Mobile Learning and computational thinking

Apprendimento su dispositivi mobili e pensiero computazionale

José Manuel Freixo Nunes* and Teresa Margarida Loureiro Cardoso

LE@D - Laboratório de Educação a Distância, Aberta University, Lisbon, Portugal, jmnunes@lead.uab.pt *, teresa.cardoso@uab.pt

* corresponding author

HOW TO CITE Freixo Nunes, J. M. , & Loureiro Cardoso, T. M. (2017). Mobile learning and computational thinking. *Italian Journal of Educational Technology, 25*(2), 51-58. doi: 10.174 71/2499-4324/905

1. INTRODUCTION

Computational thinking can be thought of as an approach to problem solving which has been applied to different areas of learning and which has become an important field of investigation in the area of educational research. Its main conceptual tools are decomposition, pattern recognition, abstraction and algorithm design. In the project described here, this new paradigm was used with ninth grade pupils who were involved in the development of mobile applications (apps). The choice of m-learning is motivated by its many affordances for changing or adding value to learning and teaching processes. One of these is that with mobiles it is possible to carry out new types of distributed learning activities that are not strictly confined to the classroom.

To support our pupils in app development, we used the conceptual tools of computational thinking and the App Inventor platform developed at MIT. We explained how software development can be made easier and more understandable without resorting to long and complex coding activities. For instance, in 1981 IBM launched the personal computer, IBM PC (Freed & Ishida, 1995) and users were able to develop their own programs using a BASIC programming language interpreter (Beginner's All-purpose Symbolic Instruction Code) and a manual with detailed explanations on how to program, both included in the computer.

2. MOBILE LEARNING

Mobile Learning (m-learning) can be defined as the learning process performed with apps; this entails accessing information anywhere, anytime, using a mobile device - mobile phone, smartphone or tablet.

According to the ITU (International Telecommunication Union) there are about seven billion mobile phone users worldwide (Sanou, 2015); this number is greater than the world population, meaning that there are people with more than one mobile phone. Thus, as the number of users increases, so m-learning and m-teaching, performed formally or informally, may increase. This deserves special attention, especially from educators: pupils' mobile devices are connected most of the time and their owners possess informally developed technical skills and familiarity with specific applications (games, quizzes, social networks, etc.). Other advantages of m-learning more specifically related to teaching and learning should also be mentioned. One is the possibility to help schools and pupils with low budgets because the price of mobile devices is decreasing. Another is the potential related to connectivity, communication and collaboration, which allows individuals to develop a high level of thinking where the exchange of ideas related to problem solving can be permanent (McQuiggan, McQuiggan, Kosturko, & Sabourin, 2015).

John Traxler, an expert and researcher at Wolverhampton University in the UK, says that the use of mobile or wireless handheld devices has been increasing in various sectors of education. The number of publications and projects presented on the subject at various conferences shows that mobile education has a future which should be explored (Traxler & Kukulska-Hulme, 2016). We should thus take advantage of this opportunity and enhance it by guiding our pupils to develop apps themselves, of their interest, because the educational impact will be even greater.

We have been challenging our pupils at the Colégio Manuel Bernardes (Lisbon, Portugal) to design applications to teach and learn through mobile devices/technologies. The best of these will be developed, published and shared with the whole community. The following figures show some of the projects (totalling around forty already) presented by the ninth grade pupils who have been working in groups on this school activity.

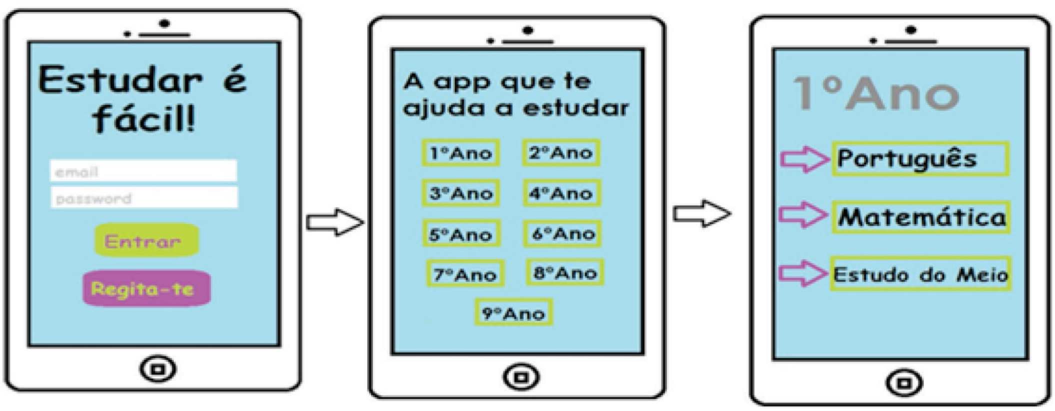


Figure 1. The “Studying is easy”app.

Figure 2. The “Body Bones” app.

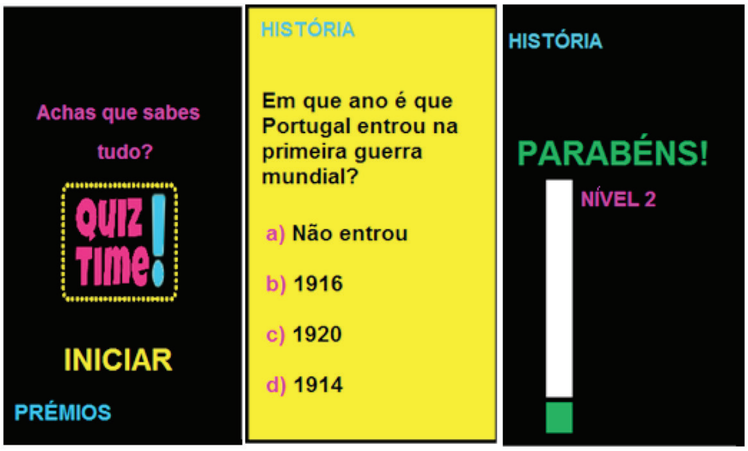


Figure 3. The "Wanted Time!" app.

Our approach to m-learning has added value to the educational process, particularly in the Information and Communication Technologies (ICT) classes in which the pupils developed their projects. They were more motivated, and had the opportunity to carry out a project from the beginning, going through the various stages of development. Group work, the sharing of ideas and collaboration were other key aspects, especially during the final phase of the project, when the pupils installed the apps they had developed on their own mobile phones. It is worth mentioning that the motivational aspects played an important role not only for the pupils but also for the teachers.

3. COMPUTATIONAL THINKING

New software development tools, such as the MIT App Inventor, make it possible to create apps fairly easily. Indeed, the Massachusetts Institute of Technology has been a pioneer in many areas, including the development of apps. As to Computational Thinking, it is worth mentioning the development of Scratch, a project of the Lifelong Kindergarten Group at the MIT Media Lab, launched in 2005 and led by Mitchel Resnick. Using Scratch, anyone can develop a program within minutes, even with no prior knowledge of programming. Moreover, "Scratch helps young people learn to think creatively, reason systematically, and work collaboratively - essential skills for life in the 21st century"[1]. A visit to the Scratch web site is enough to learn the first steps in programming by blocks. This is a system of blocks or objects that fit into each other and that are associated with certain events, making it possible to create both very simple and more complicated programs. The keywords associated with Scratch are very appealing: imagine, plan, share. But why should we learn how to program? Resnick and his team of MIT researchers give us an answer: programming develops the ability to design, create and invent with the multiple digital media we have at our disposal. It is not enough to know how to search the Internet, send messages or copy images; we need to go further, especially at young ages and during the learning stages (Resnick et al., 2009). This vision caught our attention and triggered our interest in programming. A second factor of interest was the opportunity to develop mobile applications for Android smartphones in a Scratch-like way, something made possible by the use of App Inventor. This is based on
the same block system, allowing users to develop applications without learning to code with a programming

[1] https://scratch.mit.edu/

language. The App Inventor development team was led by MIT's Hal Abelson and Google's Mark Friedman. Their work was inspired by Ricarose Roque's master thesis "Making Together: Creative Collaboration for Everyone", where the author analyses the way to develop learning environments aimed at involvement and cooperation through one-to-many and many-to-many relationships. Sharing information on the construction of a particular object is supported by networked environments and is related to collaborative learning (Roque, 2012). Indeed, sharing, collaboration, information exchange, and application development is what matters in App Inventor.

The Scratch and App Inventor models and structures are easy for people of any age to learn, whether or not they have previous knowledge of programming languages. However, there is a high cognitive value in applying the concept of Computational Thinking when using these environments for Apps development. This problem solving approach was initially proposed by Janette Wing of Carnegie Mellon University. She suggests that the steps taken by a computer scientist to solve a problem can be applied to many areas, not just in software development. She further acknowledges that this new approach will be a key skill in the 21st century, in addition to reading, writing and performing arithmetic calculations (Wing, 2006).

Those steps can be summarized as follows: (i) the decomposition of a problem into smaller and easier problems to work with; (ii) the recognition of patterns, trends and harmony in the data; (iii) the definition of abstractions or models associated to the patterns found; and (iiii) the algorithm which can solve the encountered problems. The final step is data analysis and presentation of the work (Soman, Unni, Krishnan, & Sowmya, 2012). This way of thinking and solving problems, using an algorithmic structure, is helpful because it helps to transform complicated problems into less complicated and clearer ones. Software engineers, programmers and researchers (among others) apply this technique, and the results they have achieved in terms of the development and evolution of computers and technology in recent decades are clearly evident (Riley & Hunt, 2014). Google is also committed to supporting this new way of thinking and approaching problems, having in mind not only the development of applications, but also the possibility to solve problems in areas as diverse as mathematics or history. The website called "Exploring Computational Thinking" releases computational thinking courses ("Google for Education - Exploring Computational Thinking", n.d.). Another US initiative regards the CSTA (Computer Science Teachers Association), which has created a working team to explore and disseminate this way of thinking and approaching problems.

For the development of apps by our ninth grade pupils, we also developed a specific Mobile Learning approach. First, we addressed the topic "Project work", in order to explain how to plan and carry out a project. Pupils were taught that in project work the aim is to identify, investigate, analyse and solve problems in groups or individually. Design was presented as an investigation of a theme, a problem or a situation in order to present a solution. It was clarified that all design work has a purpose, which should be well defined. It was at this initial stage that pupils heard about computational thinking for the first time, particularly about dividing the project into small parts. This makes it easier to deal with and can lead to the discovery of problem patterns and investigation of the best way to solve the problems found, using certain models. As an example, we can think of an application for calculating the area of a triangle. In this case, pupils need to understand that the problem is solved by the formula Area = (Base * Height) / 2; this formula applies to any kind of triangle. As a second step, each working group was asked to prepare a report on their project. As an example, a group report on the "UseYourBrain" application (Figure 4) is quoted:

«*This free educational application is for people from 10 years and up. The user has access to various puzzles and riddles. It is meant to test the user's thinking, and in the case of a wrong answer, it shows the sought solution and an explanation for it, so that people can learn from their mistakes. This application has different levels so that the user can adjust the level of difficulty to matches his/her knowledge. The harder the level chosen, the more difficult puzzles will be, thus requiring deeper reasoning. In daily life, we are faced with situations that require the use of higher reasoning. UYB (Use Your Brain) provides fun and, at the same time, exercises your brain*».

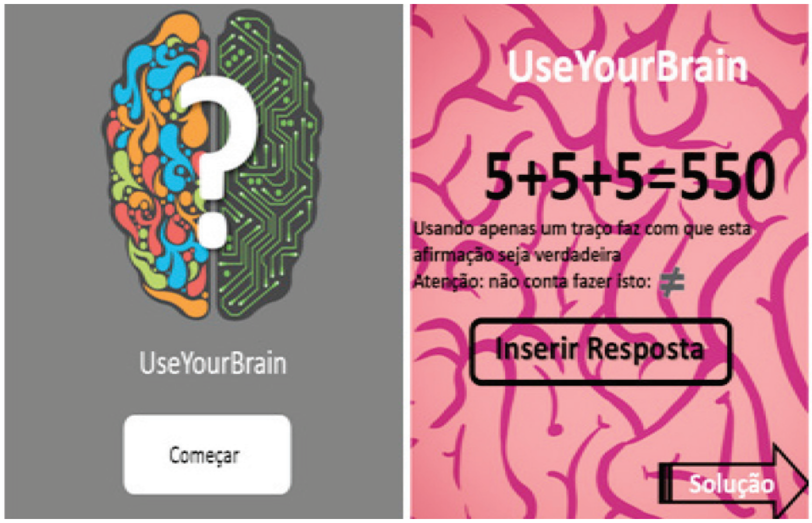


Figure 4. The “Use Your Brain” app.

The third stage of our Mobile Learning approach was the presentation of the App Inventor environment to the pupils (Figure 5). Pupils were led to helped to consider the various parts into which the screen could be split, which code blocks were associated to each part, and its organization in the form of algorithm. The last stage involved testing the application in order to correct mistakes; subsequently the apps were shared with classmates. It was then that the pupils could verify the importance of computational thinking in solving problems. Computational thinking was applied throughout the process, even during the formulation of the problem. From the beginning, pupils were asked to analyse the problem by dividing it into smaller problems, often finding in this division analogous problems and solutions.

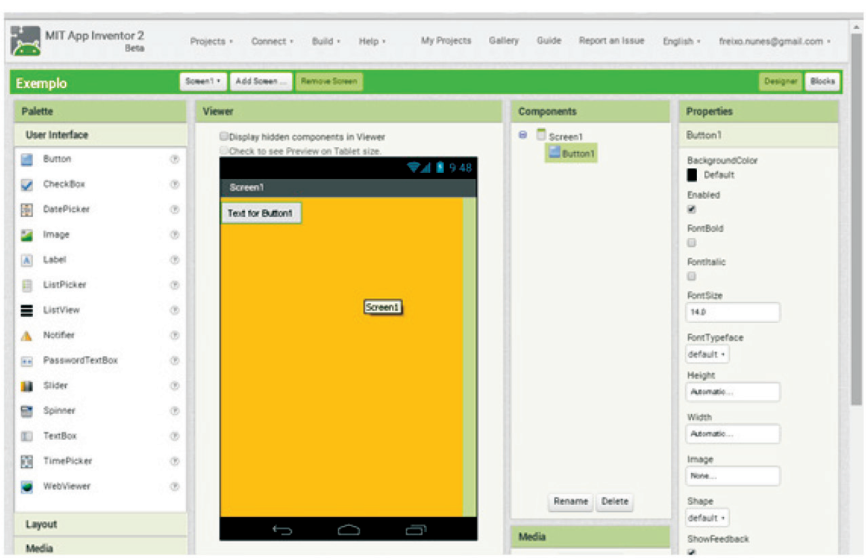
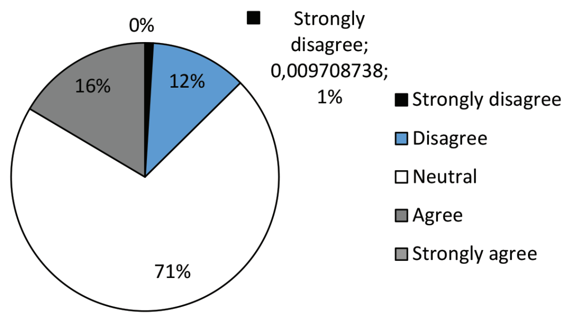
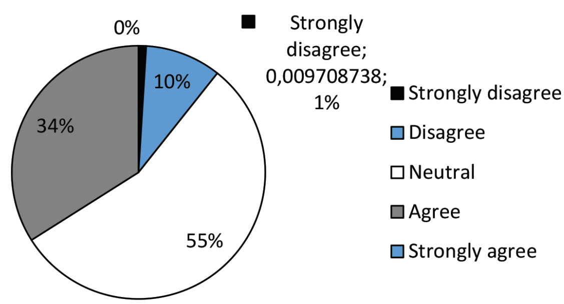
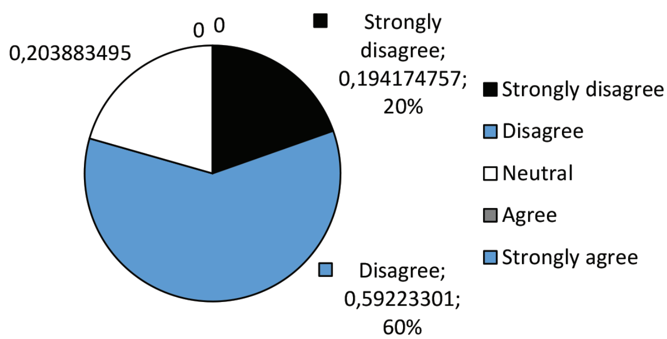


Figure 5. App Inventor.

To evaluate the project, we developed a survey on mobile learning and computational thinking, which was applied to a target group of about 100 pupils. The survey consisted of six questions with regard to computational thinking. The answers are summarized in Table 1.

<table>
<tr><th>Questions</th><th>Results</th></tr>
<tr><td>1 – Bearing in mind what you've learnt in the classroom and in your research, do you consider computational thinking an important approach in solving problems?</td><td>
</td></tr>
<tr><td>2 – One of the stages of computational thinking is to divide a problem into smaller and easier problems to solve it. Do you agree that this approach is important in developing an app?</td><td>
</td></tr>
<tr><td>3 – Another step in computational thinking is related to pattern recognition, i.e. in the development of an application there are trends, and regularities in data. Does this step (pattern recognition) facilitate the development of an app?</td><td>
</td></tr>
</table>

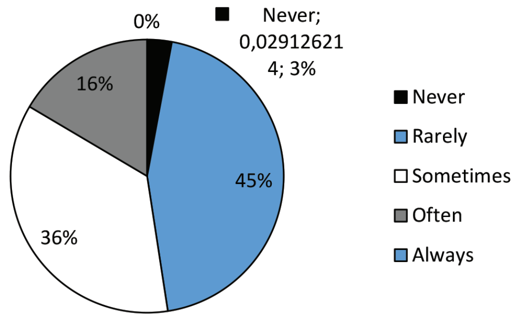
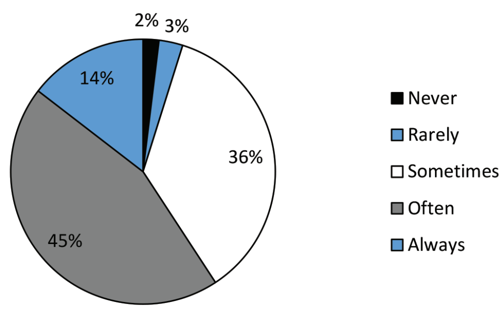
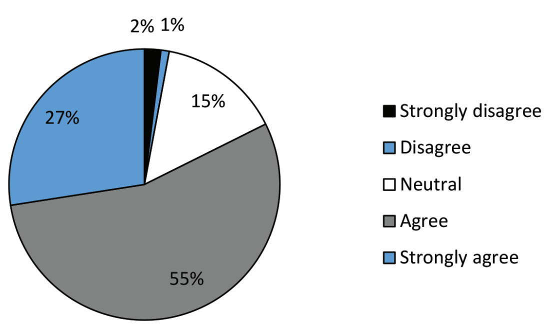
4 – The identification of certain principles (command blocks or instructions) that generate the standards referred to in the previous question is called an abstraction. When developing your application, could you achieve this type of abstraction?	
5 – Applying algorithms in the development of an application means using a set of instructions or commands, sequentially, repetitively or by taking decisions. When developing your application, did you apply this approach?	
6 – Can computational thinking be applied to other areas of study, and not only in the development of applications?	

Table 1. “Mobile Learning and Computational Thinking” survey (Questions / Results).

4. CONCLUSIONS

The pupils who participated in our project had no previous experience of Computational Thinking. The development of mobile applications with the purpose of learning by applying Computational Thinking turned out to be a good approach to problem solving, as shown by pupils’ answers to the survey. About 71% recognized the value of this type of approach. About 55% agreed that the development of an app is made easier when planning and dividing it into smaller problems, and 59% recognized the usefulness of finding equal or similar patterns (trends and regularities in data) in the development of an app. The process of abstraction was less clear. In fact, only about 45% of the pupils were able to make proper use of it. The application of computational thinking to other areas of study was considered important by 54% of the pupils, and 63% thought it was a good contribution to the design and development of the app. Extensive testing of the developed applications and verification of the achievement of learning goals through application use were not among the goals of the project. In other words, the ultimate goal was not to test the learning process in a given topic, but to investigate the application of computational thinking in the development of mobile apps. Nevertheless, the apps developed by our pupils were shared among the different groups.

We can foresee the evolution of our pedagogical project through the testing of the applications with other pupils. In fact, during the current 2016/2017 school year, a ninth grade class is developing an app to explain and test the Pythagorean Theorem. We believe they will complete this app at the end of the 2nd Term, and then it will be tested in four eighth grade classes to assess what was learned about this theorem in the maths class. For the future, we will most likely continue to combine mobile learning, computational thinking and group activity.

5. REFERENCES

Freed, L., & Ishida, S. (1995). *History of computers*. Emeryville, CA: Ziff-Davis Publishing Co.

McQuiggan, S., McQuiggan, J., Kosturko, L., & Sabourin, J. (2015). *Mobile learning - A handbook for developers, educators and learners.* Hoboken, NJ: Wiley & Sons, John.

Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., ... & Kafai, Y. (2009). Scratch: programming for all. *Communications of the ACM, 52*(11), 60-67. doi:10.1145/1592761.1592779

Riley, D., & Hunt, K. (2014). *Computational thinking for the modern problem solver.* La Crosse, WI: CRC Press.

Roque, R. V. (2012). *Making together: Creative collaboration for everyone* [Doctoral dissertation, Massachusetts Institute of Technology]. Retrieved from https://dspace.mit.edu/handle/1721.1/77814

Sanou, B. (2015). ICT Facts & Figures. The world in 2015. Itu 150 Años (1865 - 2015), 6. [Infographic] Retrieved from http://www.itu.int/en/ITU-D/Statistics/Documents/facts/ICTFactsFigures2015.pdf

Soman, K. P., VG, M. U., Krishnan, P., & Sowmya, V. (2012). Enhancing Computational Thinking with Spreadsheet and Fractal Geometry: Part 1. *International Journal of Computer Applications, 55*(14), 1-8. doi:10.5120/8820-2741

Traxler, J., & Kukulska-Hulme, A. (2016). *Mobile learning: The next generation*. New York, NY: Routledge.

Wing, J. M. (2006). Computational Thinking. *Communications of the ACM, 49*(3), 33–35. doi:10.1145/1118178.1118215