EMERGING TECHNOLOGIES

MOBILE APPS FOR LANGUAGE LEARNING

Robert Godwin-Jones
Virginia Commonwealth University

It wasn't that long ago that the most exciting thing you could so with your new mobile phone was to download a ringtone. Today, new iPhone or Android phone users face the quandary of which of the hundreds of thousands of apps (applications) they should choose. It seems that everyone from federal government agencies to your local bakery has an app available. This phenomenon, not surprisingly has led to tremendous interest among educators. Mobile learning (often "m-learning") is in itself not new, but new devices with enhanced capabilities have dramatically increased the interest level, including among language educators. The Apple iPad and other new tablet computers are adding to the mobile app frenzy. In this column we will explore the state of language learning apps, the devices they run on, and how they are developed.

THE CHANGING MOBILE ENVIRONMENT

As long as there have been portable audio-video and computing devices, there has been interest in exploring their use in language learning. As portable cassette players yielded to iPods and other MP3 players, the new capabilities of the hardware led to enhanced use of audio-based learning such as language podcasts with integrated transcripts. As PDA's (personal digital assistants) became more widespread with the advent of the Palm Pilot and its successors, language dictionaries, e-book grammars, and flashcard programs began to appear. Palm also was the producer of some of the first smartphones, which integrated PDA functions with new capabilities including SMS messaging, built-in cameras, and voice recording. With a small internal grant, I was able in 2002 to purchase smartphones for each of the participants in a study abroad program in Austria. The picture taking, text messaging, and dual-language dictionaries proved to be very useful, but the main point of having the phones—for the students to write travel diaries—proved to be problematic as the text input system (T9 keyboard) was too slow and error-prone for writing longer texts efficiently. This kind of issue was not unusual at the time. Five years ago in *LLT*, George Chinnery (2006) surveyed the state of mobile language learning. He reported on projects using mobile phones for vocabulary practice, quiz delivery, live tutoring, and email lesson content delivery, and on other projects using PDA's for file sharing, video playback and stylus text entry. In many of these instances, Chinnery reported that technical problems arose due to the limitations inherent in the devices, in particular small, low-resolution screens (problematic for image/video display or even good text reading), poor audio quality (both in phoning and audio playback), awkward text entry, limited storage/memory and slow Internet connectivity. Many of the language learning projects were seriously hampered by these issues. Moreover, the predominant operating systems (OS) for phones and PDA's at the time, namely Palm OS, Windows Mobile, and Nokia Symbian, offered limited features and expandability. All did allow, however, apps to be loaded onto devices, but they were few in number and limited in functionality. Web browsing was constrained and slow; Web navigation using a mini-joystick or a stylus was awkward and error-prone.

A huge step up in functionality arrived with the Apple iPhone in 2007. It is not only the iPhone's own advanced features which have proven to be a game-changer in the mobile area, but also the fact that its success has led competitors to create other equally capable devices. With the iPhone, Android devices, and Windows Phone 7 products, what used to be phones with added-on computing capabilities have morphed into mini-computers which can also make phone calls. These devices go a long ways towards solving the issues arising from early efforts in mobile assisted language learning. Screens are considerably larger, with higher resolution and clarity, and capable—through more powerful processors—

of playing back high-resolution video smoothly. Almost all smartphones today feature a responsive touch screen which makes Web navigation much easier. Text entry is enhanced through a relatively large virtual keyboard or a full physical mini-keyboard. Many phones are capable not only of video capture, but of video (and image) editing as well as of voice recognition. Most of the new generation of smartphones have faster 3G or 4G cellular connectivity along with even faster Wi-Fi. Built-in storage is greatly enlarged, with flash memory having in recent years become cheaper, smaller, and higher capacity. Some of the functionality of current smartphones even surpasses in some ways what is available on laptops, as many include GPS chips, accelerometers, compasses, high-resolution cameras, and proximity sensors. Most incorporate Bluetooth and USB connections as well. Clearly having such powerful devices available anytime, anyplace provides tremendous opportunities for educational use. However, it is not just—or even primarily—hardware enhancements of the iPhone generation devices that hold the most promise for use in language learning. Equally important is the software and the new opportunities that arise from mobile application development.

APPS ON THE RISE

One of the significant software enhancements of the iPhone when first released in 2007 was the much greater usability of its Web browser, Mobile Safari. Coupled with a larger, high-resolution screen, a more powerful processor, more internal (RAM) memory, and faster Internet connectivity, Mobile Safari was able for the first time on a device its size to access and display the "full" Web. Previous phone browsers used either text only browsing, server-based on the fly re-formatting (Opera Mini), or reliance on WAP (Wireless Application Protocol), a way to rewrite HTML for display on phones. Web browsing on a phone did not deliver the same Web experience as desktop browsers. Web pages on the iPhone, by contrast, are not dumbed down in any way, but are displayed as they would appear in a normal Web browser on a desktop computer. The smaller screen size effects the readability of full-page display, but the iPhone introduced touch actions such as double tap and two-finger zoom to allow smaller text to be read. Other smartphones have similar browsers. In fact, most are based on the WebKit rendering engine developed by Apple for use in Safari. Apple has made Webkit an open source project. Another significant development with Mobile Safari was robust JavaScript support, the language that supplies much of the interactivity on the Web. Also supported was CSS 2 (cascading style sheets), which not only is important for formatting Web pages but also plays a key role in structuring the page's "document object model" (DOM), an essential element in being able to change dynamically and programmatically elements of a page. At the same time, Apple introduced extensions to HTML and CSS which enhance the Web display on iPhones. As WebKit is used now across smartphone platforms, these tags are commonly supported and, in fact, are making their way into the specifications of HTML 5, the new version of the Web formatting language, not yet finalized, but already largely supported in many browsers. A major component of current Web publishing is, however, not supported on iPhones or on other mobile Apple devices, Adobe's Flash; Apple believes that HTML 5 will gradually replace the use of Flash. That remains to be seen.

Given the enhanced capability of mobile Safari, Apple initially encouraged developers to add functionality to the iPhone by creating Web apps, that is, HTML-based programs which used JavaScript and CSS to provide interactivity. Developers, however, were not satisfied with this approach, which did not provide full access to the capabilities of the iPhone, and in 2008 Apple announced that it would allow 3rd party native applications for the iPhone. Subsequently, a SDK (software development kit) was released for development of iPhone apps, built into Apple's programming environment, XCode. At the same time Apple created a curated environment for distributing the new apps, the Apple App Store, integrated into the iTunes Store. The App Store has proven to be wildly successful, with some 400,000 apps to date. Other smartphone OS's have implemented similar systems, although in general without the strict scrutiny apps submitted to the Apple store undergo. Google's Android OS, in particular, has gained significantly in the past year in both users and number of apps. There are predictions that the number of

Android apps will soon surpass those for Apple devices.

Among these iPhone and Android apps are a good number supporting language learning. Claire Siskin has provided a nice list of apps for language learning, and others have listed and reviewed apps for all languages, or for specific languages such as Japanese, French, and ESL. Many of these apps are of similar kind to those available for some time on phones, including flashcard programs, dual language dictionaries, and phrase books. Not all are of the highest quality. In some instances, newer hardware and software have allowed for enhanced functionality. Phrase books, for instance, can now hold much more content, including video as well as audio, and integrate with online sites. Some travel guides such as the Lonely Planet apps feature advanced features such as drag-and-drop trip planners, audio phrase books, and even augmented reality, which uses phone camera views to overlay local site information. Vocabulary development programs have become more sophisticated and powerful. One I have been using for studying Chinese is eStroke. Its primary purpose is to help in learning stroke order for writing Chinese characters, but it also includes an extensive dual-language dictionary, features excellent animations, and includes personal library and quizzing functions. Another popular app for Chinese is Pleco, which starts out as a free app, but adds functionality through a large number of paid add-ons such as specialized dictionaries, enhanced handwriting recognition, and optical character recognition. ChinesePod has a nicely designed app which offers a variety of tools to work with lesson podcasts and their vocabulary/phrases. The app also automatically syncs the user's learning status on the app with that on the Web site and allows lesson content to be downloaded for off-line study, one of the benefits of apps over the live Web. Another nice feature new smartphones offer Chinese learners, and anyone else using a non-Latin writing system, is the ease with which one can switch the virtual keyboard's text input system, making it possible on the iPhone, for example, to enter Chinese characters by drawing them with one's finger or switching to pinyin text entry with then the corresponding character equivalents displayed for selection.

Flashcard programs have also become more powerful. A popular program of this type is Anki, a spaced repetition vocabulary study program (discussed in the LLT 14/2 column). The mobile version offers essentially the same powerful functionality as the desktop version, including deck and individual card editing, audio support, and customizable review options. It also syncs with the desktop and Web versions. The popular Quizlet flashcard system also offers a mobile app, which has an interesting auto-define function when adding new items, which allows the user to see/choose definitions that other Quizlet users have entered for that term. Wordreference.com's app links to language discussion forums that reference the term searched. Conjugation Nation offers apps in a variety of languages for drilling verb forms. Linking a mobile app to Web services or an online database is being used more and more in language learning apps, in particular in commercial products such as Rosetta Stone or Transparent Language's Byki, as well as in online services such as Babbel.com or hello-hello, all of which have mobile apps which sync mobile and desktop versions. Complete language courses, such as Living Language for French, are now also being offered as apps. An interesting approach to leveraging the client-server relationship on mobile devices is the Cloudbank project described by Pemberton, Winter, and Fallahkhair (2009). It uses crowdsourcing to build a database of informal English language usage, featuring an Android app communicating with a database through Web services. It also uses RSS feeds to keep users informed of new content added.

Cloudbank leverages not only the ability of a smartphone to exchange information with an online database; it also makes use of a peer-to-peer network. In fact, with the rise of social networking, we are seeing more language learning apps that take advantage of this trend. The Byki app for example, allows users to search for use of terms within Twitter messages. The Micromandarin project uses the location aware program Foursquare to provide contextually relevant content for language practice. The app uses GPS to determine a user's location and supplies vocabulary information and practice appropriate to that location: food and drink vocabulary, for instance, if the user is in a restaurant. The CLUE project makes

similar use of GPS to supply location appropriate content and adds another dimension through tagging objects with RFID tags (radio frequency identification), whose information then can be retrieved on the smartphone. Beaudin, Intille, Tapia, Rockinson and Morris (2007) describe a similar project for vocabulary learning, using objects in the home with stick-on sensors. It seems likely that we will see app development in the future take greater advantage of some of the hardware features of new smartphones beyond the GPS chip. The accelerometer, for example, used extensively in mobile game applications, could be used in language learning games as well.

Claire Siskin's list of apps for language learning includes a category called "repurposed apps," which discusses general purpose apps that could be used in language learning, including voice search, voice email, postcard creation, audio recording, and children's games. Integrating audio capabilities adds a crucial component of language use and learning. A good many e-books are becoming available, especially for the iPad, which combine text, images, and audio in an attractive way. Some also include games. Many of these, such as the Town Musicians of Bremen, are designed for children, but clearly would be of interest for language learning. Google Translate for Android offers an interesting experimental feature using voice. "Conversation mode" lets users translate an utterance into the target language, which is then read aloud. One's conversation partner can then speak in the target language and have in turn that response be translated and read aloud. Another Android voice translation app is Talk to Me, which has gotten positive reviews. While newer smartphones include voice recognition, including in some cases for languages other than English, this feature does not yet appear to have worked its way into apps.

DEVELOPING FOR MOBILE DELIVERY

In order to take full advantage of the hardware and OS capabilities of a mobile device, developers need to create an app using an approach consistent with that device's programming environment. For Apple iOS devices (iPhone, iPod Touch, iPad), that means using Objective-C and Apple's XCode developers' tool. However, such apps will not run on Android devices, for which apps are written in Java running on a version of Linux. iOS apps are available exclusively from the Apple App Store (unless the iPhone is "jailbroken," i.e., unlocked for open access), while Android apps are more widely available, from the official Android Market, Amazon's Appstore for Android, Handango, and other repositories. As is the case with iOS apps, Android apps are free or available for a price, usually under ten US dollars. Some apps are considerably more expensive—buying all the add-ons to Pleco costs US $149. Apps are usually available in English only, few are localized for other languages. Other smartphone environments (Blackberry, WebOS [Palm/HP], Windows Phone 7) use different programming environments, all mutually incompatible. All of the different smartphone software companies make development tools available, which typically include a desktop phone simulator. Most are free, or available for a modest cost. Except for iOS development, which is Mac OS only, app development can be done on Windows or Macintosh machines, and, in some instances, on Linux.

App development is currently progressing at a feverish pace, and app developers are in high demand. However, for language learning purposes, native app development may not be the best choice. Not only are the programming environments different, each also involves knowing or learning a programming language such as Objective-C or Java. Moreover, there is little carryover from developing an app in one environment to re-creating that app for a different platform. Obviously, the fundamental functionality and user interface design could be the same, but the programming will be altogether different and done using different tools. For educational use, as things stand now, it would seem that one at the least would want to have an iOS and Android version of an app, probably a Windows Phone 7 app as well, and possibly WebOS or Blackberry versions, depending on one's user base and the market rise and fall of the various companies' products. This could prove to be a time-consuming and expensive development process.

An alternative to developing native apps is to create instead a Web app. This involves using more familiar and easier-to-learn HTML, JavaScript and CSS. All are scripting rather than programming languages,

hence do not involve compilation into byte-code. The only tool needed is a text editor. Web apps will run and perform similarly in most smartphone environments, particularly as all but Microsoft now use WebKit. The look and feel can be quite similar to built-in apps, particularly if one uses relatively new HTML/CSS tags such as the "viewport" meta tag and CSS "webkit-border" rules. Icons/shortcuts to the Web app on the home screen allow it to be launched in a similar way to native apps. Distribution for Web apps is through a Web server, rather than from an app store. What does one sacrifice creating a Web app rather than "going native"? Execution speed is likely to be slower and the user interface not as slick. There will also be more limited access to the device hardware, including its camera, audio player or GPS. These considerations may or may not be of consequence, depending on the nature of the application. They may be outweighed by the advantage of creating one app which can be universally deployed. My second year German students have been using for the past year a simple flashcard Web app I created, which is linked both from the Blackboard course Web site and from an open, mobile-friendly link. This allows the students to use the app both from desktop browsers and mobile devices, something not doable with native iPhone or Android apps.

Another possibility is to create a "hybrid app," a Web app which is then ported through a tool such as PhoneGap to the native environment of the smartphone. This facilitates linking to some hardware features of the device. It also allows for possible distribution through one of the app stores. A number of Web apps created with PhoneGap are available from the various app stories. Creating a Web app for mobile distribution through PhoneGap or similar tools such as Appcelerator Titanium can be much easier through using templates such as those available from Mobile Boilerplate or by using a mobile-oriented JavaScript library. Among the latter is jQuery Mobile, an extension to the popular and free jQuery library. Using jQuery Mobile makes it easy to create parts of a Web app such as navigation, form elements, and page transition effects without having to write the JavaScript oneself. It supports most smartphone platforms (but not yet Windows Phone 7) and features progressive enhancement, meaning that its advanced features degrade gracefully if not supported in a given mobile browser, while maintaining across all browsers the same essential content and functionality. If not supported, for example, page transitions such as fading, flipping or sliding will simply not appear, but the new page will still be displayed.

Another kind of mixed environment approach that is getting wide usage is the creation of Web-based content that automatically re-formats itself for display on a small screen. This approach uses a feature of CSS 3 called CSS media queries, which is widely supported on both mobile and desktop browsers. This involves adding a tag to the HTML header to direct a Web browser to use a size appropriate CSS style, as in the following example:

<link media="only screen and (max-device-width: 480px)" href="mobile.css" type="text/css" rel="stylesheet" />

In this case the page formatting will be determined by the "mobile.css" style, rather than the main CSS linked in the header of the page, if the device being used has a maximum width of 480 pixels. A similar process has been possible for some time to enable optimization of a print copy of a Web page. What is new here is the ability to specify a screen width to be used in connection with a particular style. A mock-up of an online journal page from the Web design site "A List Apart" demonstrates this and displays differently depending on screen width, with the pictures either displayed in 2 columns on a phone (480 pixels wide or less), 4 columns on a typical monitor (480 to 600 pixels) or 6 columns on a widescreen monitor (wider than 600 pixels). The navigation buttons also change location depending on the screen size, namely moving to the top for a small screen. While this approach has a number of devotees, others advocate creating separate HTML pages for mobile devices. It is a trade-off between more complex code which adjusts automatically to different screen size or simpler code which must be maintained and synced in different file locations.

One approach that many language developers have used in the past in creating Web-based interactivity is

problematic in the mobile sphere, namely Flash. Traditionally, Flash has been used for video streaming, animation, and for general interactivity. Flash is not likely to ever be supported on iOS devices, but it does run on other mobile devices. Android 3.0 and some 2.x versions support Flash. However, Flash performance on mobile devices is not as robust as it is on desktop platforms. It tends to run more slowly and occasionally crashes the system on some devices, due mostly to memory issues. Adobe has been working on better performance on mobile devices, and it seems likely performance will improve in the near future. However, if it's possible to use HTML 5 rather than Flash—which may or may not be possible—that is advisable for the widest possible compatibility.

OUTLOOK: TABLETS ALSO ON THE RISE

Complicating app development even more is the arrival of touchscreen tablets. The iPad, introduced in 2010, has been a phenomenal success for Apple, with sales far exceeding most expectations and eclipsing sales of earlier tablet computers, which never caught on except in narrow niche markets. Apps developed for the iPhone will also run on the iOS-based iPad but to take advantage of the larger screen need to be modified, which may entail a revamping of the user interface. One of the first commercial language learning apps designed specially for the iPad was the heavily marketed hello-hello app, available for several different languages. Meanwhile, tablets from other manufacturers are becoming available, many using the Android OS. The Android tablets vary in sizes, most either 7 or 10 inches, with likely more variation in future models. Given this scenario, it seems all the more advisable for developers to consider creating a Web app with a fluid grid that adjusts automatically to different environments. It looks likely that tablets will be a popular product in the near future, so having language learning applications that work in that environment seems highly desirable. Of course, a special use case may make creation of a native app more appropriate, especially if the target audience has a marked predominance of one platform, or if hardware linkage is an important part of the project. It's unfortunate that today in mobile software development, we seem to have gone back to the days when developers had to make a choice that excluded a large part of their possible market, as in deciding between Mac-based HyperCard or Windows only Toolbook. The Web has been an environment which has brought peace to the platform religion wars but we are starting to see a new war of words being waged between iOS and Android partisans.

As recently as 2007, a comprehensive review of mobile assisted language learning by Agnes Kukukska-Hulme and Lesley Shield found that for the most part uses of mobile devices were pedestrian, uncreative, and repetitive and did not take advantage of the mobility, peer connectivity, or advanced communication features of mobile devices. Most activities were teacher-led and scheduled, not leveraging the anytime, anyplace mobile environment. Oral interactions and learner collaboration were infrequently used. The problem is less one of hardware/software shortcomings and more in developers' conceptualization of how language learning could be enhanced in new, innovative ways with the assistance of mobile devices. The new mobile computing environment ushered in by the arrival of the iPhone gives us even more capability of which to take advantage. It would be a shame to fall into only the same use patterns as in the past. In a recent post to his mobile ESL blog, David Read describes what he would like to see in a language app. He envisions a photo translation function that would make use of the built-in camera as a scanner to read in, recognize, and translate items from menus, posters, or other realia, similar to how that works now in the SnaPanda program (Android). He would also like to see new approaches to accessing language corpora on small screens as well as ways to look up and display items simultaneously from a variety of online dictionaries, with the added ability to add items from all these sources—scanning, look-up, corpora—to a personal word bank. It would be interesting to see such functions combined as well with an intelligent tutoring system (ITS). A step in that direction is the TenseITS project (Cui & Bull, 2004) which featured a mobile ITS using context and location of the user to determine which verbs to use in drill exercises. The PDAs used in the project were hampered by limited memory capacity; the new generation of mobile devices could make mobile ITS a more doable proposition. Chen and Li (2010) describe a project which combines context/location awareness with a rudimentary kind of ITS, with content delivered based on a

user's profile/learning history and current location. In this case location was determined by nodes in a wide-area network, but GPS could also be used. With the good connectivity now available on mobile devices, adding a social dimension to location-aware learning apps would be beneficial, allowing users to share context-specific learning experiences.

It is not just the mobility, enhanced hardware, and better software of new mobile devices that should encourage new thinking. The devices in and of themselves encourage a new kind of relationship between user and machine. The responsive touchscreen interface seems to create a more personal, even intimate connection, becoming part of one's personal identity. According to a recent report on creating mobile apps from Forrester Research, the emotional bond often created is something to keep in mind when developing mobile apps. The devices are more personal also in the sense that they are individually highly customizable and small enough to be always within reach. It's also the case that both smartphones and tablets tend to focus the attention more on one task at a time than is the case with regular computers. Although multi-tasking to one degree or another is available on these devices, the screen size and touch interface tend to invite users to focus exclusively on the program running in the foreground. For educational uses, this may present a welcome opportunity to capture, at least for a short time, the full attention of the learner. Desktop and laptop computers will continue to be used, but as mobile devices proliferate, their use may change. Apple devices are still tied to using a computer for storage and syncing, but the predominant movement these days is towards over-the-air syncing and resources residing "in the cloud" rather than on a personal computer. With faster Internet connections, client-user interactions through Ajax (JavaScript-based server interactions) or other means work faster and smoother, making it possible to draw data more efficiently from online sources for smoother interactions in an ITS or other program involving heavy data usage.

As personal devices, smartphones are ideal for individualized informal learning. The user determines which apps to acquire and how to use them. As language educators, we should encourage and assist the learner autonomy this enables and provide means for learners to combine formal and informal learning. Song and Fox (2008) describe a project which features an open-ended, student-oriented approach to vocabulary learning in which EFL students were provided access to and guidance on using a variety of vocabulary building tools. The article describes the considerable variety of tools and approaches eventually chosen by the students. This kind of activity becomes even more powerful when coupled with the ability for students to show or discuss their methods and findings with their peers. The photo blogging project described by Wong, Chin, Tan, and Liu (2010) involved students using iPhones to take photos to illustrate Chinese idioms being studied and to share their photos and comments with the class through a wiki. Students were encouraged to take photos based on their daily lives using their immediate surroundings. This use of the student's actual environment improves upon similar projects that have used an artificial space such as a lab (Stockwell, 2008) or a classroom (Liu, 2009). We know that learning becomes more real and permanent when tied to learners' lives outside the academic environment. Mobile devices are a great way to achieve that goal. Of course, it's important to keep in mind that we are far from seeing universal ownership of smartphones—they are still too expensive for many budgets. Designing exclusively for smartphone usage will necessarily exclude many users. Smartphone penetration will likely gain worldwide in coming years, but not at the same pace everywhere. At the same time, phone and tablet models—both hardware and software—will evolve from their current state. Given how competitive and profitable that market has become, the pace of innovation is likely to be rapid. As mobile devices become even more powerful and versatile, we are likely to see more users make them their primary, perhaps their sole computing devices. This is not a trend language educators can ignore.

REFERENCES

Beaudin, J. S., Intille, S. S., Tapia, E. M., Rockinson, R., & Morris, M. E. (2007). Context-sensitive microlearning of foreign language vocabulary on a mobile device. In B. Schiele, A. K. Dey, H. Gellersen, B. de Ruyter, M. Tscheligi, R. Wichert, E. Aarts, & A. Buchmann. (Eds.), *Ambient intelligence* (pp. 55–72). (Springer Lecture Notes in Computer Science). Volume 4794/2007. Berlin: Springer.

Chen, C-M., & Li, Y-L. (2010). Personalized context-aware ubiquitous learning system for supporting effective English vocabulary learning. *Interactive Learning Environments, 18*(4), 341–364.

Chinnery, G. M. (2006). Going to the MALL: Mobile assisted language learning. *Language Learning & Technology, 10*(1), 9–16. Retrieved from http://llt.msu.edu/vol10num1/pdf/emerging.pdf

Cui, Y., & Bull, S. (2005). Context and learner modelling for the mobile foreign language learner. *System, 33*, 353–367.

Kukulska-Hulme, A., & Shield, L. (2007). An overview of mobile assisted language learning: Can mobile devices support collaborative practice in speaking and listening? Paper presented at EuroCALL 2007, Conference Virtual Strand, September, 2007. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.84.1398&rep=rep1&type=pdf

Liu, T.-Y. (2009). A context-aware ubiquitous learning environment for language listening and speaking. *Journal of Computer Assisted Learning, 25*(6), 515–527.

Pemberton, L., Winter, M., & Fallahkhair, S. (2009). A user created content approach to mobile knowledge sharing for advanced language learners. *Proceedings of mLearn 2009*, Orlando, Florida, 184–187.

Song, Y., & Fox, R. (2008). Using PDA for undergraduate student incidental vocabulary testing. *European Association for Computer Assisted Language Learning, 20*(3), 290–314.

Stockwell, G. (2008). Investigating learner preparedness for and usage patterns of mobile learning. *ReCALL, 20*(3), 253–270.

Wong, L.-H., Chin, C.-K., Tan, C.-L., & Liu, M. (2010). Students' personal and social meaning making in a Chinese idiom mobile learning environment. *Educational Technology & Society*, *13*(4), 15–26.

RESOURCE LIST

Language Learning and Mobile Apps

- Language Learning Applications for Smartphones, or Small Can Be Beautiful – Clair Siskin's list

- Brief Review of Language Learning Apps – HRC Blog

- Learnosity Blog : Mobile Applications for Language Learning

- Move Over, Rosetta Stone: Mobile Language Apps Make Learning Fun

- Mobile Application for Language Learning – MALL Research Project Report from the schools online initiative

- Cool Apps for Language Learning

- 50 iPhone Apps to Help You Learn a New Language

- How I'm using my iPad to learn languages

- Google Docs gets Android, iPhone editing in 44 languages

- Mobile Language Learning: Learn Japanese on the Go

- Mobile ESL: My perfect language learning mobile app

- TOTALe Companion – For Rosetta Stone

- Byki Mobile App

- Quizlet – App

- Conjugation Nation – Verb form app

- WordReference.com app

- CloudBank Project – Crowd-sourcing project with Android app

App Development

- Mobile application development – Good introduction from Wikipedia

- Mobile app development trends - what languages should you be learning? – Nice overview of different platforms

- Mobile App Design Best Practices - Forrester Research – Comprehensive but expensive

- What is Android? | Android Developers – Good starting point

- iOS Development Center – Starting point from Apple for developing iPhone and iPad apps

- BlackBerry Developer Zone

- WebOS Developer Center

- Introduction to Windows Phone 7 Development)

- How To Port an iPhone Application to the iPad

- Green's Opinion: From iPhone to iPad: Creating a Universal Application

Web Apps and Mobile-friendly Web Publishing

- Mobile Web Best Practices 1.0 – From the W3

- ADL Mobile - Mobile Learning Research

- Designing Web Sites for Phone Browsers – Microsoft

- New to Mobile? Welcome to the One Web Debate

- Mobile Application Development: Web vs. Native - ACM Queue

- Responsive Web Design or Separate Mobile Site? Eh. It Depends

- A List Apart: Articles: Responsive Web Design

- A Flexible Grid

- jQTouch - jQuery plugin for mobile Web development

- Sencha - Desktop and Mobile JavaScript Frameworks

- PhoneGap

- Baker Ebook Framework

- jQuery Mobile | jQuery Mobile

- Nuance Mobile Developer Program: Dragon Mobile SDK

- Need a Mobile Web App Template? Mobile Boilerplate 1.0 is Here

- CSS3 Media Queries

- css3-mediaqueries-js – Library to use css media queries in supported browsers

Quizlet is a learning app - a computer program you use on your mobile device. It can help users build and test their knowledge of English words and terms. Quizlet has word sets for millions of subjects. And, it is quickly becoming a useful mobile tool for language learners. Thorne teaches at Columbia University in New York City. Thanks to learning apps, Jeff Strack, another English teacher, has also noted improvement in his students' ability to remember information. Strack teaches at Hostos Community College, also in New York. He and Thorne are part of a growing number of language educato Mobile learning enhances the mobility Â· of learning process without time constraint. Â· Mobile -Assisted Language Learning (Mall). Â· Mobile -assisted language learning is the subdivision of both. M-Learning and computer-assisted language learning (CALL). Beatty defines CALL as "a term used for the. Â Classification of Mobile Learning Apps The statistical data of mobile apps exhibits the rapid progress in numerous apps being developed and downloaded every year by people around the world. English learning apps available in the app stores are plenty and choosing the right app is definitely a tiresome job.