to communicate design features is to create simple screen shots, diagrams and images using a word processor or an html editor. Adding comments to images is also a useful feature in Microsoft Word, which can help illustrate both layout and functions of your proposed application.

account at the iPhone Developer Program website at <developer.apple.com/iphone/index.action>. After creating an account, the iPhone Software Development Kit or **SDK** can be downloaded and installed. A newer Intel **OS** is required along with Apple **OS** 10.5.7 or later to run the Apple **SDK**. Your new iPhone can be tested using the Apple's **SDK** iPhone simulator (include in the **SDK** package) or by enrolling in the free iPhone Developer University Program that allows development and testing of iPhone applications within an institution. The iPhone Developer University Program is intended for development of apps but not for institutional distribution. Applications are typically distributed via the iTunes store, which requires enrollment in the iPhone Developer Standard Program at a cost of **USD\$99** per year. Note that even free applications need to be approved by Apple and need to be distributed via the Apps Store.

Developing iPhone application templates

Similar to Adobe Flash, iPhone applications can be developed as templates to enable embedding of content such as text, images, audio and video from sources outside the application. Using a template approach it is possible to separate the content from the application. A template-type iPhone allows teachers with no programming experience to insert custom content into an iPhone app. The custom learning content is typically stored as **XML** or Extensible Markup Language files and uploaded to a website. The **XML** code can be generated using a web-based html form accessible to teachers and content developers. Web-based forms are used to separate input fields for each instance of content, such as *text_1*, *image_1*, or *audio_1*. The actual **XML** files can be generated on the fly without any knowledge of **XML** coding. A sample **HTML** form which is used to generate **XML** code to be embedded into an iPhone app is shown in Figure 1.

The screenshot shows a web browser window titled "iPhone xml tools - Windows Internet Explorer". The address bar shows "http://www.kochi-tech...". The form contains the following fields and controls:

- Title:
- Level1:
- Level2:
- Image (png 320x250):
- Audio (mp3):
- Video (m4v):
- Add new Frame
- Number: 1
- Image (png 320x250):
- Text:
- Vocabulary:
- Translation:
- Timing:
- Add new Frame
-

Figure 1. Sample HTML form for generating XML code 145

The **XML** code and content is loaded into the iPhone application via a network connection. The example shown in Figure 2 is a reading application that displays text and audio along with images or video within a “frame.” Once the content is loaded into the iPhone application, the navigational buttons appear at the top of the screen, the images appear in the center, and the text, vocabulary and translations appear beneath the image.



Figure 2. Reading application for displaying text, audio, images, video.

Template-based applications for the iPhone allow instructors and learners to embed custom content and activities without manipulating the actual iPhone application code. Using template-driven iPhone applications, it is easy to continually update course content without worrying about making changes to the application itself. The latest content can be downloaded to the iPhone in real time via a web connection or the content can be downloaded and stored on the iPhone for offline use.

Developing web-based content

Designing web-based content for the iPhone is perhaps the simplest solution if you wish to quickly deploy language learning applications or create an application that can run on other mobile handsets. Although the screen size can be a limitation, iPhone's native Safari web browser can render most web sites for viewing. Safari can also play **MP3**, **WAV**, **M4A** audio files and **MOV**, **MP4**, **M4V**, and **3GP** video.

Using a web based application forgoes the need to enroll or distribute applications via the iTunes app store since applications can be uploaded to any web server and deployed from there. Coding web applications is also simpler and less expensive than coding iPhone apps. Sample code and tips on how to enhance and optimize your website for Safari can be found at Apple's Safari Dev Center at: <developer.apple.com/safari/>. The disadvantage of using web apps over iPhone apps is that you lose some flexibility with the ability to interact with the content. Complex web apps such as Google Docs and web apps made with Flash

are not currently compatible with the iPhone mobile Safari browser, but less sophisticated apps using **HTML5**, **CSS** and limited JavaScript are possible.

The quickest method to begin designing web content that is “made for the iPhone” is to use a pre-fabricated framework or web kit. A ‘kit’ typically includes sample **CSS** files to control the layout of the content, some JavaScript files that need to be uploaded along with your **HTML** files, and some working **HTML** examples. The following section introduces a few open-source frameworks or kits available for download. These kits are best used for delivering static content to iPhone users.

iuui (iPhone user interface) can be used to develop iPhone web sites and web applications that have a look and feel of typical iPhone applications. iuui can be downloaded from the Google Code Project at: code.google.com/p/iuui/. The download includes a JavaScript library, a **CSS** (cascading style sheet), and navigation images for iPhone menus and interfaces. JavaScript programming knowledge is not needed to use the libraries although a basic understanding of how JavaScript is inserted into html code is helpful. iuui includes sample code that allows the display of content in either a vertical or horizontal display mode.

UIKit (Universal iPhone User Interface Kit) is a project hosted on Google Code and can be downloaded from: code.google.com/p/iphone-universal/. The free kit includes a useful **CSS** example and several sample html pages. The sample web pages included in UIKit can be used as templates and allow you add your own content.

iWebKit is another open-source kit that makes the development of iPhone web sites very easy. The iWebKit website assures that anyone can create iPhone compatible sites with no knowledge of **HTML**. The kit includes samples and instructions on how to create iPhone-like pop-up menus and multiple list types which are useful for navigating content in a touch-screen environment. It also includes code for displaying content in full-screen and for creating web-based forms. You can demo these features by visiting: demo.iwebkit.net from your iPhone browser.

Video on the iPhone

Apple’s Safari Reference Center offers several articles on creating video for Apple Safari on the iPhone. The articles labeled “Sizing Movies Appropriately,” “Don’t Let the Bit Rate Stall Your Movie,” and “Using Supported Movie Standards” are practical and informative. When adding video content to iPhone web pages or applications, keep in mind the screen size of the device. The iPhone has a 3.5 inch screen which is 480×320 pixels in the horizontal mode or 320×480 in vertical mode. Your iPhone video should retain the same aspect ratio of the video content and fit within a 480×360 rectangle. Standard video (4:3 aspect ratio) should be 480×360 and widescreen video should be saved as 480×270 in order to keep the video sharp and in full-screen view. You do not need to specify a media player in your **HTML** code when embedding video in web sites as iPhone plays video uses its own built in player. Video will be automatically resized to fit the screen. The following video compression standards are supported by the iPhone:

- ✧ H.264 Baseline Profile Level 3.0 video, up to 640×480 at 30fps.
- ✧ MPEG-4 Part2 video (Simple Profile)
- ✧ AAC-LC audio, up to 48kHz

The video file extensions that are supported are **MOV**, **MP4**, **M4V**, and **3GP**. Encoding and optimizing video in these formats can be accomplished easily using Quick Time Pro. When exporting a video, check that the iPhone video option is selected.

If you don't have access to an iPhone or iPod, you can test your websites using an online iPhone simulator such as the one at: <www.testiphone.com>.

The future of mobile devices

While the iPhone and iPad both offer new and exciting opportunities for language learning, there are some hurdles to overcome. Firstly not all students have access to an iPhone or iPad so when deploying learning applications for these platforms, devices need to be made available to learners. Some institutions are either giving or loaning devices to students. Also, there are the difficulties involved in application development and the inability to bypass the iTunes App Store to distribute applications within an educational institution. In addition, the iPhone screen is still too small for more sophisticated language learning applications and, related to this point, there is the inability to play video on only a portion of the screen. Lastly, the iPhone and iPad also do not currently support Java or Adobe Flash, two widely used technologies for creating interactive web-based applications.

References

- Apple iPhone Dev Center. (2010). Developing for the iPhone. Retrieved July 15, 2010 from <http://developer.apple.com/iphone/index.action>
- Apple Safari Reference Center. (2010). Creating video. Retrieved July 27, 2010 from <http://developer.apple.com/safari/library/documentation/AppleApplications/Reference/SafariWebContent/CreatingVideoforSafariiPhone/CreatingVideoforSafariiPhone.html>
- Daniels, P. (2006). Wired: Learning on the move. *The Language Teacher*, 30(6), 28–29.
- Wayner, P. (2009). iPhone development tools that work the way you do. Retrieved July 25, 2010 from <http://www.infoworld.com/d/mobilize/iphone-development-tools-work-way-you-do-309>