

"The Hour of the Code": Computational thinking workshop in a primary school in Guarda, Portugal

Carlos Jorge Brigas*
Polytechnic Institute of Guarda

José Alberto Quitério Figueiredo
Polytechnic Institute of Guarda

*Corresponding Author: brigas@ipg.pt
Received: 08.08.2019
Revision: 06.09.2019
Accepted: 23.09.2019

How to cite this paper: Brigas, C. J., & Figueiredo, J. A. Q. (2019). "The Hour of the Code": Computational thinking workshop in a primary school in Guarda, Portugal. *Research in Social Sciences and Technology*, 4(2),129-136.

Abstract

In recent years, there has been an increased effort to introduce coding and computational thinking in early childhood education. In accordance with the international trend, programming has become an increasingly growing focus in European education. The idea of introducing computer programming in the classroom dates back to the late 1960s. The introduction of programming language provides an opportunity to engage in logical and abstract thinking, problem solving, and the creative design process. One of the best-known initiatives is The Hour of Code. This initiative is being implemented in 180 countries and more than 700 million students have participated. The Hour of Code started as a one-hour introduction to computer science, designed to demystify "code," to show that anybody can learn the basics, and to broaden participation in the field of computer science. This article describes a workshop held under the "The Hour of the Code" initiative and implemented in the primary schools of Guarda, Portugal. The activity took place at the school and lasted two hours. Students from the first to the fourth grade participated, with ages ranging from 6-11 years. The teachers of the activity were university professors of computer engineering and students of computer engineering.

Keywords: Coding and computational thinking, ICT in Education, Elementary education

Introduction

In recent years, there has been an increased effort to introduce coding and computational thinking in early childhood education. In accordance with the international trend, programming has become an increasingly growing focus in European education. The idea of introducing computer programming in the classroom dates back to the late 1960s. The introduction of programming language provides an opportunity to engage in logical and abstract thinking, problem solving, and the creative design process.

Seymour Papert (2000) identified the potentialities of introducing children to programming languages as an incubator of powerful ideas, that is, as a tool to engage children in new ways of thinking and "thinking about thinking" (Papert, 2005). Programming language allows and creates an opportunity to develop into logical and abstract thinking, problem solving, and the creative design process. According to Papert (1980), the potentialities of programming languages become evident through the observation of children's participation in computational thinking projects. In the 1960s, the MIT team with which Papert was working developed the idea of introducing computer programming into the classroom. Papert developed the first programming language designed for children, LOGO.

Some projects have since been promoted in the United States. For example, the Computer Science for All initiative was launched to bring programming into every educational level (Smith, 2016). In Europe, some countries integrate coding in the curriculum at the national, regional, or local level, including Austria, Bulgaria, the Czech Republic, Denmark, Estonia, France, Hungary, Ireland, Israel, Lithuania, Malta, Spain, Poland, Portugal, Slovakia, and the UK (Balanskat & Engelhardt, 2014; Bocconi et al., 2016; Livingstone, 2012).

One of the best-known initiatives is The Hour of Code. This initiative is being implemented in 180 countries and more than 700 million students have participated. The Hour of Code started as a one-hour introduction to computer science, designed to demystify "code," to show that anybody can learn the basics, and to broaden participation in the field of computer science.

Computational Thinking in elementary school

In recent years, there has been a growing effort to introduce coding and computational thinking in elementary school.

The term Computational Thinking, which is now widely known, was first used in 1980 by Papert. Various definitions and critiques of the concept "have been criticized for inaccuracies, ambiguous definitions and visions of Computational Thinking, and arrogance, as well as bold and unfounded claims about the universal benefit of Computational Thinking" (Tedre & Denning, 2016, p. 120). For Wolfram (2016), Computational Thinking "is about formulating things clearly enough and, in a sufficiently systematic way, it is possible to tell a computer how to do them." For us, Computational Thinking is an essential competence in child development and has the particularity of being transversal in all disciplines.

In the learning environment, it can be implemented by combining problem solving, understanding problems, or creating new solutions. These innovative learning environments can use computers to formulate solutions, and these environments are rich for enhancing the expressive creativity and evaluative power of students' ideas. Collaboration around a computer makes learning meaningful and supportive. In this sense, Computational Thinking is both delightful and effective for students.

Why should we use Computational Thinking in elementary school?

Some researchers (e.g., Waite, 2018) argue that Computational Thinking requires dealing with levels of abstraction that are not compatible with the cognitive development of primary age children. But we accept that it is possible to introduce Computational Thinking in elementary school. Some research presents pragmatic reasons for beginning Computational Thinking at this teaching level:

1. Cross-curricular approaches: Computational thinking allows the creation of transversal activities to different teaching areas.
2. Promote gender equality: Several researches point out that promoting computational thinking in learning environments in primary school is a very effective way to increase gender equality (Graham & Latulipe, 2003; Lapan et al., 2000; Turner et al., 2008).
3. Motivation, confidence, pleasure: Students design and participate in activities that involve Computational Thinking and, if implemented using technology such as PC, tablet, and smartphones, there is a clear satisfaction in learning itself, providing the pleasure of "enthusiasm" through innovative environments, which allows free choices and effective outcomes. As Millwood (2008) points out, "such enthusiasm is usually fostered through play, where children are significantly responsible for choices and outcomes even when founded on imagination and fantasy."
4. Collaboration: Computational Thinking projects can be designed as teamwork, which enables children to acquire the ability to consult, work, and learn from others. It also has the ability to rely on play and, in particular, the delight of "coexistence" (Millwood, 2008).
5. Creativity: Computational Thinking activities enable children to participate in rich environments and can use arts, sciences, and technology to solve problems or create new solutions.

How to teach Computational Thinking in early childhood

Learning to code involves children in new ways of thinking that some researchers have called Computational Thinking (Barr & Stephenson, 2011; International Society for Technological Education and The Computer Science Teachers Association, 2011; Lee et al., 2011; Tarman & Baytak, 2012; Valiandes & Tarman, 2011; Wing, 2006). It allows children to develop and acquire analytical mental tools that are inherent in the field of computer science, including recursive thinking, the application of abstraction to discover a complex task, and the use of heuristic reasoning to find a solution. These mental tools are universally applicable (Wing, 2011).

You do not need to be in Computer Science courses to gain this knowledge, but in an integrated manner with other curricular subjects in school and from an early age; instead of teaching children about specific programming languages and hardware, which often complicates and

alienates children, teachers and schools should prepare children to think with creativity, complexity, and logic.

Computational Thinking is understood as a combination of four skill categories:

- Pattern recognition: Pattern recognition is the process of identifying, defining, extending, and creating patterns. Pattern recognition requires the classification of data. For example, a preschooler learns pattern recognition when she sorts blocks according to attributes like shape or color.
- Creating and using algorithms: An algorithm is a set of steps to solve a problem. Programming a computer and solving an algebraic proof involves creating and using algorithms. One can solve tasks that involve creating and using algorithms in the form of step-by-step instructions.
- Decomposition: Decomposition is an analytical process that involves breaking something down into smaller parts; normally, this kind of activity involves dividing a large thing into smaller things. In mathematics, it is very easy to create this kind of activity.
- Understanding abstractions: We understand abstraction as something that exists only as an idea. Understanding abstractions requires the ability to make generalizations and use other problem-solving thought processes to imagine something we cannot see or touch.

Computational Thinking workshop in a primary school in Guarda

The activity took place at the students' school and the duration of two hours. Participants in the activity were students ages 6-11 in the first to fourth years of the first cycle of primary education. The teachers of the activity were college professors of computer engineering and students of computer engineering.

Instead of teaching children about a specific programming language, which is often too complicated for children, we define in this workshop activities where children must solve problems by creating and using algorithms.

In the following algorithm, children must define a set of step-by-step instructions that solve a problem. In this case, the children play a robot role (Figure 1).



Figure 1. Robot Workshop.

After a short explanation, the children needed no help exploring and creating algorithms to solve problems. Students had the possibility to customize the robot with different appearances and sounds according to their likes (Figure 2).

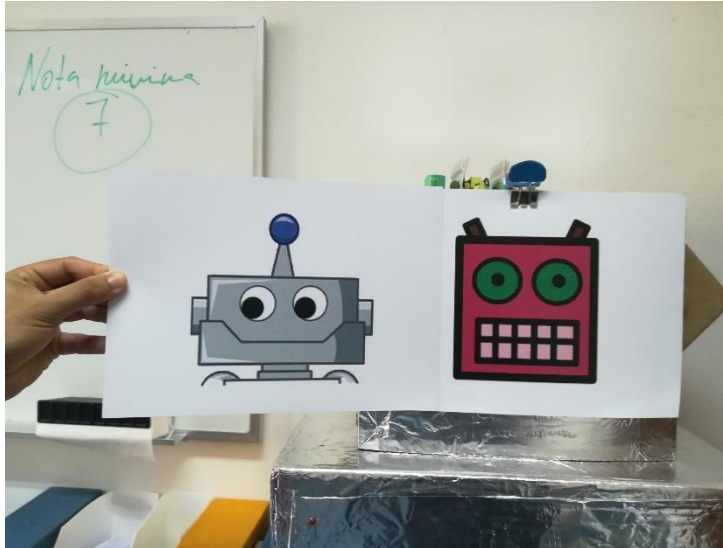


Figure 2. Robot face.

These activities were popular, and all of the children wanted to assume the role of robot. All the children helped each other, showed collaborative behavior, created several solutions to the same problem, and always tried to overcome previous mistakes; it was also evident that it created some competitiveness between them.

During the exploration of the model, a large number of children changed their minds and modified the algorithm for better resolution.

Conclusions

The major goal of these workshops was to evaluate the difficulties and potentialities of using Computational Thinking in elementary schools. The methodology used allowed us to observe the exploration and authoring of activities performed by children in a "natural" environment, the classroom. Working with the whole class allowed for observing diverse ways to explore the tools, too. The use of Computational Thinking in education facilitates the development of several important capacities (understanding, reasoning, decision-making, etc.) that we observed. Using Simulkids, children could make predictions and explain events which otherwise would not be possible with the same ease.

References

- Balanskat, A., & Engelhardt, K. (2014). Computing our future: Computer programming and coding priorities, school curricula, and initiatives across Europe. *European Schoolnet*. http://www.eun.org/c/document_library/get_file?uuid=521cb928-6ec4-4a86-b522-9d8fd5cf60ce&groupId=43887
- Barr, D., Harrison, J., & Conery, L. (2011). Computational thinking: A digital age skill for everyone. *Learning & Leading with Technology*, 38(6), 20-23.
- Barr, V., & Stephenson, C. (2011) Bringing computational thinking to K-12: What is involved and what is the role of the computer science education community? *ACM Inroads*, 2(1), 48-54. doi: 10.1145/1929887.1929905
- Bocconi, S., Chiocciariello, A., & Earp, J. (2018). The Nordic approach to introducing computational thinking and programming in compulsory education. Report prepared for the Nordic@BETT2018 Steering Group. Available at <https://doi.org/10.17471/54007> (Accessed 28 March, 2018.)
- Graham, S., & Latulipe, C. (2003). CS girls rock: Sparking interest in computer science and debunking the stereotypes. *ACM SIGCSE Bulletin*, 322-326.
- International Society for Technology in Education and the Computer Science Teachers Association. (2011). Operational definition of computational thinking for K-12 thinking. International Society for Technology in Education (ISTE) and the Computer Science Teachers Association (CSTA).
- Lapan, R. T., et al. (2000). Seventh graders' vocational interest and efficacy expectation patterns. *Journal of Career Development*, 26(3), 215-229.
- Lee, I., Martin, F., Denner, J., Coulter, B., Allan, W., Erickson, J., et al. (2011). Computational thinking for youth in practice. *ACM Inroads*, 2(1), 32-37.
- Livingstone, S. (2012). Critical reflections on the benefits of ICT in education. *Oxford Review of Education*, 38(1), 9-24.
- Millwood, R. (2008, 15 May). "An Analysis of Delight." Available at <http://blog.richardmillwood.net/2008/05/15/an-analysis-of-delight/>
- Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. Basic Books, Inc.
- Papert, S. (2000). What's the big idea? Toward a pedagogy of idea power. *IBM Systems*

Brigas & Figueiredo. "The Hour of the Code": Computational thinking workshop in a primary school in Guarda, Portugal.

Journal, 39(3.4), 720-729.

Papert, S. (2005). You can't think about thinking without thinking about thinking about something. *Contemporary Issues in Technology and Teacher Education*, 5(3/4), 366-367.

Tarman, B. & Baytak, A. (2012). "Children's online language learning: A constructionist perspective", *Energy Education Science and Technology Part B: Social and Educational Studies*, 4(2) 875-882

Tedre, M., & Denning, P. J. (2016). The long quest for computational thinking. In *Proceedings of the 16th Koli Calling International Conference on Computing Education Research* (pp. 120-129). New York, NY: ACM. doi: 10.1145/2999541.2999542

Turner, S., et al. (2008). Gender differences in Holland vocational personality types: Implications for school counselors. *Professional School Counseling*, 11(5), 317-326.

Valiandes, S. & Tarman, B. (2011). Differentiated Teaching and Constructive Learning Approach by The Implementation of ICT in Mixed Ability Classrooms, *Ahi Evran University Journal of Education Faculty, (KEFAD)12* (1), 169-184.

Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33-35.

Wing, J. M. (2011). Research notebook: Computational thinking-What and why? *The Link Magazine*, Spring. Pittsburgh, PA: Carnegie Mellon University.