MOBILE LEARNING APPLICATIONS USING GOOGLE APP INVENTOR FOR ANDROID

Armand Florin BERTEA
Textile, Leather and Industrial Management Faculty, „Gh. Asachi” Technical University, Blvd. Dimitrie Mangeron, Iasi Romania
armand_bertea@yahoo.co.uk

Abstract: Google Apps is a suite of online applications, launched for the first time in February 2007, which allows the creation, sharing, and publishing of documents, spreadsheets, presentations, from any computer with an Internet connection. Many of the services offered by Google Apps may be used in educational purposes. In July 2010 Google launched App Inventor for Android software, which permits users to create applications for Android in a graphical environment. In this paper we present the main features of App Inventor, analysing its capacity to develop m-learning applications for smartphones. The process of creating tests and quizzes using App Inventor for Android is examined and discussed.

Keywords: mobile learning, Google Apps, App Inventor, QuizMe.

I. LEARNING IN A MOBILE WORLD

The world of today is a place where everything tends to become mobile: the phones are mobile, computer games are handheld and advertising is ubiquitous. Hence there is no surprise that the learning technologies are following the trend and the e-learning becomes more and more m-learning.

1.1 The mobile revolution

The rapid evolution of mobile learning originates mainly from the rise of the mobile telephony, and, only in the second place, from the educational and pedagogical needs of flexibility and the anytime anywhere demand for learning, because the saying “Technology first, pedagogy later” is today as valid as ever [1].

There are some that believe that mobile learning will substitute e-learning in the years to come [2], but there are voices that consider that m-learning holds a secondary place in the field of modern educational technologies [3]. Even more, there are opinions that mobile learning, especially when associated with mobile phones, is more of a play than a form of learning [4]. In any case, it may be noticed that it is for the first time when a form of educational technology was more gladly embraced by the students than by the teachers [5].

A major moment in the history of mobile learning is the appearance of smartphones, which are in fact handheld computers integrated with a mobile telephone, capable of installing and running applications similar to the ones for laptops or desktops. Smartphones have become important tools in XXI century society, because of their abundant functions including communication, entertainment and online office, and, last but not least, mobile learning [6].

1.2 Software for mobile devices

Smartphones run dedicated operating system software that offer a platform for application developers, and mobile software development gained in importance. Amongst the most popular mobile OS platforms are Java ME, BlackBerry, Windows Mobile, Palm OS, Symbian, iPhone and Android [7]. The majority of them are proprietary platforms, with the exception of Android. Each mobile OS
platform uses a programming language: Objective-C for iPhone OS, Visual C#/C++ for Windows Mobile, Java for Android and BlackBerry OS, C/C++ for Palm OS and C++ for Symbian.

One of the leading operating systems for smartphones is Android, an open source system platform backed by Google, together with the members of Open Handset Alliance (a business alliance of firms for developing open standards for mobile devices that includes, besides Google, HTC, Dell, Intel, Motorola, LG and Samsung, to name only some of them) [8].

The Android operating system consists mainly of a virtual machine that runs on the Linux kernel, plus APIs (sets of data structures, protocols, routines and tools for accessing web-based software applications) and a collection of built-in applications. The OS has a built-in web browser, based on the WebKit layout engine, also used by iPhone and Palm webOS, in addition to Chrome, Google’s desktop browser.

1.3 Applications for mobile learning

The mobile technologies lead to changes at all the levels of educational institutes, especially in faculties. One major advantage is the fact that students are capable of taking their learning with them beyond the school perimeter; beside the independence from physical boundaries, new ways of interactions in the classroom become available. The academic community is taking advantage of the opportunities offered for interdisciplinary work and collaboration [9].

The mobile learning applications can offer assistance and enhancement of traditional learning, synchronous interactions between the students and teachers, and many ways of interacting with course content.

From the point of view of their usability, mobile applications can be categorized in five groups: learning management, supportive, content-based, context based and collaborative [10].

II. APP INVENTOR AS PART OF GOOGLE APPLICATIONS

Google started in 1996 as a small Internet search engine experiment, and in less then 10 years, because of the huge popularity of this search engine, Google became a verb. Besides this, Google’s growth has generated a series of products beyond the company's core web search engine, such as the Android mobile phone operating system or Google Apps, a group of web-based programs. The programmes that are encompassed in Google Apps run in a web browser, without the need to install software, and consist of communication tools, productivity tools, a customizable start page and Google Sites. The most innovative creations from worldwide developers are grouped in Google Labs, situated on a separate site at googlelabs.com.

2.1 Short description of Google Apps and Google Lab

Google Apps is a suite of tools that Google has brought together to meet the needs of businesses, schools, government agencies and other organizations of any size: Gmail with built-in voice and video chat, Google Talk, Google Calendar, Google Docs (which includes Google Documents, a word processor similar to Microsoft Word, Google Presentations - a multimedia presentation tool similar to Microsoft PowerPoint, Google Spreadsheets - a spreadsheet program similar to Microsoft Excel and Google Drawings - a graphic design program), iGoogle and Google Sites (Figure 1) [11].

It was launched in 2006, and besides a free edition, there is Google Apps for Business, Google Apps for Government and Google Apps for Non-profit, that offers more storage space (25GB email storage per user) and supplemental features (Google Video for Business, Google Groups for Business, SSO, forced SSL, custom password strength requirements) on yearly fee basis.

Google Apps for Education Edition contains exact the same extras as the Business Edition and is offered at no cost to K–12 and higher education. In July 2010 Google launched App Inventor for Android software, which permits users without traditional computer science backgrounds to create applications for Android in a graphical environment [12].
On the other hand, Google Labs is an antechamber of Google Apps. If the applications created by developers in Google Apps have favourable feedback and proves to be worthy, they can be integrated into Google.com (for example, Google Maps started as a Google Lab project). Presently, some of the most popular programs in Google Labs are Google Body, a detailed 3D model of the human body, Google Books Ngram Viewer that shows how often phrases occur in the world's books over the years and Earth Engine Global, a scale analysis for global-scale problems. There are even experiments grouped by type of Google Application: Calendar Labs, Gmail Labs, and Google Maps Labs.

### 2.2 App Inventor main characteristics

Google App Inventor for Android was created with the intension to permit users without coding experience to make simple apps for mobile phone. Initially the access to the application was allowed only by invitation, but from December 2010 Google granted full access to anyone that has a Google account.

The applications, that have a graphical drag-and-drop programming interface, can be used with ease because, instead of writing code, one can visually design an application using blocks to specify the app’s behaviour. The blocks of code are shown as graphic images and represent different smart phone capabilities. Using App Inventor is like putting together a puzzle, as the blocks are connected to make mobile phone applications with no need of code writing knowledge.

App Inventor consists of two main parts to: a browser-based design screen and the Javabased Blocks Editor. The image of a blank application designer window in a web browser is shown in Figure 2.
It can be seen in the left a palette with some elementary user interface components: Buttons, Canvas, Labels, Checkbox, Image, etc. Below these basic tools, there are more complex tools such as Media, animation, Social and Screen Arrangement.

This Editor is launched in a separate window by clicking a button. There is need of an Android phone or an Android phone emulator, which can be downloaded and installed. The window of the Blocks Editor with the emulator is shown in Figure 3.

To use App Inventor, users work in their computer's browser, and the application appears on the Android phone connected to the computer, or in the window of the emulator.
2.3 Using App Inventor to make simple tests for m-learning

In order to make a test using App Inventor, two apps have to be made, MakeQuiz and TakeQuiz, the first one for the professor that creates the test, the second one for the students taking the test. The application is build using QuizMe as template, a Google App trivia game. Using MakeQuiz, the questions and answers are saved into database, from where TakeQuiz loads them. The two apps can connect via a shared web database, as it can be seen in Figure 4.

![Figure 4. Connection between Makequiz and Takequiz via shared web database](image)

The buttons for questions and answers have been made using Items from the Basic and the Screen Arangement Pallette, and afterwards their names have been changed according to the needs of the test; after opening the Block Editor and connecting with the emulator, the results of the transformations can be seen on the screen of the phone.

The Blocks Editor has been used in order to add behaviours to the Components.

![Figure 5. MakeQuiz final layout screen](image)

In Figure 5 it can be seen the final layout screen of the MakeQuiz application. The logic of the application can be described by presenting the processes which take place after clicking the “Add question” button and the “Delete question” button. To show them, the Blocks Editor shall be used. The content of the 2 blocks can be seen in Figure 6.
Only by reading the blocks, one can easily understand the logic of the application. When clicking the “Add question” button there are 7 steps that will be done. The first 2 are identical and they imply adding the values entered in the question textbox (TextBox1.Text) and answer textbox (TextBox2.Text) to the designated lists of items which are the QuestionList and AnswerList. After that a predefined procedure is called to display the newly input data. The name of that procedure is “displayQAs” as seen in the third step. The fourth and fifth steps are meant to clear the contents of the question and answer textboxes. Finally the two lists are stored in the online public database using the property “StoreValue”. Each of them has a unique tag that must be used accordingly when retrieving the contents of that list. This will be presented in the TakeQuiz application.

The “Delete question” button has the following behaviour: when clicked, the application first checks if the question to be deleted is actually in the questions list. If so, then the question and the corresponding answer are removed and the “displayQAs” procedure is called to display the remaining items in the lists.

Figure 7 shows the final MakeQuiz application running in the emulator.

The TakeQuiz application uses the created quiz and allows the users to answer the questions, getting instant results. The application screen can be seen in the designer as shown in Figure 8.
There are 3 buttons on the screen. The “Take test again” button is not visible until the user has answered the final question of the quiz. The logic of the application is best described by presenting the blocks behind the “Answer” button which can be seen in Figure 9.

The first block of the “Answer” button onclick event is a control “if-else” block. So first it is checked if the answer provided by the user is the correct one, as specified in the list of answers. If so, the message “Correct!” is displayed on the screen and the variable in which the correct answers are counted is incremented by 1. Otherwise, if the answer is wrong, the according message is displayed along with the correct answer. After this control block, the “Answer” button is disabled and the “Next question” button is enabled. By doing so, it is prevented the user from answering the same question more than once during the same quiz. Finally, it must be checked if the current question is the last one in the quiz, so another control block is used. If it is the last question, the quiz results are displayed and the “Take quiz again” button is enabled.

In Figure 10 is shown the final TakeQuiz application running on the emulator.
III. CONCLUSIONS

App Inventor is a straightforward and intuitive Android tool, implying no programming knowledge, which can be used to provide useful mobile learning applications.

Besides its extraordinary potential, this very new application has some problems. One important drawback of App Inventor is the fact that, even though simple applications are easy to make, in order to create complex apps more profound knowledge is necessary. But this can be seen also as an advantage as well, as Google App Inventor can be considered as an invitation to novices to extend their knowledge in code writing. In this context, it is a pity that Google does not give the possibility to look into the back-end code. An option to generate the code would offer larger possibilities for more experienced users.

Programs made using App Inventor usually consist of a single screen, and they have no or little capability to interrelate with other programs or access functionality outside the device. Another deficiency is that App Inventor doesn’t (yet) include the capacity to store files on a secure digital card, or a way to list the files on the SD Card.

Acknowledgements

This paper was realised with the support of POSDRU CUANTUMDOC “DOCTORAL STUDIES FOR EUROPEAN PERFORMANCES IN RESEARCH AND INOVATION” ID79407 project funded by the European Social Found and Romanian Government.

References


