CODING AND COMPUTATIONAL THINKING WITH ARDUINO

Veronica Rossano, Teresa Roselli and Gaetano Quercia
Department of Computer Science – University of Bari
Via Orabona, 4, Bari - Italy

ABSTRACT

The Computational Thinking recently has been recognised as one of the basic knowledge to be developed since childhood. Coding and computers are not just programming, but tools that help students to develop problem solving skills and more deep understand of the way things work. For these reasons, great attention has been focused on this topic both from a pedagogical and technological point of view. In this paper, a first approach to Computational Thinking using ARDUINO is presented. To this end, some learning activities have been designed to introduce middle school students, without any experience in coding, to the process of building the algorithm from simple exercises to more complex tasks. The pilot test involved 25 subjects, many of them do not like to study mathematics, science and technology, but the results were promising. The approach was appreciated by the students and the results of the questionnaires confirmed the learning effectiveness too.

KEYWORDS

Computational Thinking, Coding, Problem Solving, Arduino

1. COMPUTATIONAL THINKING

The importance of Computational Thinking (CT) was underlined since the 1980s when Seymour Papert stated that for children developing procedural thinking was necessary to develop basic skills about problem solving and deep learning. Papert defined LOGO (Papert, 1980, 1991) programming language that allows children to use a turtle to manipulate the computer and build simple programs. The current challenge of educational technology field is to make aware children that computers, and technologies in general, can be smart only if they are programmed by humans. One of the most dangerous and widespread misconceptions among young people is “technology is smart since the circuits are smart”. Moreover, as Jeannette Wing (2006 - p. 33) stated, computational thinking is “a universally applicable attitude and skill set everyone, not just computer scientists, would be eager to learn and use”. Several empirical evidences confirm that coding is effective in improving problem solving skills, divergent ways of thinking, creativity, communication and group work (Choi et al., 2013, Vaca-Cárdenas, et al., 2015, Siegle, 2017, Lye, 2014).

Following this trend, Google has activated the Exploring Computational Thinking (ECT) program (https://edu.google.com/resources/programs/exploring-computational-thinking/#!home) where a collection of lesson plans, videos, and other resources on computational thinking (CT) are published. To prove that CT is effective in any school subjects, the website offers didactic resources that can be used in different subjects, for example Algebra, Music, US-Story and so on.

The aim of the research is to verify if the use of physical technology, such as Arduino board, can be effective in teaching coding and develop computational thinking skills, in general. To this aim, a learning path has been designed and developed. The activities have been organised in order to introduce junior high school students, with no experience in coding, to the process of building the algorithm from simple exercises to more complex task. In addition, to make more engaging the activities some wooden models have been realised to allow students to see the effects of the programming on physical objects.

The paper is organised as follows: the next section describes the technological solutions available for learning coding and CT; section 3 describes the learning activities defined for the learning path; section 4 describes the pilot study to measure both the student's appreciation of the technological approach to education and the student's knowledge gain. Finally, some conclusions and future works are described.

2. TECHNOLOGICAL SOLUTIONS

It is clear that children are attracted by technology. The electronic devices have now entered in their everyday life: video games, computers, smartphones. Until a few years ago, these tools were used exclusively by adults, but today with the generation of "digital natives" it is common to see smartphones in children's hands.

Different technological solutions have been implemented in order to support the acquisition of computational thinking skills. Bee Bot®, as an example, is a floor robot that pupils, starting from 3 years old, can program using directional commands (e.g. forwards, backwards, left and right turns) to follow a path. For the same learning purpose there are also unplugged solutions. The Montessori play set, named Cubetto (https://www.primotoys.com/fr/), for example, is based on a tangible approach, children will be able to learn programming without the support of screens or the need to master basic bed-writing skills. Other solutions, combine technology with phisical objects, such as Coding blocks from Osmo (https://www.playosmo.com/en/) that is a tangible play platform to learn coding with physical blocks that interact with iPad and iPhone. There are also solutions based on software approach, the most known is the Hour of Code (https://code.org/) where different games based on Scratch are supplied. Scratch (Maloney, 2010) is a visual programming language that can be easly used by children to build media-rich projects, such as stories, games and animation. It is based on Papert's Logo project and currently is one of the most used language to introduce the programming at different ages (Vaca-Cárdenas et al., 2015).

To improve motivation and engagement in programming skills acquisition, in the latest years also robotic education is spreading. The idea is to create artifacts that can be programmed to do some tasks. The idea is to allow students to work with real hardware that will be able to do real physical work when programmed. The LEGO® Mindstorms are an example, they allow to develop programmable robots based on Lego building blocks. This approach has been successfully applied in different contexts (Haak et al. 2018, De Vries et al. 2018, Umbleja et al., 2017). Another device commonly used in teaching and learning contexts is Arduino (Plaza et al., 2018), an open source hardware and software for building digital devices and interactive objects that can sense and control objects in physical and digital world. Arduino board uses a variety of microprocessors and micontrollers that can be programmed using a language similar to C. The microcontrollers, unlike microprocessors which integrate only the processing unit (CPU), include permanent and volatile memories and I/O ports. This makes Arduino an autonomous system: the program, saved in the permanent memory, instructs the microprocessor which uses volatile memory to store data useful during the execution of the program. Moreover, there are different sensors which allow Arduino to measure light, temperature, degree of flex, pressure, proximity, acceleration, carbon monoxide, radioactivity, humidity, barometric pressure. Thus, the Arduino board can be used in order to solve real complex problems with physical objects, allowing students to program artifacts that can be touched with hands. The learning process is, in this way, more engaging and motivating.

3. DESIGN OF LEARNING ACTIVITIES

Until now, there are a lot of teaching experiences using coding to introduce the concept of problem solving and computational thinking from the early years of school (Lye & Koh, 2014). In this context, the research aims at defining a path for the acquisition of the basic concepts of programming with the use of Arduino and Scratch4Arduino. The learning path is addressed to 12-14 years old students and aims at improving motivation and engagement in technology subjects. The lessons were designed to introduce: the algorithm definition, the logical connective, the if-then-else and the while statements. The lesson involved the following tools: the Arduino Uno logic board, the IDE Scratch for Arduino and some wooden models built for this specific purpose.

Activity #1. Led controlled by switch

Learning objective: introduction to the electonic components of Arduino. No coding activity is required to handle this task. The Arduino board has been used as a power source. The circuit built in this way allows the LED to light up when the switch is pressed.

Activity #2. Blinking led

Learning objective: introduction to the algorithm concept. The idea is to write a simple program that is able to make the led bliking when the user makes an action, for example the user uses the spacebar.

Activity #3. Blinking led more than once

Learning objective: introduction to the while statement. Starting from the algorithm defined in the previous activity, the students were asked to build a program that repeats the instructions more than once.

Activity #4. Turn on and turn off the led

Learning objective: introduction to the if-then-else statement. Starting from the program in Activity #2 the idea is that the led should be turned on if the user pushes a specific button and turned off otherwise.

Activity #5. AND operator

Learning objective: this activity was designed in order to introduce the AND logical operator. Starting from the program in Activity #4 the idea is that the led should be turned on if the user pushes two specific buttons and turned off otherwise.

Activity #6. OR operator

Learning objective: introduction to the OR logical operator. Starting from the program in Activity #4 the idea is that the led should be turned on if the user pushes one of two specific button and turned off otherwise.

Activity #6. The Traffic Light

Learning objective: this activity allows the student to apply the knowledge acquired with the previous activities in real task. The idea is to build a program to tackle two traffic lights controlling a crossroads. In order to make simpler the problem solving process, a wooden model was used to see its effects on phisical objects (figure 1).

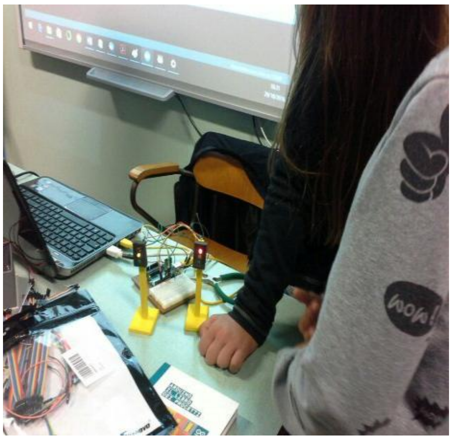

Figure 1. The wooden traffic light model

ISBN: 978-989-8533-81-4 © 2018

Activity #7 Rail crossing

Learning objective: this activity allows the student to apply the knowledge acquired with the previous activities in a more complex task. In order to make more engaging the activity a real world problem was proposed. The aim is to build a program to control both the traffic lights and the rail crossing (Figure 2). The user presses a button to close the bar, while the red lights start flashing. When the train has passed, the user presses another button to open the bar. The red lights shall continue to blink until the bar is completely open.

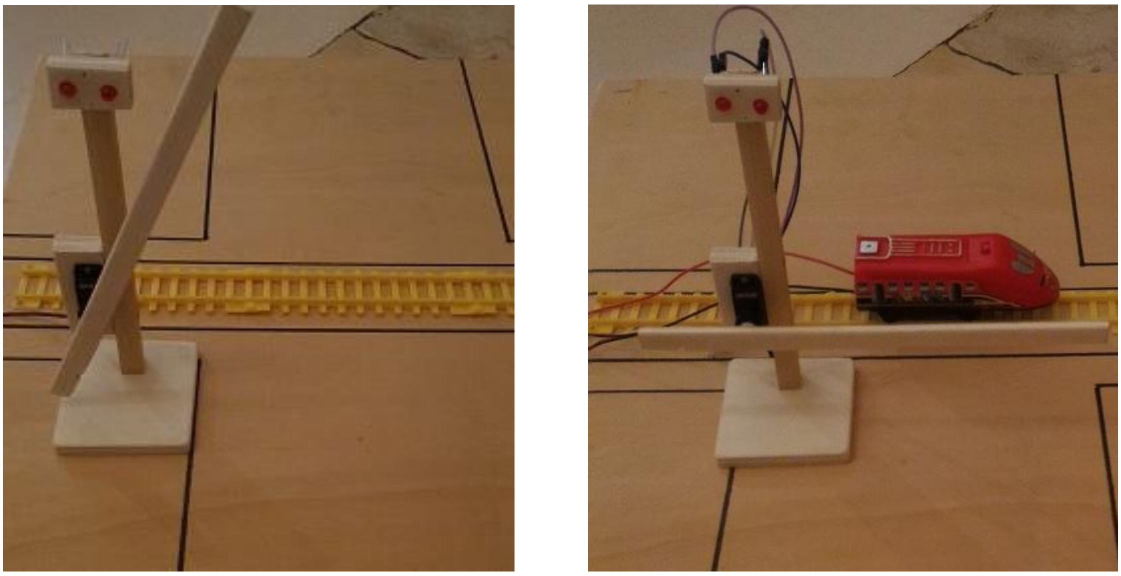

Figure 2 The wooden rail cross activity model

4. THE PILOT STUDY

4.1 The Resources

The pilot study was carried out in the laboratory of the school. It is composed of 20 Pentium 4 equipped with Windows XP Professional. The "Scratch 4 Arduino" tutorial was installed on these machines. Two wooden models used in Activity#6 or Activity#7 were built before the experiment.

4.2 The Participants

The participants were 25 students (11 girls and 14 boys) of the middle school in Terlizzi, a town near Bari, attending the 3rd class. All participants were 13 years old. Together with the teacher, we chose not to inform the participants about the activity in advance, in order to study their reaction and interest. None of the students had ever worked with Arduino and Scratch.

4.3 The Experimental Design

The within-subject design has been applied for the evaluation of effectiveness. The experiment was organized in five sessions distributed in two weeks: three meetings in the first week, two meetings in the next one.

Before the activity was presented, a pre-test was administered to collect information concerning previews knowldege about CT and coding. The pre-test was divided into two sections: the demographic section and the knowledge assessment section. The first section made up of twelve questions, was useful to understand how and how long the students use the PC, how much they are attracted by the technology (smartphone, game platforms, etc.), if they know the programmer's work and whether they have ever tried to program anything. The second section consists of six questions to assess previous knowledge about computational thinking. Specifically, in the first question the knowledge about the algorithm definition is investigated; two questions measure the ability to build an algorithm, to make easy the concept some algorithms related to daily activities were proposed (e.g. brushing teeth, making a phone call with the smartphone); two questions measure the knowledge of logical connectives AND and OR (in this case we use a colored figure on which to assess the truth value of some statements); the last question verifies the knowledge about flow-chart.

After the pre-test, the planned activities were carried out. In the first week Activities #1, #2, #3 and #4 were completed and in the second week Activities #5 and #6 were completed. Activity #7 was only presented to the class, but no actions were required from the students side.

The post-test was aimed at assessing the improvements of student's knowledge and the appreciation of the teaching approach. All the knowledge assessed in the pre-test were measured also in the post-test.

4.4 The Results

The pre-test reveals that 24 out 25 students got a pc at home. The students were asked to self-assess their level of e-skills, the average ratings was 7/10. Moreover, students usually spend 1.5 hours a day to study and/or play with PCs. Many of them do not like subjects such as Mathematics, Science and Technology, indeed the teacher confirms that their average grade was 5.5/10. As regards the computational thinking, only one student knows the binary system, no one knows the programmer's work, only 3 students have tried to program something; the students are interested in using technology (mean value 7.44/10).

For what concerning the algorithm skills, all students have many difficulties. Some of them have been able to build the algorithm only guided by the tutor and/or the teacher. The knowledge about the logic connectives was very low (mean 4.2), the grade obtained in the pre-test are reported in figure 3. In addition, data reveal that almost the whole sample has well-consolidated knowledge of flow charts and their proper use.

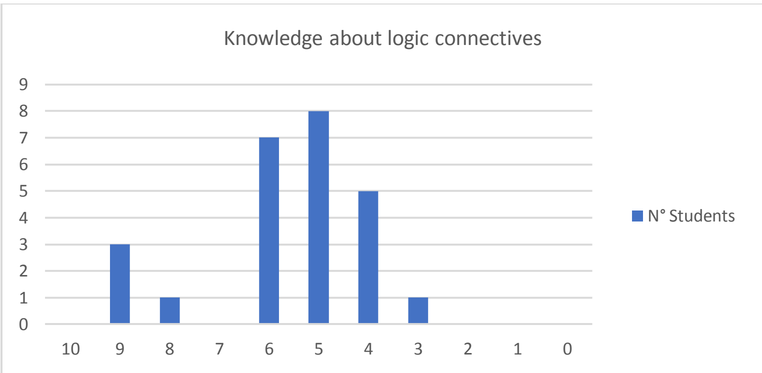

Figure 3. Previous knowlege about logic connectives AND and OR

4.4.1 Appreciation of the Teaching Approach

The 19 out of 25 students appreciate using Arduino 1, most of the sample (21 out of 25) considered enjoyable to assemble the circuit, and almost all the sample (23 out of 25) considered the whole project engaging. An unexpected result was that many students reported difficulties using Scratch (Figure 4), indeed 21 out of 25 students stated that they had no difficulties in carrying out the proposed activities thanks to the methodological approach adopted.

ISBN: 978-989-8533-81-4 © 2018

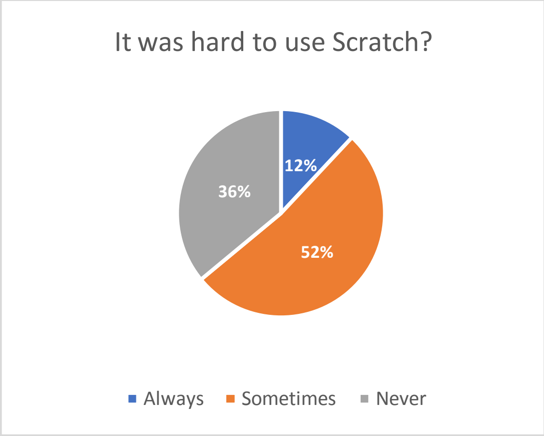

Figure 4. Students judge not simple using Scratch

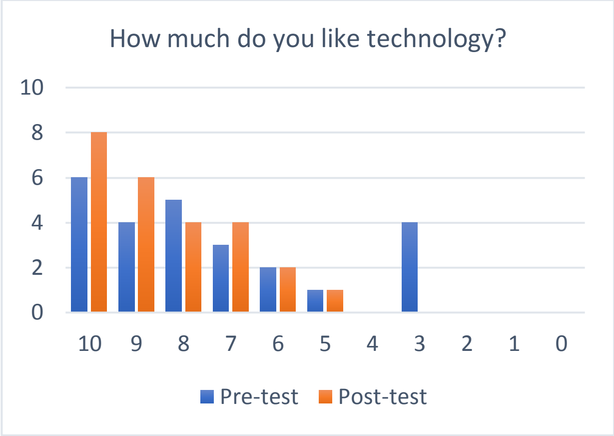

Figure 5. The improvement of student's interest in technology

An interesting result was related to the question, already posed in the pre-test, "How much are you attracted by technology and innovation?". As Figure 5 shows the distribution of the grades has changed, higher values than the pre-test were obtained, this could mean that thanks to this experience they see another application of technology and perhaps they perceive a higher utility of it.

4.4.2 Student's Knowledge Gain

All students stated that they learned from this new experience. This perception was confirmed by the student's higher marks obtained in the post-test (Figure 6).

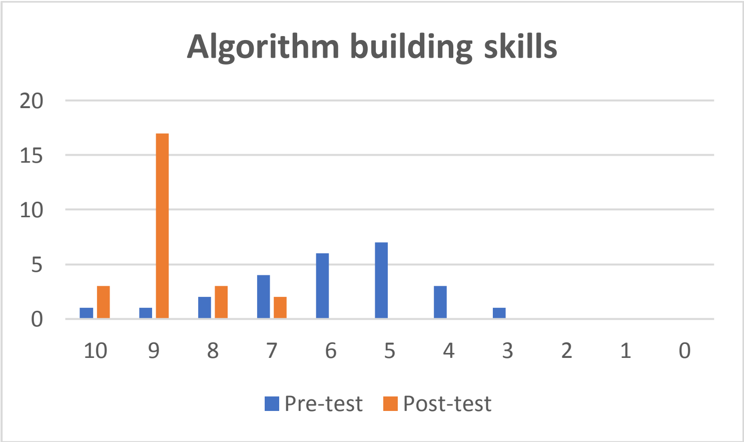

Figure 6. The improvement of ability to build an algorithm

All pupils achieved excellent evaluations of their ability to use logical AND and OR connectives (average grades 9.36 out of 10).

5. CONCLUSION

The importance of Computational Thinking in all levels of education has been largely recognised in the latest years. Problem solving abilities are necessary to tackle all daylife activities since childhood. The spread of learning initiatives is a confirmation that fostering CT skills is basic for all students. In this view, the research aims at introducing computational thinking and coding using Arduino board. In particular, a learning path was defined to let the students acquire knowledge and abilities about the algorithm, the logical connective, the if-then-else and the while statements. The pilot study conducted in a junior high school confirms that the learning activities were engaging and motivating for the students and teacher. All students actively participated in all the proposed tasks, the teacher was surprised by some of them who usually show less interest in all kinds of proposed activities.

In the future, an experimentation with a larger sample should be conducted. Moreover, it would be interesting to record the difference among male and female performances, that in this case it has not been annotated. A more exhaustive learning pathway should be designed to ensure that all basic programming knowledge is acquired.

ACKNOWLEDGEMENT

We would like to thank the junior high school "Gesmundo – Moro – Fiore" in Terlizzi (Ba), in particular the principal Maria Chiapparino, the teacher Silvana Chiapparino, and finally all students involved in the study.

REFERENCES

Choi, B., Jung, J. & Baek, Y. (2013). In what way can technology enhance student learning?: A preliminary study of Technology Supported learning in Mathematics. In R. McBride & M. Searson (Eds.), *Proceedings of SITE 2013--Society for Information Technology & Teacher Education International Conference*. New Orleans, Louisiana, United States: Association for the Advancement of Computing in Education (AACE), pp. 3-9.

De Vries, C., et al., 2018. Using LEGO Kits to Teach Higher Level Problem Solving Skills in System Dynamics: A Case Study. *Advances in Engineering Education*, 2018, Vol. 6, N.3.

Maloney, J., et al., 2010. The scratch programming language and environment. *ACM Transactions on Computing Education* (TOCE), Vol. 10, N. 4, 16.

Marques, A., Guimarães, C., Salgado, A., (2019). Scratch 3 – Beginners programming course in 3rd year of primary school. *Lecture Notes in Electrical Engineering*, 505, pp. 1160-1166.

Haak, V., et al., 2018, Conception of a Lego Mindstorms EV3 simulation for teaching C in computer science courses. *In: Global Engineering Education Conference (EDUCON)*. IEEE, pp. 478-483.

Lye, S. Y., & Koh, J. H. L. (2014). Review on teaching and learning of computational thinking through programming: What is next for K-12?. *Computers in Human Behavior*, 41, 51-61.

Papert, S. 1980. *Mindstorms: Children, computers, and powerful ideas*. Basic Books, New York.

Papert, S., 1991. Situating constructionism. In *I. Harel & S. Papert (Eds.), Constructionism*. Norwood, NJ: Ablex, 1–11.

Plaza, P., et al., 2018. Traffic lights through multiple robotic educational tools. *In: Global Engineering Education Conference (EDUCON)*, IEEE, 2018, pp. 2015-2020.

Siegle, D. (2017). Technology: Encouraging creativity and problem solving through coding. *Gifted Child Today*, 40(2), 117-123.

Vaca-Cárdenas, L. A., Bertacchini, F., Tavernise, A., Gabriele, L., Valenti, A., Olmedo, D. E., ... & Bilotta, E. (2015). Coding with Scratch: The design of an educational setting for Elementary pre-service teachers. *In Interactive Collaborative Learning (ICL)*. IEEE, pp. 1171-1177.

Umbleja, K., 2017. Learning to Program with Lego Mindstorms–Difference Between K-12 Students and Adults. In: *International Conference on Interactive Collaborative Learning*. Springer, Cham, p. 447-458.

Wing, J. 2006. Computational thinking. *Communications of the ACM*, Vol. 49, N. 3, pp. 33–36