

A Cross-Platform Mobile Application for Learning Programming Basics

<https://doi.org/10.3991/ijim.v12i7.9442>

Alaeeddine Yassine^(✉), Mohammed Berrada, Ahmed Tahiri, Driss Chenouni
Sidi Mohamed Ben Abdellah University, Fes, Morocco
alaeeyassine@gmail.com

Abstract—This paper presents and discusses a new mobile application for teaching and learning programming basics. The purpose is to help students acquire competencies while having fun and using their own devices. The mobile application was designed according to cross-platform approach to reach the broadest possible audience of learners, saving time and effort of development and maintenance. The code is fully shared between mobile platforms (iOS, Android and Windows Phone) allowing learners to install the application on any device. The core application is based on a multi-agent system to make the application interactive, flexible and dynamic and provide students with tailored instructions. A prototype is presented showing the main features of the application.

Keywords—Cross-platform, Mobile Application, Programming Basics, multi-agent system, Learning Outcomes, m-learning, Xamarin.Forms, self-learning.

1 Introduction

Mobile devices have evolved in the last decade enabling new ways of learning called m-learning. The purpose of mobile applications for learning is to leverage self-learning using technology that is already available for many students. Furthermore, the 21st century learners are digital natives, most of them spend countless hours glued to mobile and video game consoles. They heavily rely on smartphones for communication and entertainment purposes[1]. This is important to take into consideration while designing application for learning.

Mobile learning is a promising way for delivering content. It provides student access to courses, anytime and anywhere. It also promotes self-pacing by providing students with updates just-in-time. However, m-learning should make the best use of limited screen space: GUI (Graphical User Interface) takes greater importance than ever [2].

To this end, teaching should be changing shape in response to the light speed growth in the area of educational technology and many challenges should be addressed while designing m-learning applications: the target audience, features supported by the device, the target operating system, network availability and screen size. Another concern is instructional design approach. For instance, m-learning requires small chunks of content that could be readable on small devices.

Building Mobile applications for learning is a challenging research area. From the technical point of view, it requires developers to learn many languages and APIs for the various platforms. As stated by [3], one challenge facing developers is lack of support for automated migration across platforms. Developers have to rewrite the code to match the target platform. One popular solution is to develop a responsive web application for mobile usage and minimize the effort for making native code for each platform. Still, this solution has many limitations. For instance, it requires constant access to Internet which is not possible for all learners.

In this paper, we present and discuss a new Cross-platform Mobile Application for Learning Programming Basics. Programming is a multidisciplinary course. It's a part of most engineering curriculums. It is based on the assumption that students need to acquire 21st century skills such as critical thinking and problem solving skills. However, learning how to program is perceived by many students as hard subject to master. This could be explained by a misguided perception: students think writing a code requires a complex cognitive effort as well as deep understanding of abstract concepts. At this stage, we posit that m-learning could foster student's motivation and engagement for learning by providing them with challenging and contextualized learning. Learners could be able to learn at their own pace, anywhere using any device. The remainder of this article is organized as follows: section 2 reports and discusses related works. Section 3 introduces Cross-platform mobile development approach. Section 4, presents the mobile application architecture and pedagogical foundations. Section 5 discusses the application and evaluate its effectiveness. Finally, Section 6 presents our conclusions and outlines some perspectives.

2 Related works

Learning how to program is often recognized as a frustrating activity. An analysis of related works revealed that many mobile applications are used for learning programming basics. These m-learning applications are divided into three categories: the first one is devoted to teach and learn a programming language as a whole (C, Java, JavaScript, Ruby, etc.). The second category is focused on a particular concept considered as the most difficult concept to teach and learn. Finally, the third category aimed at teaching fundamental concepts of programming such as variables, loops and conditions and are not related to a programming language in particular.

For example, Light-Bot [4] is a programming puzzle game available for many platforms: iOS, Android as well as a web version. The purpose of Light-Bot is to teach programming concepts such as loops and conditions. Another application is Kodable[5], a mobile application for teaching logic and fundamental concepts of programming. The scenario is about a family who needs to explore many mazes and collect coins. The player must drag and drop commands to advance in the game. For both applications, the focus is on algorithmic skills rather than the syntax of a particular programming language.

Move the turtle [6] is a mobile game that teaches how to plan complex operations based on simple steps. The players use predefined commands to build a program and

help the turtle get diamonds. The application is available on iOS for novice programmers.

Robozzle is a robot-programming game in which players are asked to construct a program using a set of predefined commands. When executed, a program is performing and picking up stars in a grid environment. The application is available for iOS, Android and Web.

We postulate that learning a programming language need many hours of practice, students get confident only by applying concept to solve a variety of problems. After that, students would be able to learner any new programming language. Some researchers pointed out that learning a programming language could be fostered through practical situation [7] using problem solving educational approach [8]. Another common problem revealed by many teachers is: the difference between the natural language and a programming language. If students are not familiar with English then they couldn't easily understand the meaning of instructions they use while programming. Also, teachers use an ordinary way to teach abstract concepts. Students taking a programming course are struggling with high-level abstractions and complex instructions, which make the code difficult to understand and master. The used method usually doesn't fit the needs of students who seem to not get the logic behind the code. Another approach to foster learning how to program could be individualized feedback. For instance, [9] Proposes an Intelligent Tutoring System for learning C# language to help student better learn programming languages. The purpose is to make the learning experience very pleasing by providing students with tailored instructions.

Using mobile devices for learning is engaging, motivating and provides hands-on means to practice skills[10]. Still, it implies some design implications and challenges. The study[11], posits that using mobile device for learning leverages self-initiated learning tasks. Furthermore, one aspect to take into consideration is prompt feedback to learners to foster their motivation.

To sum up, many studies suggest to use technology to foster student's motivation for learning. M-learning is considered as a type of "ubiquitous learning" to provide learners with a better education experience by supporting context-aware features[12].

3 Mobile development approach

Mobile development approaches could be divided into 4 categories: (1) Native Mobile Applications, (2) Hybrid Mobile Applications (3) Mobile Web Applications and (3) Cross-Platform Mobile Applications.

Native mobile applications are tailored to work on a specific platform, it uses specific programming language and Integrated Development Environments (IDE) provided by the platform vendor. They are installed on the smartphone from the platform's App store. One advantage of native approach is providing fast performance and full access to native device features and capabilities such as its camera and GPS and address book, calendar, Bluetooth, etc. Still, the development cost is very high and time consuming. Developers have to create various applications to support all the target

platforms. The following table summarizes prerequisites for developing Native Mobile Applications for most used platforms Android, iOS and Windows Phone.

Table 1. Native mobile development approach prerequisites

	Programming language	IDE	App store	GUI
iOS	Objective-C	XCode	iTunes Apps Store	Cocoa Touch
Windows Phone	C#	Visual Studio	Windows Phone Store	XAML
Android	Java	Eclipse	Google Play Store	XML files

Mobile Web Applications are web applications designed for mobile users. No installation or updates are required. The application is hosted in a remote server as web site and requires browser to display content. The application is accessed through any mobile or computer. Still, a good network connection is needed for loading content and interacting with users because the application is not installed on the device. Another drawback is limited access to device hardware, the browser doesn't allow access to features like device storage, GPS, camera, etc.

Hybrid development falls in the middle of the web and native approach. The purpose of Hybrid Mobile Applications is to provide access to device features by combining native and web advantages. The application is developed using technologies such as HTML5 and JavaScript Frameworks and then hosted inside a native container so to have access to all device capabilities. One advantage of hybrid development is that the same application works on all platforms also if an update is required, it is always seamless for users and installation step is done the same way as native applications.

Another approach has been gaining popularity over the last few years: cross-platform development. The purpose is to write a common code that will be deployed on multiple platforms. This approach is based on the assumption that developing multiple versions of the same application is time consuming and requires many competencies. Every vendor proposes its own programming language, tools and environment. Cross-platform approach decreases cost of development and maintenance by avoiding code redundancy. The main advantage of this approach, a native app is generated from the same shared codebase for each target platform vendor[13]. Table below summarizes the main mobile development approaches.

Table 2. Comparison of mobile development approaches

	Limitations	Maintenance	Performance	Cost	Ergonomic GUI
Native	+	-	+	-	+
Hybride / Mobile	-	+	-	+	+
Cross-Plateforme	+	+	+	+	-

To sum up, using web technologies the application runs in web browser or in a browser-view wrapped into a native application container in the case of hybrid approach. It's an alternative method to native but have some limitations.

Developing mobile application can be achieved through several approaches. In order to develop a cost-saving solution and make the application available for different devices and platforms we excluded native approach because it's time consuming and require developers to master many programming languages, tools and environments. Cross-platform development approach fulfils the requirements in order to reach the broadest possible audience of learners, saving time and effort of development and maintenance. The result is a native application that could be published in the App store.

4 Mobile Application Architecture

In this section we present and discuss the cross-platform mobile application approach used to develop the application as well as system architecture and the instructional design model.

4.1 Cross-Platform Mobile App Development

For developing the mobile application we used Xamarin.Forms Framework[14]. It's a Framework for developing native mobile applications. The advantage is to maximize code sharing as well as native user interface and native performance [15]. For instance, we developed two blocks of code: the code that entails the application logic and the code to construct the Graphical User Interface as depicted in Fig 1. The code is fully shared between mobile platforms (iOS, Android and Windows Phone). Using Xamarin.Forms framework, it is possible to produce native Android, iOS and Windows application in C#[16]. According to[17], Xamarin compiles to native code and achieves better performance and native UI compared to its competitors[17].

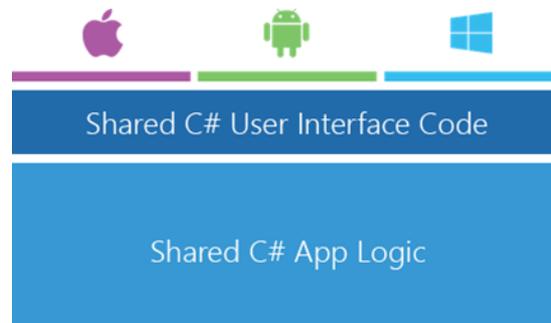


Fig. 1. Portable Class Library Architecture

We developed A Portable Class Library (PCL). It's a type of Visual studio project that can be used to develop cross-platforms applications such as Xamarin.iOS and Xamarin.Android, as well as WPF (Windows Desktop). The project contains core application which is a common code across all the target platforms with specific customization for every platform. As to date, 90% of the code is sharable. If an updated or correction is expected, it's done on the common code once.

4.2 Application Logic

A multi-agent system (M.A.S) is important in our context. Presenting the application in a static way wouldn't be beneficial for all learners who acquire knowledge according to their own pace and might be interested in different concepts at different levels. While some learners could be progressing, others could be struggling to complete a level and need further attention. The integration of MAS provides players with automated instructions based on their performance and progress.

Our m-learning application is based on a multi-agent system. The system includes various intelligent agents interacting to provide students with the best learning scenario. These agents imitate the decisions of human tutor who analyzes student progress and provides him/her with best instructions to progress. The purpose is to make the application flexible and dynamic, the learner is guided throughout the learning process based on decisions made by the system.

The MAS system includes 4 agents: "Behavior Agent", "Performance Diagnostic Agent"; "Instructional System Design Agent" and "Helper". Each Agent performs a specific task, shares information and communicates with other agents. The functioning of each agent is as follows:

- **Behavior Agent:** this agent collects data about student's behavior in the application. For instance, the agent collects information about time elapsed, the interaction of the learner with different objects of the screen and steps pursued by the learner (actions) to solve an exercise.
- **Performance Diagnostic Agent:** this agent is charged to compare the student's current state of mastery with the desired one using data provided by behavior agent.
- **Instructional System Design Agent:** this agent plays the role of a human tutor; it makes decisions about what will be presented to learner according to performance diagnostic agent. The application is divided into several units and pages; each page contains many instructions and addresses one or many learning goals. The instructional design agent decides what environment will be presented with specific learning goals and learning tasks.
- **Helper:** this agent is activated by the player or the performance diagnostic agent to provide students with feedback. In one hand, the helper is activated by the player if he/she asks explicitly for help. The helper should work intelligently to provide the learner with help only when it's justified. For instance, help is not activated unless the learner has been working for a while on the task. This information is given by the behavior agent. In the other hand, the agent is activated implicitly when informed by the behavior agent that the learner is taking too long to finish a task. The helper agent sends a notification to the Instructional System Design Agent who decides what will be presented to the learner according to the situation reported by the behavior agent.
- For security reasons, the communication between the application interface and the MAS system is done through HTTP requests. The mechanism of saving data uses a checkpoint system. For instance, when reaching a certain point of a level (sensors), the saving function is automatically triggered, application attributes are then sent as

variables wrapped in JSON file. These variables are collected and used by the MAS system.

The architecture of our application based on MAS and the communication between agents is depicted in Figure 2.

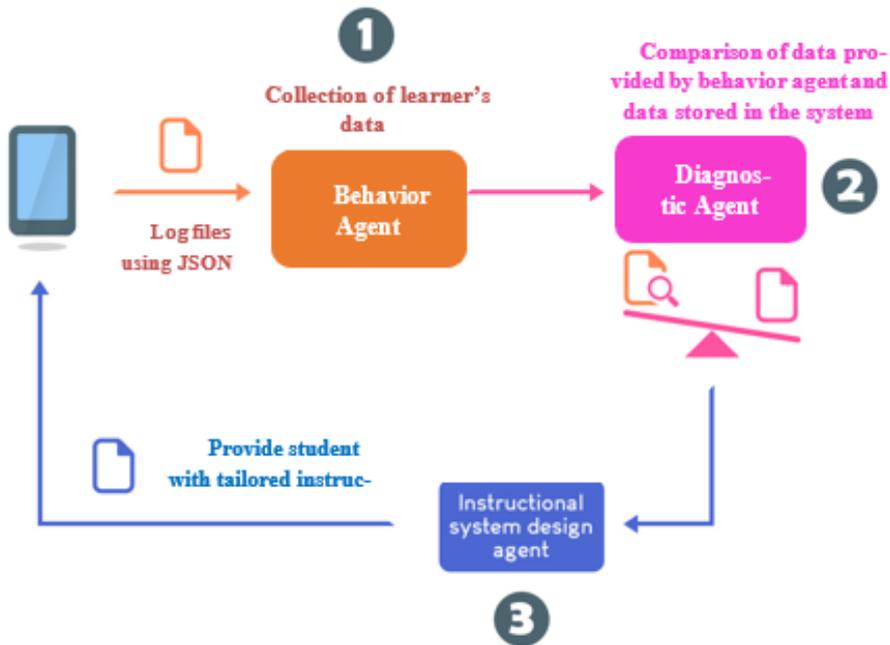


Fig. 2. MAS System to support tailored feedback

Using a MAS architecture, the application works intelligently, it provides students with hints and adjusts the difficulty to the current level of the learner. These agents need data about student's level and couldn't be performing at the beginning. For instance, while reaching a checkpoint or replaying a level that the learner failed to complete, he/she will be given the opportunity to experience new challenges with a new environment according to intelligent agents functioning.

4.3 Instructional Design

For managing and storing knowledge we used a domain ontology. We build the instructional model using Protégé Editor[18]. The visualization is ensured through VOWL Plugin[19]. The main benefits of using an ontology is to provide Communication between systems, reuse and organization of knowledge[20]. Domain ontology is a standard-based way for knowledge sharing. In our case, intelligent agents use the ontology to publish and share data.

The domain ontology is published online with the aim to provide mobile application access to data. One benefits of using ontology is that many applications might share the

data. For example, learning management system (LMS) might use the question bank in an assessment module. It also facilitates the communication and knowledge sharing between agents.

For the instructional design model. The courseware is divided into several units. Each unit encompasses many pages and addresses many competencies. Each page uses one or many learning resources adapted to mobile device. The domain ontology contains a question bank. Each question belongs to a category and is related to a specific learning unit. It also implies a limited number of attempts after that a feedback is presented to learner. It is also possible to provide students with hints while answering a question.

In order to provide students with automatic tailored feedback, the domain ontology stores questions as well as the correct answer(s). This means that text answer are not allowed since it is difficult for agents to perform automatic text processing. This requires choosing correct answer(s) from many items. For each item, we suggest to use some parameters such as: difficulty, description, feedback and isCorrect which is set to true if the item is a correct answer to a question. This structure has the potential to help agents randomly picks questions. Furthermore, the ontology is independent and can be used by tutors to load questions and new content.

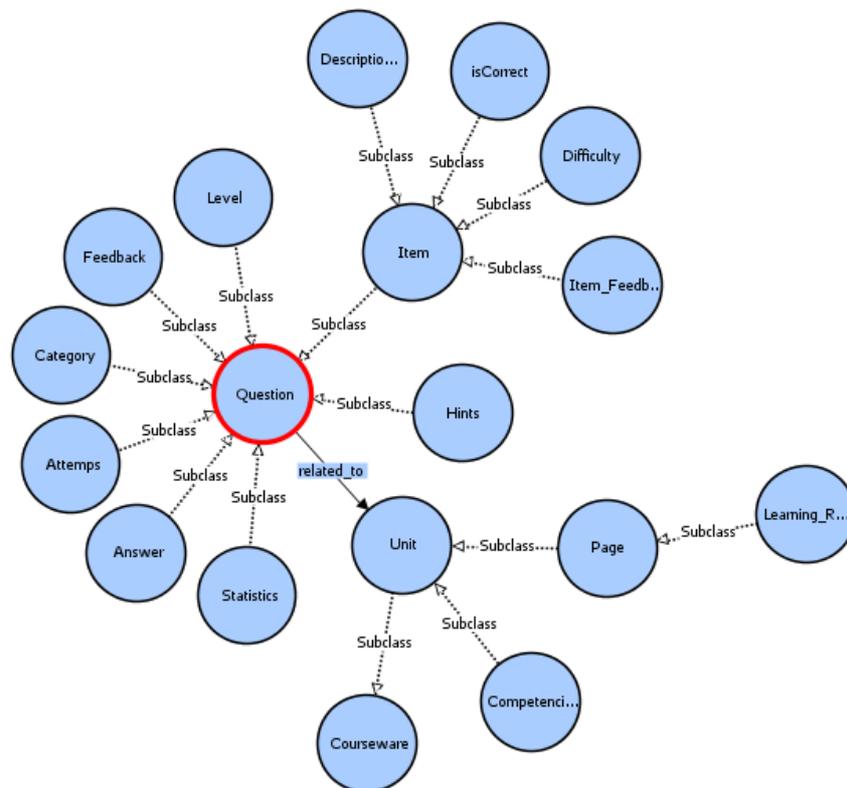


Fig. 3. Instructional Design Model

5 Mobile application for learning programming basics

The main objective of this project is to implement a cross-platform mobile application for learning C programming language basics. We developed a prototype of a mobile application using C# programming language and visual studio IDE. The application is based on Xamarin.Forms in order to make it available for all platforms. The target audience is students who want to learn programming basics.

Specifically, the application aims to the following:

- Enhancing the learning experience by introducing Ubiquitous Learning Environments and technology.
- Designing QCM that can be graded automatically without the intervention of teachers.
- Providing students with individualized feedback. Learners are notified of their progress.
- Design pleasant Graphical User Interface.
- Make the application available for all platform vendors.

The mobile application entails a programming basics courseware. Learners might log-in into the application using their Facebook or Google profile. The application is structured in levels with increasing level of complexity. It is based on catchy scenario to promote curiosity and adventure.

The scenario is about the opposition between two rival camps: globes and viruses. The viruses are a constant threat and the learner is asked to destroy them in a series of knowledge tests. The heroes represent the entities of the c programming language, namely the variables or the pointers. The choice of the nature of the characters has its source in our desire to concretize an abstract theoretical knowledge to facilitate its understanding and its assimilation. Indeed, representing a variable through the features of an animated character with a role to play allows the player to better understand how it works. Thus, the learner embodies the role different heroes and fights viruses with different characteristics. We thought of a story that could have an effect on learner's immersion. By having a mission to accomplish throughout the application, the learner should feel as the one in control of the situation.

First levels provides players with basic concepts that students should remember. While progressing, the learner is also developing problem solving and analytical skills. These skills can be used to solve a variety of problems. The application starts with basic concepts that all students should remember and smoothly moves to help learner solve complex programming problems

Each level encompasses many pages designed in carousel layout as depicted in Fig 4. The learner is invited navigate between the levels and access to content to learn basics of programming before playing a quiz-game (Fig.5). After completing a specific level the next one is automatically enabled. The purpose is to help the learner acquire a competency in a step by step process. Student's behavior and answers are collected by the behavior agent and sent for analysis in order to provide student with a broader understanding of the study material.

Interactivity is insured by maintaining a dialogue with the learners. The pages are animated and the learner might interact with different objects. The application also leverages risk-taking. For instance, players are won't be afraid of making mistakes. When they fail, they can start over from the last-saved. Despite the probability of failure, the application provides the learner with the ability to try again at will.

Knowing that, there are various types of learners with their own objectives and paces, it's important to enables a more personalized learning experience[21]. In our application, an individualized feedback is ensured through the implementation of a Multi-Agent System in which many intelligent agents collaborate in order to provide students with the best feedback throughout the student's progress.

For about information, it is delivered using two ways: information is shown under JIT concept (Just-in-time) to make immediate use of it in the right context, or On-demand access when the user asks explicitly for it. For instance, the help icon, which is available in all scenes, could be used on demand when the learner gets stuck. It provides him/her with hints to complete the current learning goal. The images bellow show some screenshots of the prototype we developed.

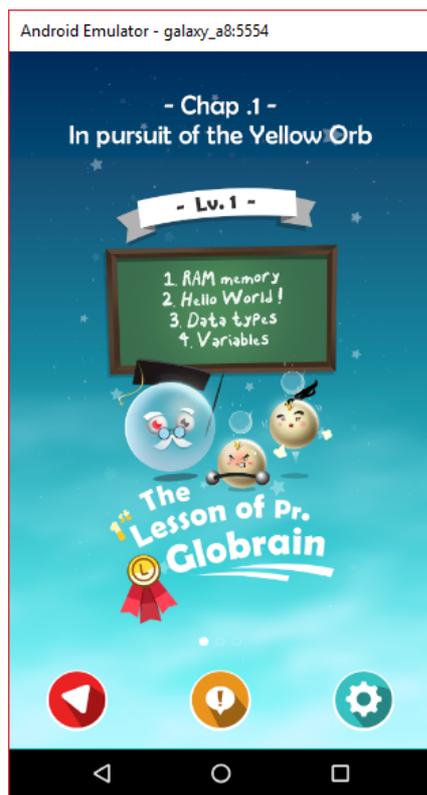


Fig. 4. Screenshot of Level 1

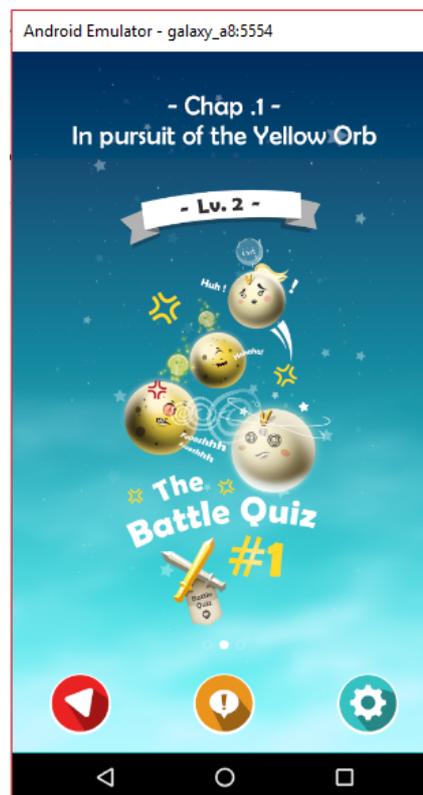


Fig. 5. Screenshot of the Battle Quiz



Fig. 6. Screenshot of score



Fig. 7. Chapter 1 presentation

6 Conclusion

Developing Native Mobile Application requires developers to carefully choose approach, framework and tools. One solution could be to develop a native application for each mobile platform. For example, an application that would work on Android should use Java as Programming Language or Objective C for iOS platforms. Another solution could be to develop the application as a responsive web application using HTML5 so that the User interface would fit on any screen due to Responsive Design principle. This solution saves valuable time but doesn't provide developers access to native options because it runs inside the mobile browser and doesn't have the access to full features of the mobile. It's also worth mentioning that availability of the content depends on internet connection

The purpose of using a Cross-Platform approach is to target as many students as possible by providing the application for different platforms and reaching the broadest possible audience. One advantage of mobile learning is that many learners own a device

and carry it wherever they go. However, designing learning activities for mobile devices requires reliance on instructional design that fits this kind of technology. M-learning enables learning outside the classroom at any time. As future work we intend to make the application available on App stores and propose a multiplayer mode in order to make the application more challenging and to leverage autonomous learning.

7 References

- [1] R. Alhassan, 'Mobile Learning as a Method of Ubiquitous Learning: Students' Attitudes, Readiness, and Possible Barriers to Implementation in Higher Education', *Journal of Education and Learning*, vol. 5, no. 1, 2016. <https://doi.org/10.5539/jel.v5n1p176>
- [2] A. I. Wasserman, 'Software engineering issues for mobile application development', *ACM Transactions on Information Systems*, pp. 1–4, 2010.
- [3] M. E. Joorabchi, A. Mesbah, and P. Kruchten, 'Real challenges in mobile app development', in *International Symposium on Empirical Software Engineering and Measurement*, 2013, pp. 15–24. <https://doi.org/10.1109/ESEM.2013.9>
- [4] 'LightBot'. [Online]. Available: <http://lightbot.com/>. [Accessed: 10-Aug-2018].
- [5] Kodable, 'Programming for Kids | Kodable'. [Online]. Available: <https://www.kodable.com/>. [Accessed: 12-Aug-2017].
- [6] Movetheturtle, 'Move the Turtle - Programming for Kids on the iPhone and iPad'. [Online]. Available: <http://movetheturtle.com/>. [Accessed: 12-July-2018].
- [7] I. Milne and G. Rowe, 'Difficulties in Learning and Teaching Programming—Views of Students and Tutors', *Education and Information Technologies*, vol. 7, no. 1, pp. 55–66, 2002. <https://doi.org/10.1023/A:1015362608943>
- [8] M. A. Miljanovic and J. S. Bradbury, 'Robot on!', *Proceedings of the 5th International Workshop on Games and Software Engineering - GAS '16*, pp. 33–36, 2016.
- [9] B. G. H. Al-Bastami, 'Design and Development of an Intelligent Tutoring System for C# Language', *European Academic Research*, vol. 9, no. 10, pp. 8795–8809, 2017.
- [10] C. D. Rodríguez and T. M. Cumming, 'Employing mobile technology to improve language skills of young students with language-based disabilities', *Assistive Technology*, vol. 29, no. 3, pp. 161–169, Jul. 2017. <https://doi.org/10.1080/10400435.2016.1171810>
- [11] O. Viberg and Å. Grönlund, 'Understanding students' learning practices: challenges for design and integration of mobile technology into distance education', *Learning, Media and Technology*, vol. 42, no. 3, pp. 357–377, Jul. 2017. <https://doi.org/10.1080/17439884.2016.1088869>
- [12] N. Martín, Sergio; Sancristobal, Elio; Gil, Rosario; Díaz, Gabriel; Castro, Manuel; Peire, Juan; Milev, Mihail; Mileva, 'Enabling Innovations in Mobile-Learning: A Context-aware and Service-based Middleware', *International Journal on Advances in Software*, vol. 3 nr 1&2, pp. 175–185, 2010.
- [13] V. Tunalı and Ş. Z. Erdoğan, 'Comparison of Popular Cross-Platform Mobile Application Development Tools', 2. *Ulusal Yönetim Bilişim Sistemleri Kongresi (YBS2015)*, no. October 2015, pp. 1357–1365, 2015.
- [14] 'Xamarin Documentation - Xamarin | Microsoft Docs'. [Online]. Available: <https://docs.microsoft.com/en-us/xamarin/>. [Accessed: 20-Aug-2018].
- [15] D. Hermes, *Xamarin Mobile Application: Cross-Platform C# and Xamarin. Forms Fundamentals*. 2015.

- [16] M. Martinez and S. Lecomte, 'Towards the Quality Improvement of Cross-Platform Mobile Applications', in 2017 IEEE/ACM 4th International Conference on Mobile Software Engineering and Systems (MOBILESoft), 2017, pp. 184–188.
- [17] M. Carapina et al., 'Developing a multiplatform solution for mobile learning', Proceedings of the 23rd International Conference on Computers in Education ICCE 2015, pp. 384–389, 2015.
- [18] 'protégé'. [Online]. Available: <https://protege.stanford.edu/>. [Accessed: 30-Aug-2018].
- [19] S. Lohmann, S. Negru, and D. Bold, 'The ProtégéVOWL Plugin: Ontology Visualization for Everyone', 2014, pp. 395–400.
- [20] T. Bürger and E. Simperl, 'Measuring the Benefits of Ontologies', Springer, Berlin, Heidelberg, 2008, pp. 584–594. https://doi.org/10.1007/978-3-540-88875-8_82
- [21] A. Drigas and P. Angelidakis, 'Mobile Applications within Education: An Overview of Application Paradigms in Specific Categories', International Journal of Interactive Mobile Technologies (iJIM), vol. 11, no. 4, p. 17, May 2017. <https://doi.org/10.3991/ijim.v11i4.6589>

8 Authors

Alaeddine YASSINE is with Laboratory of Computer Science and Interdisciplinary Physics (LIPI), Ecole Normale Supérieure, Sidi Mohamed Ben Abdallah University, Fes, Morocco (e-mail: alaeyassine@gmail.com).

Mohammed BERRADA is with the Laboratory of Computer Science and Interdisciplinary Physics (LIPI), Ecole Normale Supérieure, Sidi Mohammed Ben Abdallah University, Fes, Morocco (e-mail: mohammed.berrada@gmail.com).

Ahmed TAHIRI is with the Laboratory of Computer Science and Interdisciplinary Physics (LIPI), Ecole Normale Supérieure, Sidi Mohammed Ben Abdallah University, Fes, Morocco (e-mail: tahiri_ahmed@hotmail.com).

Driss CHENOUNI is with the Laboratory of Computer Science and Interdisciplinary Physics (LIPI), Ecole Normale Supérieure, Sidi Mohammed Ben Abdallah University, Fes, Morocco (e-mail: d_chenouni@yahoo.fr).

Article submitted 30 August 2018. Final acceptance 19 October 2018. Final version published as submitted by the authors.