Using App Inventor as tool for creating mathematics applications for mobile devices with Android OS

Bogdan CIORUȚA[1,2*], Mirela COMAN[2], Alin-Andrei CIORUȚA[3]

[1]*Technical University of Cluj-Napoca - North University Centre of Baia Mare, Office for IT and Communication, 62A Victor Babeș str., 430083, Baia Mare, Romania*
[2]*Technical University of Cluj-Napoca - North University Centre of Baia Mare, Faculty of Engineering, 62A Victor Babeș str., 430083, Baia Mare, Romania*
[3]*Technical University of Cluj-Napoca - North University Centre of Baia Mare, Facuulty of Science, 76 Victoriei str., 430122, Baia Mare, Romania*

ABSTRACT

People is programming on their personal computers since the 1980s, but today's mobile applications are making computing “personal” as never before. Never in the history of the use of technology in education has there been a technology so widely available to citizens as mobile technology. According to this trend, we are moving into a new era of consuming information via mobile computing, one that promises greater variety in applications, highly improved usability, and networking. Our consuming information culture gives us all sorts of opportunities for entertainment, even learning using high-tech devices, which are considered to be black boxes to most of us, because only few people can create applications. Even that, more and more students are interested in developing their own mobile applications. In this context, we present our experience with App Inventor - a friendly user platform and drag-and-drop block language to create Android apps. The present paper is a intro to the creation process for an applied mathematics app, started 4 years ago, corresponding to the requirements of our classroom consuming information culture.

Keywords: App Inventor, Android devices, applied mathematics, mobile learning

1. INTRODUCTION

People is programming on personal computers since the 1980s, but today's mobile applications are making computing “personal” as never before [1]. Never in the history of the use of technology in education has there been a technology so widely available to citizens as mobile technology. According to this trend, we are moving into a new era of consuming information via mobile computing, one that promises greater variety in applications, highly improved usability, and networking.

We confront with another informatics perspective - mobile learning which seems to be very similar to e-learning, but different at the same time (**Error! Reference source not found.**) because mobile devices - the fundamental vectors for delivering mobile learning - bring a

* Tel.: +40 742179033; fax: +40 264202986.
E-mail address: bogdan.cioruta@staff.utcluj.ro.

new dimension to learning and education allowing learning to occur in authentic contexts and extend to real environments.

These authentic contexts and real environments, together with the mobile devices, enable student learning activities characterized by communication, collaborative knowledge building, observation (facilitated by the mobile devices cameras), and innovation (using new devices and new contexts). Thus, using mobile devices students are able to construct useful knowledge in real situations [2].

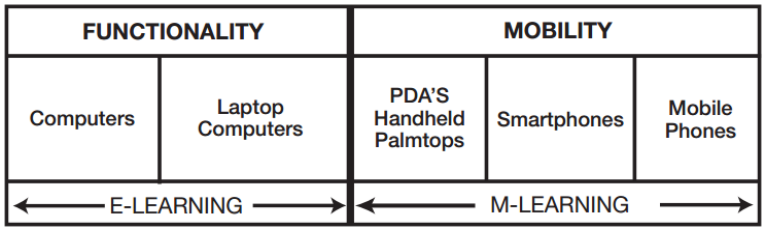


Fig. 1. The main differences between e-Learning and m-Learning

Our information consumer culture gives us all sorts of opportunities for entertainment, pleasure and sometimes even learning [3] using computers and other devices. The high-tech devices (cell phones, tablet computers, TVs etc) that we use today to consume entertainment and information are considered to be black boxes to most of us, because only few people can create applications for these gadgets. Even that, more and more students are interested in developing their own mobile applications. Incorporating mobile technology into classrooms can greatly capture student attention, better student engagement, and improve student achievement.

Today, we have mobile portable computers, e.g. smart-phones, tablets. More significantly, today is increasing the social and personal experiences that fundamentally transforms our experience and our perception of our world. Experts predict this “perfect storm” requires transformation in higher education practices to make a quality postsecondary education affordable, relevant, accessible, and desirable. The Educause Center for Applied Research study the university student perceptions and usage of technology. Their reports are shown in Figures 2-5 (Students & Technology, 2011).

Fig. **2** shows that 43% of students agree the idea that educational institutions needs more technology, concluding that improving the availability and effectiveness of technology on university campus is an real opportunity for learning system.

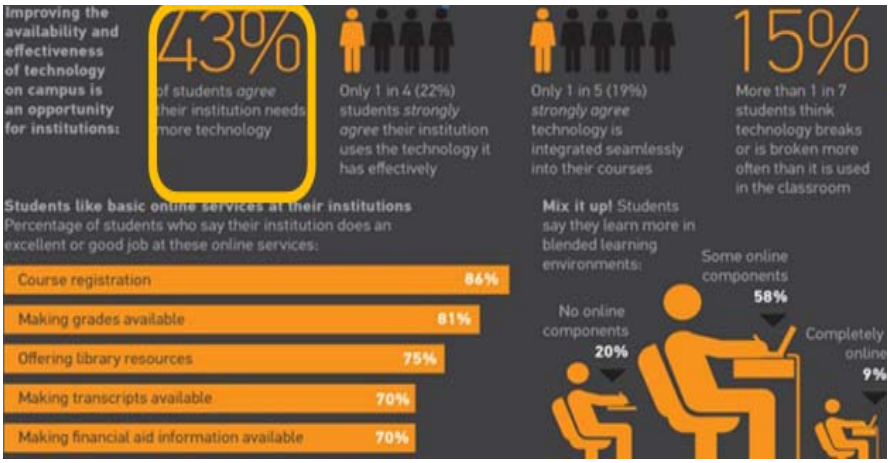


Fig. 2. Student perceptions of technology use at universities (ECAR, 2011)

Fig. **3** presents the student perceptions referring to software use in higher and university educational system. I underline the information that seems to be the most relevant for the topics of this thesis such as the student perceptions connected to programming languages.

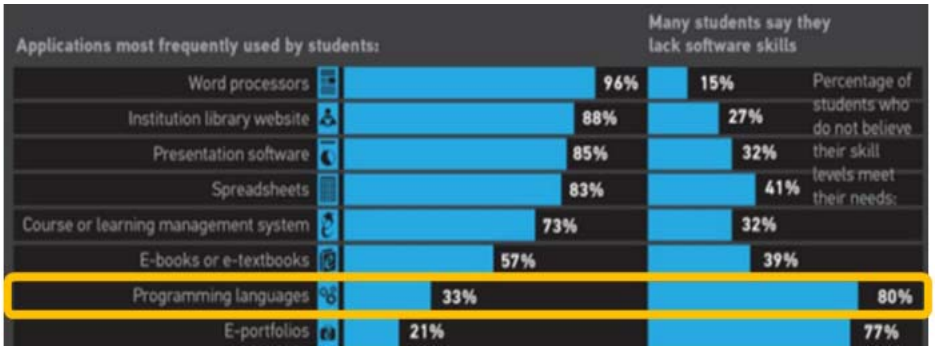


Fig. 3. Student perceptions of software use in higher and university education (ECAR, 2011)

Fig. **4** illustrates the student perceptions of hardware use in educational system. As I underline about 55% of questioned students own smart phones as technology devices, and only 37% of them have used this devices for academics before.

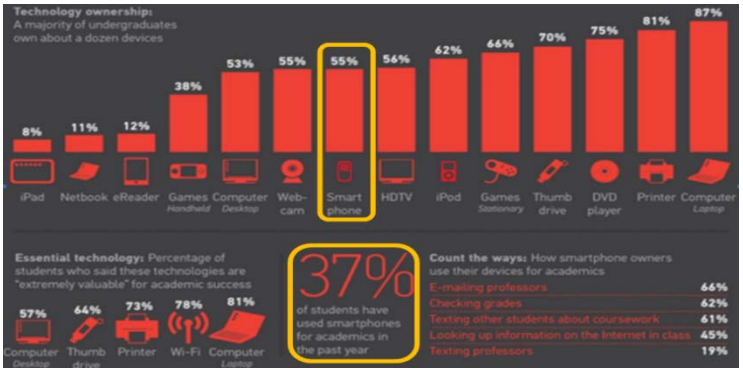


Fig. 4. Student perceptions of hardware use in higher and university education (ECAR, 2011)

There are already many mobile platforms on the market today, including Symbian, iPhone, Windows Mobile, BlackBerry, Java Mobile Edition, Linux Mobile and more. Android is the first environment that combines a truly open, free development platform based on Linux with a component-based architecture inspired by Internet services, automatic management of the application life cycle, high-quality graphics and sound, plus portability across a wide range of current and future hardware.

According to the students perspective presented above, in this paper, I present my experience, findings, and lessons learned from working with App Inventor - a friendly user platform to create easily Android applications.

App Inventor for the Android, as part of information consumer culture, is a blocks language [4] for mobile phones and a powerful visual and drag-and-drop tool. The software could be used to build mobile applications: game design, multimedia quizzes and guides, educational software, design complex applications that control robots or communicate with the web - with no programming experience required.

2. METHODOLOGY

There are a lot of players out there in the mobile development framework world (Fig. 5). This section points out the differences, strengths, and weaknesses in a few of the tools out there in order to understand the mobile development environment concepts.

	App Inventor	DroidDraw	Rhomobile	PhoneGap	Appcelerator	WebView	AML
			Mobile Platform Compatibility				
Android	Yes	Yes	Yes	Yes	Yes	Yes	Yes
iPhone/iPad	No	No	Yes	Yes	Yes	Yes	Planned
Windows Mobile 6	No	No	Yes	Maybe	No	Yes	Planned
Windows Phone 7	No	No	Maybe	Maybe	No	Yes	Planned
Palm	No	No	No	Yes	No	No	No
BlackBerry (RIM)	No	No	Yes	Yes	Planned	Weak	No
Symbian	No	No	Yes	Yes	No	No	No
			Development Environment OS Support				
Windows	Yes	Yes	Yes (no iPhone)	Yes (no iPhone)	Yes (no iPhone)	Yes (no iPhone)	Yes (no iPhone)
Mac	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Linux	Yes	Yes	Yes (no iPhone)	Yes (no iPhone)	Yes (no iPhone)	Yes (no iPhone)	Yes (no iPhone)

Fig. 5. The Android IDE and other platforms for develop Android apps

App Inventor has empowered students to create applications for android phones. This task was impossible before due to the numerous demands and requirements of the Android app creation process: knowledge of Java, comprehension of computer science principles, and familiarity with software development tools such as Eclipse, NetBeans, etc.

Each of the software development tools and environments for mobile devices are described below using a table format with a few different sections:

Strengths - Why is this a good tool?

- free, extremely easy to get started and use, IDE runs on any OS that supports Java;
- no coding necessary for layout or behavior and includes an device emulator and predefined “blocks”, no phone required.

Weaknesses - Why might it not be a good tool?

- create apps for only one platform, no other platforms supported, with pure visual design style that may be difficult for seasoned developers;
- does not output native source code, only native package, so no direct code creation or modification is possible.

Development Requirements - How do I get started building apps with it?

- create an account in order to access or to work at projects and install different other development kits.

Deployment Requirements - How do I get my apps onto a device?

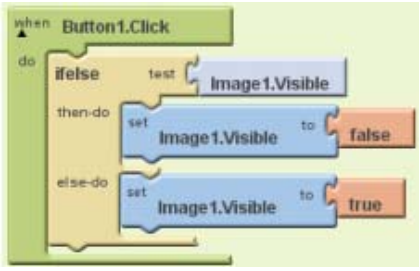
- build and sign apps packages for desired platforms, or nothing special required.

App Inventor is the newest tool out there, a friendly GUI for building and coding Android applications, great for anyone who want to learn programming concepts using their Android device. For other platforms or other types of developers, it's probably not the best choice. Since its conception (release in 2010), App Inventor has been used as a teaching tool in many schools and universities. It has gained popularity amongst educators and students around the world as a learning tool (having the characteristics from Fig. 6) for fundamental concepts in computer science, and more importantly, as a medium through which students can exercise creativity [5, 6].

	Strengths	Weaknesses	Development Requirements	Deployment Requirements
App Inventor	✓✓✓	✗✗	✗✗	✗
DroidDraw	✓	✗	✗	✗
Rhomobile	✓✓	✗✗	✗✗✗✗	✗
PhoneGap	✓✓✓	✗✗	✗✗✗	✗
Appcelerator	✓✓✓✓	✗✗	✗✗	✗✗
WebView	✓✓	✗✗✗	✗✗	-
AML	✓✓✓	✗✗	✗	✗

Fig. 6. The App Inventor main characteristics as an educational learning tool

App Inventor is first of all an educational learning tool, provided by Massachusetts Institute of Technology - Center for Mobile Learning, that allows (enables) users - almost everyone - to develop applications (apps) for Android-based smart mobile devices without writing a single line of code (or without any knowledge of programming languages - Fig. 7).




```
Button1.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if(Image1.isShown()) {
            Image1.setVisibility(ImageView.INVISIBLE);
        }
        else {
            Image1.setVisibility(ImageView.VISIBLE);
        }
    }
});
```

Fig. 7. Visual code vs Textual code for handling the event of a button clicked

App Inventor can also be considered as an Online Development Environment or web-based tool [6] developed jointly by MIT and Google Labs in which the App Inventor servers store all the work (**Error! Reference source not found.**). When the app is finished it can be packaged as an “Android application package” (.apk file) that can be shared around and installed on any Android devices [7].

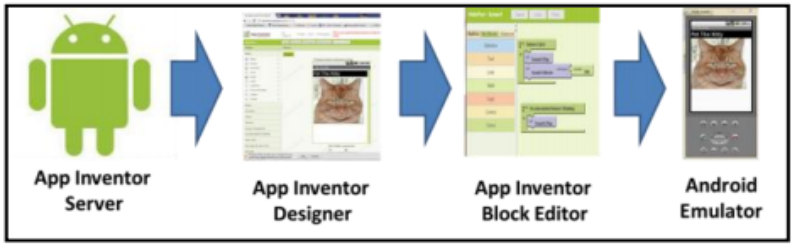


Fig. 8. App Inventor development environment and MIT logo

App Inventor is supported by a wide range of operating systems and web browsers, with these minimum specifications [8]:

- computer operating system requirements: Windows (Windows XP, Windows Vista, Windows 7+), Linux (Ubuntu 8+, Debian 5+) and Mac (Mac OS X 10.5+);
- web browser requirements: Internet Explorer 7.0+, Mozilla Firefox 3.6+, Google Chrome: 4.0+ and Apple Safari 5.0+.

Additionally App Inventor requires two pieces of software to be installed on computer: Java - to provide software libraries for App Inventor and App Inventor Setup - to provide client-side support for the web-based App Inventor development environment.

App Inventor is a mobile applications design tool consisted of three integrated major parts:

- ***Design View*** - for specifying the visual components of an application;
- ***Blocks Editor*** - for creating behaviors for the components and
- ***Android Device Emulator*** - for running (testing) the applications.

Programming in App Inventor is done with colorful blocks that are designed to snap together like puzzle pieces. The blocks are like words that, when snapped together, form sentences of instruction to your phone. The Design View (for version 1.2 - **Error! Reference source not found.**) is laid out in five basic columns from left to right: Palette, Viewer, Components, Media and Properties.

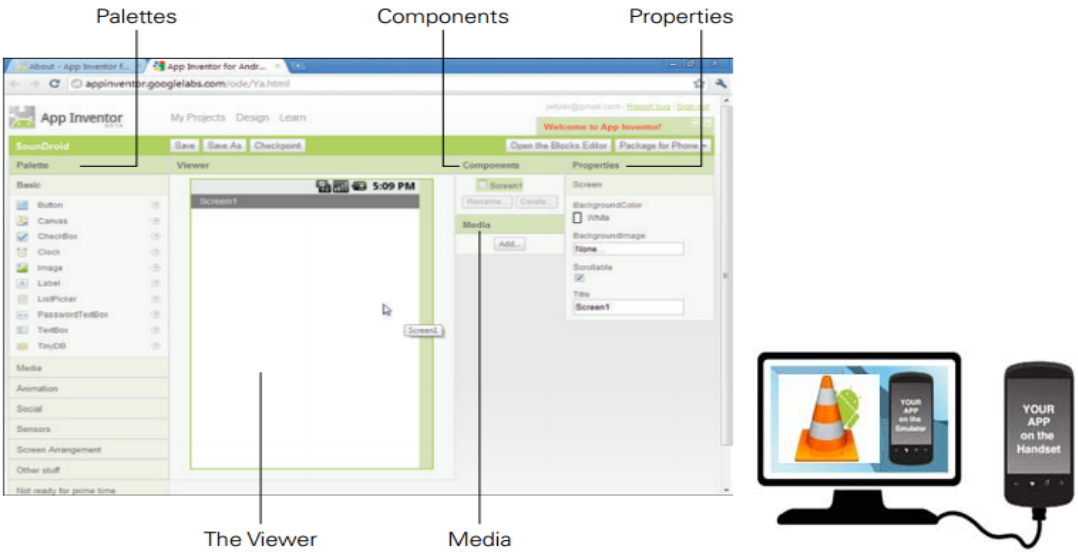


Fig. 9. App Inventor Designer (a) and the debugging perspective for an Android application using an emulator or an Android handset (b)

The Component Designer is a fairly typical WYSIWYG (what you see is what you get) tool that provides the components for designing user interfaces, such as Basic, Media, Animation, Social, Sensor, Arrangement and Other. Each component can have methods, events and properties [9, 10].

The Blocks Editor (Fig. 10) is used to assign behavior to the components (e.g. to indicate what events should cause action in the app and what the resulting algorithm is after the event occurs). It allows a user to set of different properties, also it can make a reference to a component via a variable (argument) and manipulate dynamically.

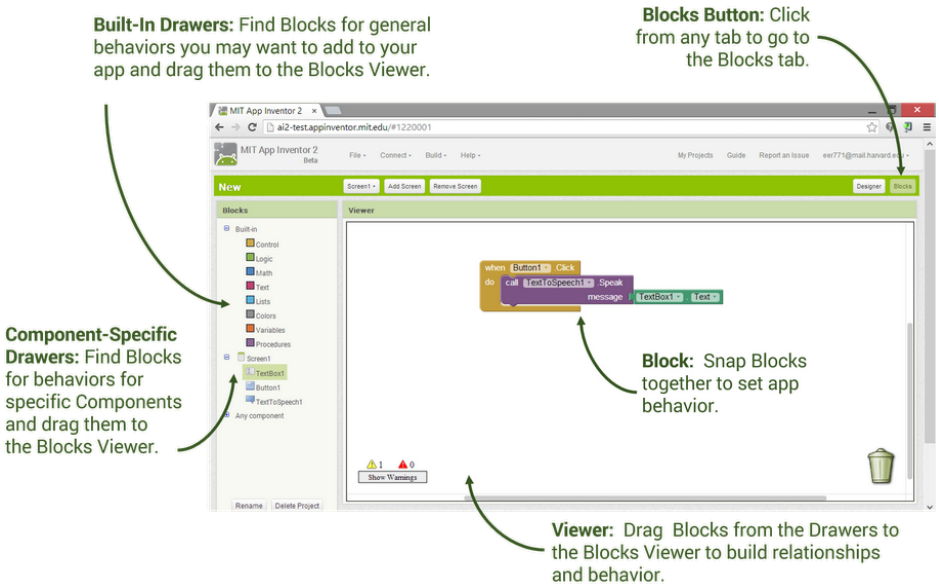


Fig. 10. The Blocks Editor for an apps created with App Inventor v. 2.0

There are seven block categories, not including the My Blocks section: ***Definition Blocks***, ***Text Blocks***, ***List Blocks***, ***Math Blocks***, ***Logic Blocks***, ***Control Blocks*** and ***Color Block***.

One of the most important tools included in the Android SDK is emulator.exe which launches the Android Emulator [9]. The Android Emulator is used to run the applications in a pseudo-Android environment. At the same time, the emulator help to prototype, develop and test Android applications without using a physical device.

The Android emulator mimics all of the hardware and software features of a typical mobile device, except that it cannot place actual phone calls. The Android Device Emulator provides a variety of navigation and control keys [10], which can be used to generate events for the applications. It also provides a screen in which the applications are displayed, together with any other active Android applications.

3. RESULTS AND DISCUSSION

Education has evolved considerably over the last 50 years and the industry is now faced with a series of significant trends (Fig. 11) that offer the potential to create dramatic new opportunities for effective learning.

Fig. 11. The evolution steps of the informatics education

The complexity of the business landscape in actual information society has increased demands for continuous learning, with executives demanding access to learning opportunities any time, any place and anywhere. Technology offers the opportunity to empower the learner and overcome barriers to learning. Furthermore, it empowers education providers to serve up learning in multiple formats to suit individual learning styles.

Over the past five years advances in the sophistication of mobile technologies have catalyzed the way in which mobile learning is being adopted and used in educational contexts and it has evolved into much more than "e-learning with a phone" [11, 12].

The implications of mobile learning are far-reaching, and its potential effect on all education is profound; as mobile learning capabilities continue to expand new forms of learning will continue to evolve and the next few years will see a period of rapid growth for mobile learning, with evolutionary rather than revolutionary changes.

Mobile and wireless devices, especially mobile phones, have recently become increasingly common among young students. This provides new possibilities, opportunities and challenges for education (Fig. 12), and especially for mathematics education.

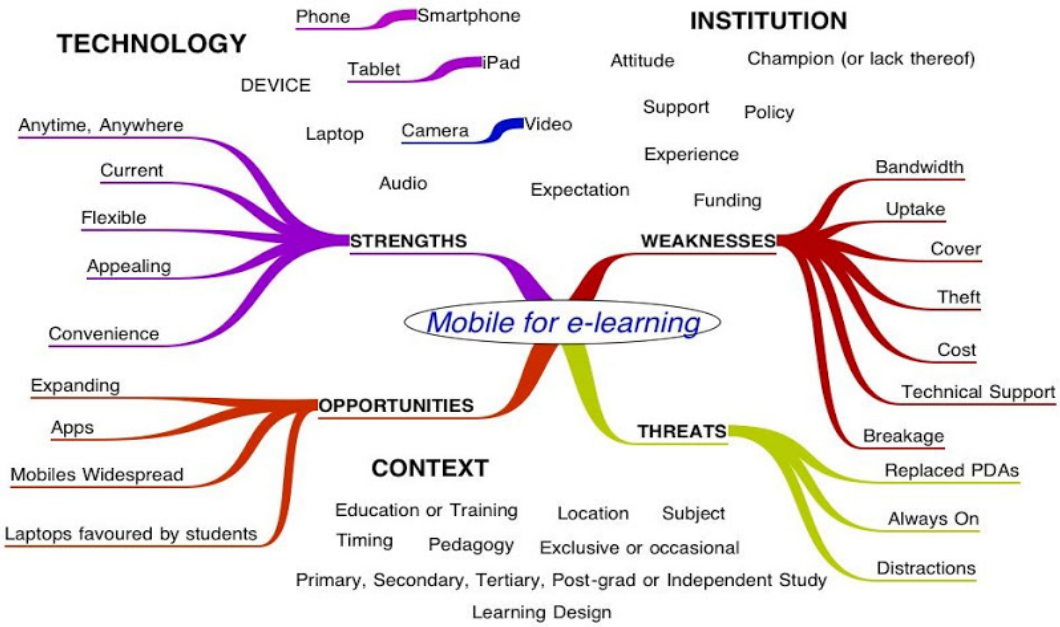


Fig. 12. A brief SWOT analyze for mLearning

Recent studies examined the use of mobile phones in mathematics learning among teachers suggest that we are at the beginning of a new era for mobile phone integration in the mathematics classroom. According to this trend the MathApp Works! demonstrate that is possible to give mobile education using an integrated development environment [13, 14].

App Inventor has several strengths: is free, easy to get started (web app) and use; IDE runs on any OS that supports Java; no coding necessary for layout or behavior, includes an Android device emulator, no phone required, includes predefined “blocks” for accessing web and other services; provides limited visual live debugging and testing etc). Several of its weaknesses are: it is just for Android; pure visual design style that may be difficult for seasoned developers; does not export Java code, and no java modification is possible.

However, App Inventor will not fit the programmers full expectations because is too restricted and simple, so for creating complex apps will be more difficult to use comparative to an IDE such as NetBeans or Eclipse. The MathApp Works! was difficult to made according its complexity!, the mobile educational software can solve many applied mathematics problems. The present paper is about an application with App Inventor for Android platform. It is an educational system for applied mathematics, necessary for those who wish to learn mathematics in an interactive way or just to improve their math skills.

The new software - MathApp Works! (Fig. 13) - have the following versions:

- MathApp Works! - Standard (for V-VIII grades) - 5 different apps: *Calculator*, *Formulas*, *Equations Solver*, *Geometry Solver* and *Unit Converter*.
- MathApp Works! - Advanced (IX-XII grades) - 6 different apps: *Glossary*, *Financial Calculus*, *Statistics*, *Matrices & Determinants*, *Derivates* and *Integrals*.
- MathApp Works! - Professional (university level) - 3 different apps: *Transport Problem Solver*, *Simplex Solver* and *Miscellanea.*

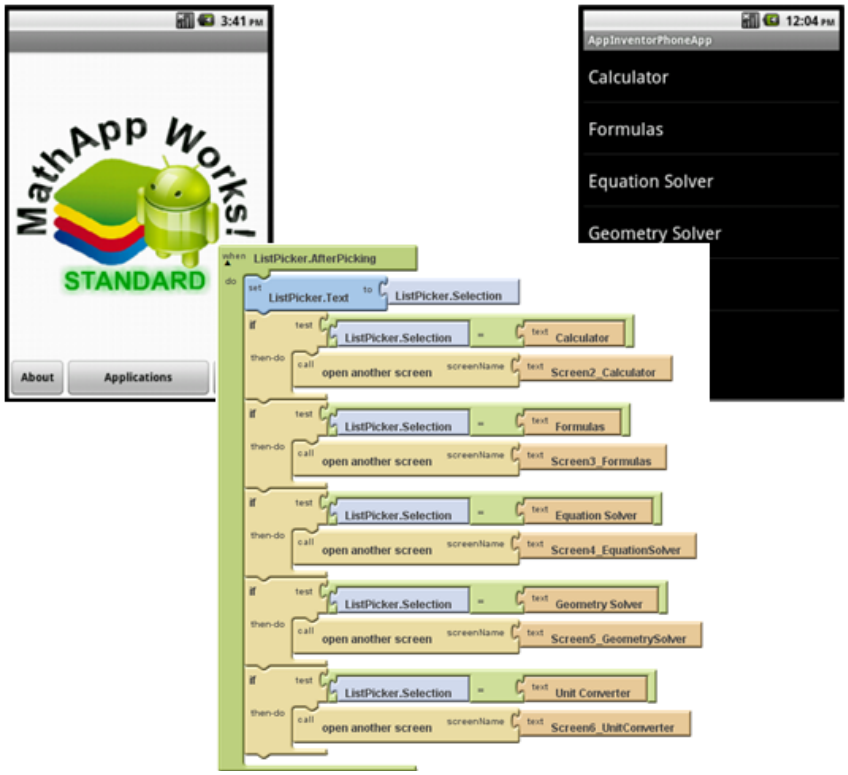
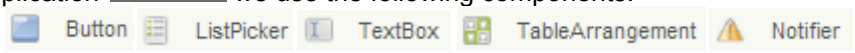


Fig. 13. MathApp Works! Standard version

The **Calculator** (Fig. 14) from **MathApp Works! - Standard** version includes options for solving exercises connected to algebra (**+**, **-**, *****, **/**, **log**, **%**, **!**), calculus (**e^x**, **π^x**, **10^x**, $\sqrt{}$, $\sqrt[3]{}$, $\sqrt[4]{}$, **abs**, **round**, **floor**, **ceiling**), trigonometry (**sin()**, **cos()**, **tan()**, **asin()**, **acos()**, **atan()** etc).

For the application Calculator we use the following components:



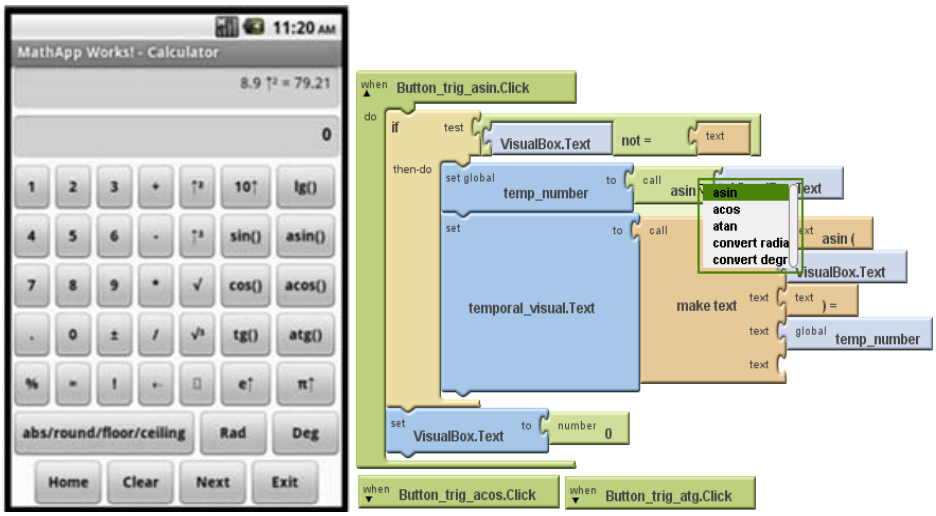
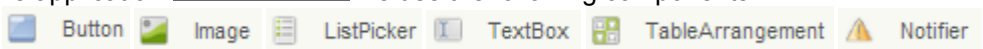


Fig. 14. MathApp Works! Standard version - calculator design view

Another interesting application is **Geometry Solver** (Fig. 15) from **MathApp Works! - Standard** version has implementated different options for representation and solving 2D figures (e.g. ***square***, ***rectangle***, ***rhombus***, ***parallelogram***, ***triangle***, ***trapezoid***, ***circle***, ***ellipse***, ***sphere***, ***cone***, ***cube***, etc).

For the application Geometry Solver we use the following components:



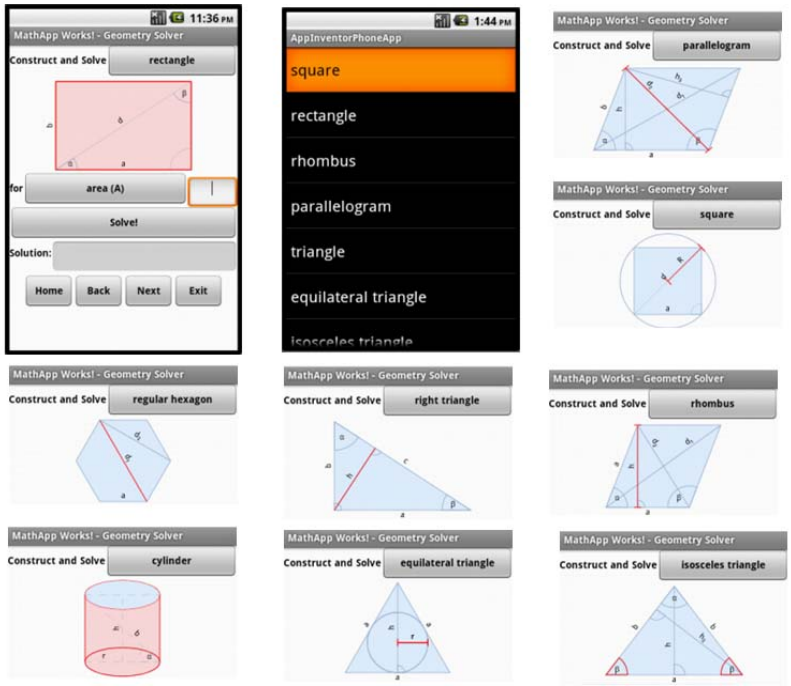
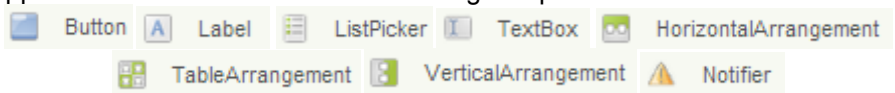


Fig. 15. MathApp Works! Standard version - Geometry Solver design view

The **Derivates** (Fig. 16) from **MathApp Works! - Advanced** version includes different options (***(x)'***, ***(ax+b)'***, ***(ax^2+bx+c)'***, ***sin(x)'***, ***asin(x)'***, ***(e^x)'*** etc) to easily access the main differential calculus.

For the application Derivates we use the following components:



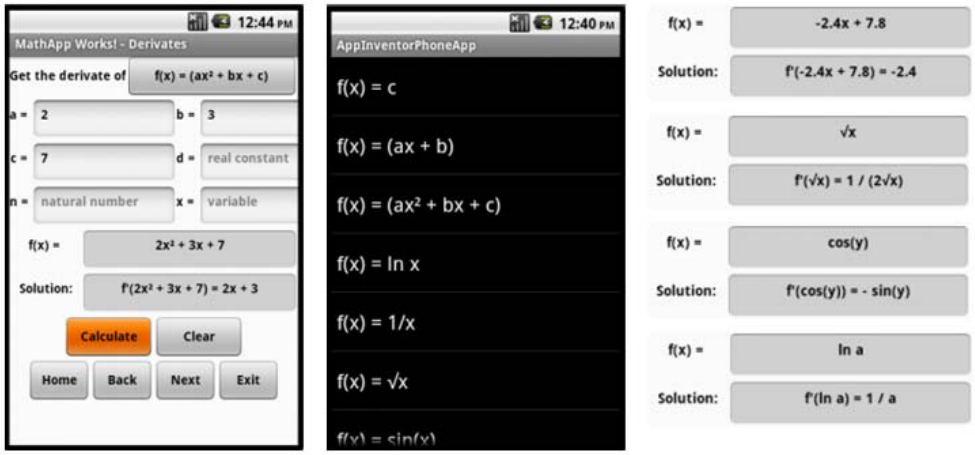


Fig. 16. MathApp Works! Standard version - Derivates design view

Mobile learning is an emerging field that requires more research and development if the potential of mobile learning to education and training is to be realized. Despite the ubiquity of mobile phones in every aspect of our daily lives, the use of these devices in education, to teach and learn mathematics in particular, is still new.

4. CONCLUSION

The current paper illustrates a few App Inventor-based applications for students, integrated as an educational-platform named MathApp Works!. The MathApp Works! platform-applications are based on a various palette of components: basic, media, animation, social, sensor, screen arrangement etc and blocks, as definition, text, list, math, logic, control, color, facilitating the project design and proper functionality. The newly introduced MathApp Works! as an applied mathematics mobile software tool for learning the basis of algebra, geometry, trigonometry, differential and integral calculus etc provide a user friendly educational interface and was successfully tested on emulator and on several Android devices, and fit perfectly to the actual educational requirements of the students.

COMPETING INTERESTS

Authors have declared that no competing interests exist.

AUTHORS' CONTRIBUTIONS

Bogdan CIORUȚA designed the study, performed the application analysis, and wrote the first draft of the manuscript. Mirela COMAN and Alin-Andrei CIORUȚA managed the analyses of the present study, and managed the literature searches. All authors read and approved the final manuscript.

REFERENCES

1. Kloss Jörg H., Android Apps with App Inventor: the fast and easy way to build Android apps, Pearson Education, Inc., USA, 2012.

2. Daher W., Building mathematical knowledge in an authentic mobile phone environment, Australasian Journal of Educational Technology, 26(1), pg. 85-104, 2010

3. Wolber D., Abelson H., Spertus E., Looney L., App Inventor - create your own Android apps, O'Reilly Media Inc., Canada, 2011.

4. ***, http://appinventor.mit.edu

5. ***, www.appinventor.org

6. Tyler J., App Inventor for Android: build your own apps - no experience required!, John Wiley & Sons, Ltd., USA, 2011.

7. Kim Hak. J., Modell J., Mobile app design tool for smartphones: a tutorial, International Journal of Soft Computing and Engineering (IJSCE), vol. 2, pg. 237-243, 2012.

8. Sherar J., Make your own android apps: your unofficial intro to MIT App Inventor, 2012.

9. DiMarzio J.F., Android - a programmer's guide, The McGraw-Hill Companies, USA, 2008

10. Deitel & Associates, Android for Programmers: An App-Driven Approach, Pearson Education Inc., USA, 2012

11. Schofield C.P., West T., Taylor E., Going mobile in executive education, UNICON, UK - www.ashridge.org.uk/MobileResearch, 2011.

12. Cioruța B., Coman M., Create your own applied mathematics software for Android mobile device, Scholar`s Press, OmniScriptum Publishing Group, Beau Basin, Mauritius, 2018, pp. 68, 978-620-2-31631-6

13. Cioruța B., Coman M., Cioruța A., The idea of implementing a mathematics platform for Android devices with help of App Inventor, International Conference "Scientific Research and Education in the Air Force" (AFASES®), 22-24 mai 2014, Braşov, pg. 293-296 www.afahc.ro/.../afases2014/Cioruta_Coman.pdf

14. Cioruța B., MathAppWorks! – an applied mathematics educational platform-software for Android mobile devices, Salonul Internațional al Inovării și Cercetării Științifice Studențești (Cadet INOVA®), "Nicolae Bălcescu" Land Forces Academy, Sibiu cadetinova.ro, Buletin științific supliment - Catalogul oficial al Salonului "Cadet INOVA" nr. 2/2017 - Cercetări și inovații în viziunea tinerilor cercetători, Ed. Academiei Forțelor Terestre "Nicolae Bălcescu", Sibiu, ISSN 2501-3157, ISSN-L 2501-3157 - Exhibition papers, pg. 155-158

DEFINITIONS, ACRONYMS, ABBREVIATIONS

AIA: App Inventor for Android

app(s): application(s)

AVD: Android Virtual Device (Emulator)

ECAR: Educause Center for Applied Research

e-learning: learning utilizing electronic technologies to access educational curriculum outside of a traditional classroom

MIT: Massachusetts Institute of Technology

m-learning: learning utilizing exclusively mobile technologies and afferent devices to access educational curriculum outside of a traditional classroom

IDE: Integrated development environment