

Engaging Primary School Children in Computational Thinking: Designing and Developing Videogames

Involucrando a los niños de educación primaria en el Pensamiento Computacional: diseñando y desarrollando videojuegos

Giuseppe Chiazese, Giovanni Fulantelli, Vito Pipitone, Davide Taibi

Consiglio Nazionale delle Ricerche - Istituto per le Tecnologie Didattiche. Via Ugo La Malfa, 153, Palermo. Italy

{giuseppe.chiazese; giovanni.fulantelli; vito.pipitone; davide.taibi}@itd.cnr.it

<http://orcid.org/0000-0002-0228-6204>, <http://orcid.org/0000-0002-4098-8311>, <http://orcid.org/0000-0002-3780-8797>,

<http://orcid.org/0000-0002-0785-6771>

Abstract

This paper presents the results of a project on Computational Thinking education for primary school pupils. During the project – called *Computational Thinking for children education* – 81 students from a primary school in Italy have been guided to the design and development of computer games through the Microsoft Kodu game development platform. Different activities have been proposed to the pupils with the aim of promoting Computational Thinking abilities and skills. A narrative approach has been adopted throughout the project. Preliminary results of the educational experience highlight as the adoption of narrative learning and physical reproduction of manipulative programming objects provide an affordance for the development of Computational Thinking abilities. Furthermore, constancy during the learning process affects the acquisition of game development skills; finally, the design and implementation of computer games using Kodu have had a significant positive influence on the perception of computer programming.

Resumen

Este artículo presenta los resultados de un proyecto sobre Pensamiento Computacional en niños de primaria. Durante el proyecto, llamado *Computational Thinking for children education*, 81 estudiantes de una escuela primaria en Italia se han orientado al diseño y desarrollo de juegos de computadora a través de la plataforma de desarrollo de juegos Microsoft Kodu. Se han propuesto diferentes actividades a los niños con el objetivo de promover habilidades de Pensamiento Computacional. Se ha adoptado un enfoque narrativo a lo largo del proyecto. Los resultados preliminares de la experiencia educativa resaltan que la adopción del aprendizaje narrativo y la reproducción física de objetos de programación manipulativos proporcionan una oportunidad para el desarrollo de habilidades de Pensamiento Computacional. Además, la constancia durante el proceso de aprendizaje afecta la adquisición de habilidades de desarrollo del juego; finalmente, el diseño e implementación de juegos de computadora usando Kodu ha tenido una influencia positiva significativa en la percepción de la programación de computadoras.

Keywords

Computational Thinking; Kodu Game Lab; Primary school children education; Computational Thinking Tools

Palabras clave

Pensamiento computacional; Kodu Game Lab; Educación primaria; Herramientas para el pensamiento computacional

Recepción: 29-11-2017

Revisión: 05-02-2018

Aceptación: 29-05-2018

Publicación: 30-06-2018

1. Introduction

Wing defines Computational Thinking as “a universally applicable attitude and skill set everyone, not just computer scientists, would be eager to learn and use” (Wing, 2006). The definition argues for the educational importance of Computational Thinking and characterizes it as a general cognitive ability that is not strictly related to computer science and computer programming (García-Peñalvo & Mendes, 2018).

In particular, as Wing suggests, Computational Thinking includes: seeking algorithmic approaches to problem domains; a readiness to move between differing levels of abstraction and representation; familiarity with decomposition; separation of concerns; and modularity (Barr & Stephenson, 2011). Following Wing, many authors have pointed out the benefits of applying computational models and techniques to afford and solve problems in any discipline (IWG, 2010; Perkovi et al., 2010). García-Peñalvo points out that “computation thinking can be defined as the application of high level of abstraction and an algorithmic approach to solve any kind of problems” (García-Peñalvo, 2016; Pinto-Llorente, Casillas-Martín, Cabezas-González, & García-Peñalvo, 2018), thus highlighting the fundamental idea that Computational Thinking may be applied to various kinds of problems that do not directly involve computer programming (Wing, 2008).

The raising role of Computational Thinking in the pedagogical debate is strictly related to the concept of coding. Several authors have outlined the role of coding as an enabler of Computational Thinking (an extensive literature is available in [Lye & Koh, 2014; García-Peñalvo, Reimann, Tuul, Rees, & Jormanainen, 2016]). Through coding, children become computational thinkers able to analyze problems, identify algorithmic solutions, elaborate ideas, and write programs by composing set of instructions in a specific programming language. Coding can be considered as the language used by humans to express a task to be executed by a digital agent.

At the same level of writing, reading and arithmetic skills, coding is becoming a teaching priority in all educational curricula around the world. Consequently, as Barr and Stephenson (2011) note, algorithmic problem solving and computational methods and tools should be introduced in the K-12 curriculum. Starting from the primary school age, the art of programming changes the children perspectives from application users to application creators.

Today coding has become a more accessible and funny activity thanks to the visual and block languages paradigm (Bau, Gray, Kelleher, Sheldon, & Turbak, 2017). It opens up new opportunities for programming without struggling with the text-based coding. Scratch (Resnick et al., 2009), Code.org (Code, 2017), Blockly Games, Gameblox, Kodu Game Lab (Fowler, Fristce, & MacLauren, 2012)

are learning environments where beginners, supported by visual programming, speed up their developing skills to build complex and exciting applications. Visual programming languages allow school teachers to organize innovative educational activities aimed at promoting Computational Thinking processes and facilitating the learning of programming concepts (Harms, Balzuweit, Chen, & Kelleher, 2016; Akinola, Akinkunmi, & Alo, 2012). In particular, Kodu Game Lab includes features that we identify as specifically suitable in primary school context. In fact, it enables children to image 3D virtual worlds and to program characters in a virtual stage by composing instructions through a language based on visual tiles (Chiazzese & Laganà, 2011). The visual tiles have the affordance of the semantic meaning of the instruction. For example, tiles represent the sensory capabilities (e.g. to see, hear, touch) or actions (e.g. move, take, fire) of the characters. Moreover, Kodu Game Lab provides an easy to use, enjoyable, creative and highly accessible programming environment. From a teaching perspective, Kodu Game Lab provides interesting opportunities to design learning activities able to foster Computational Thinking processes (Touretzky, Marghitu, Ludi, Bernstein, & Ni, 2013).

Repenning, Basawapatna, & Escherle (2017) identify three stages of the Computational Thinking process that a Computational Thinking tool must elicit:

- Problem formulation (abstraction);
- Solution expression (automation);
- Execution and evaluation (analysis).

With respect to primary school pupils, literature has emphasized the relevance of narrative strategies to foster learning processes (Szurmak & Mindy, 2013). For this reason, we propose a model for stimulating Computational Thinking in K-12 education that extends the Repenning' stages of Computational Thinking Process by including a narrative stage as a preliminary level. The narrative stage is the description of a story to be implemented in the virtual stage.

According to the 4-stage model, reading a storyline represents the initial step of the Computational Thinking process. This task stimulates children creativity and acts a reading comprehension process for formulating an abstraction of the story (problem formulation). The problem formulation is the second stage of the process aimed at identifying key elements from the storyline: characters, protagonists, antagonists, virtual stage elements, game goals, actions of characters and scoring rules. The next stage is the formulation of the solution, which in turn represents the first step towards the development of the video game; this stage involves the process of coding identified key elements into instructions. The final stage consists in letting pupils play and test (execution and evaluation) their computer game in order to identify errors and evaluate results of their previous activities. The resulting model is shown in Table 1.

This model has been used to promote Computational Thinking abilities in pupils attending a primary school in Italy, in the framework of a research project called *Computational Thinking for children education*.

The next section presents a brief description of the project; section 3 reports the project methodology, materials and, finally, the project results are presented in section 4.

2. The “Computational Thinking for Children Education” Project

Between October 2016 and June 2017, 81 pupils from a primary school in Italy have participated to the research project *Computational Thinking for children education*. The project, funded by the Italian Ministry for Education, has involved the primary school “Daniele Ajello” located in Mazara del Vallo,

Stage	Description
Storyline (Narrative)	The narrative description of characters and events in a story
Problem formulation (Abstraction)	A formulation of the story in terms of actors, protagonists, antagonists, virtual stage elements, game goals, rules and characters’ actions
Solution expression (Automation)	<i>WHEN</i> something happens <i>DO</i> something (Kodu programming paradigm)
Execution and evaluation (Analysis)	The testing phase using the <i>Play action</i> of Kodu tool.

Table 1. The proposed pedagogical model to foster Computational Thinking abilities in primary school pupils

researchers from two institutes of the National Research Council of Italy, and a private research organization named Critica.

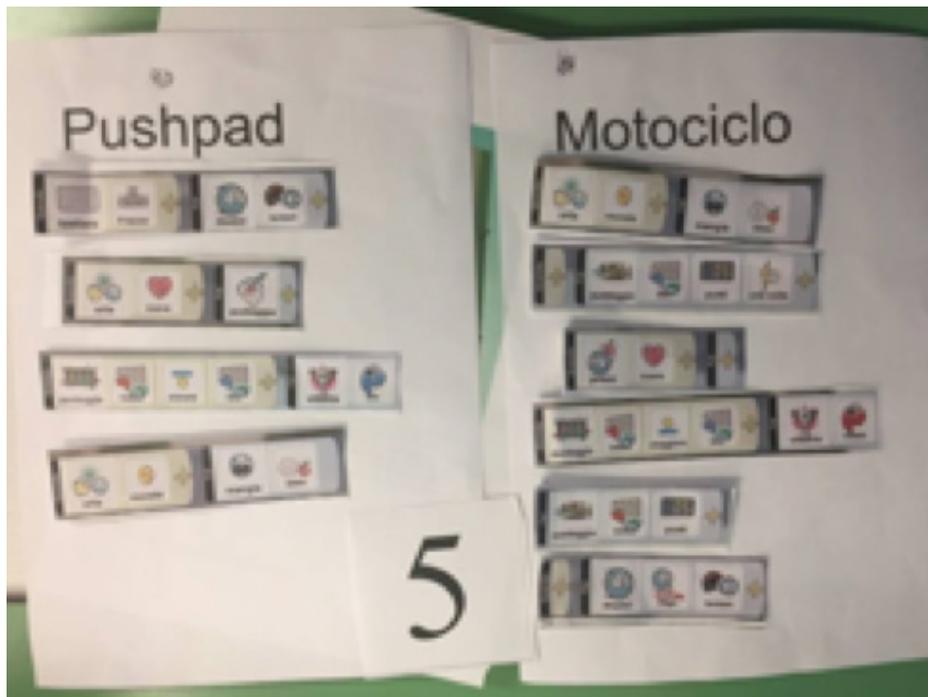


Figure 1. Two characters with their behavior described through Kodu tiles.

The project activities have been structured in two main phases. Each phase has been designed to meet a specific sub-objective, but all phases concur to the achievement of the aim of the project.

2.1. Phase 1: What is programming

The aim of the first phase has been the presentation of basic concepts of computer programming to the school children, by means of three open labs run at the National Research Council of Italy, respectively focused on:

- a) *Programming as a kind of game*: tools for teaching programming through game play.
- b) *Comparing Computer Code and Genetic Code*: differences and similarities of living and artificial organisms.
- c) *What if I want to build my video game?*

The overall activities lasted three days, one day for each open lab.

2.2. Phase 2: Kodu Coding Open Lab

This phase constitutes the core of the project. During the three introductory open labs school children have familiarized with coding; in the Kodu Coding Open Lab, they have learned programming principles and have been involved in practical activities.

The learning path has been implemented in a context where students have been engaged in a playful

learning activity that stimulates the principles of programming (Chiazzese, Cafari, Taibi, & Fulantelli, 2015). In fact, students have been engaged in a collaborative process aimed at designing and developing a video game by using the Kodu Game Lab environment.

Six sessions have been held at the computer lab of the school:

- 1 session (2 hours): introduction to visual programming and the Kodu environment. In this session, students have learned the basic concept of visual programming by manipulating physical tiles of Kodu language (Touretzky, 2014) (Figure 1).
- 1 session (2 hours): design of a virtual game scenario, definition of goals, choice of rules, construction of game actions, creation of score system.
- 3 coding sessions (6 hours): video game development and testing (Figure 2). Children have been guided in the autonomously construction of simple games starting from the narrative description. The three sessions have had increasing levels of difficulty.
- 1 evaluation session (2 hours): assessment of the programming skills acquired by pupils (Figure 3).

The three coding sessions have been informed by the 4-stage pedagogical model presented in the introductory paragraph (Figure 4).



Figure 2. Pupils during the coding session



Figure 3. A virtual world created by one of the pupil participating to the project



Figure 4. Results of the 4-stage pedagogical model

3. Methodologies

3.1. The social and territorial level of school

The *Computational Thinking for Children Education* project has involved the primary school “Daniele Ajello” located in Mazara del Vallo, in the South of Italy. It is the most important fishing port of the Country, one of the biggest in the Mediterranean Sea. In the 70s, the request of fishermen in some periods of the year attracted a large number of people from Tunisia, so that Mazara del Vallo hosted the largest Tunisian community of Italy.

The “Daniele Aiello” primary school is the reference point for developing several intercultural activities and inclusive projects in Italy. The school serves about 750 students, coming from different social and economic groups. 15% of students are 2nd or 3rd generation immigrants, and their families have been living in the city for a long time. These social factors have been taken into consideration in the developing of the learning paths, research design and evaluation of results.

3.2. Participants

Four 4th grade school classes have taken part in the project activities, with a total of 81 pupils (N=81).

Data analysis is based on a subsample of students, 29 male and 22 female, who attended more than 75% of the total number of sessions, and who completed the pre and post tests distributed at the beginning and at the end of the project activities.

3.3. Material and procedures

The following tools have been used to collect data:

- The pre-test and post-test included in the “Kodu Game Lab - Classroom Kit 1. “. These tests have been translated and adapted to the target. The tests have been used for evaluating the attitudes towards computer programming and their experiences in the creation and development of a video game.
- Three programming challenges named “The”, “Two unleashed pilots” and “The magical lake”.
- The “Q1 ELEMENTARI” questionnaires (De Beni, 1995), to assess the reading, numeracy and reasoning abilities of children.
- The final grades of pupils in Italian language and Math.
- The information on their socio-economics and familiar context.

This paper presents the results of the pre-tests and post-tests, and a qualitative analysis of the responses related to the development process of a video game. These data have been correlated with the performance of students in Italian and Math, and with their socio-economics familiar context.

3.4. Pre-test and post-test

The items of the pre and post tests are listed:

1. How many hours per day do you spend playing video games (on your computer or on your game console)?
2. Have you ever taken a computer programming class before?
3. I know a lot about computer programming
4. I know a lot about history
5. Do you think computer programming is:
 - a. creative
 - b. difficult
 - c. easy
 - d. funny
 - e. interesting
 - f. only for computer guru
 - g. Boring
 - h. Frustrating
 - i. Cool
 - j. Something I'd be good at
6. If your school offered a computer science class next year, would you take it?
7. Would you be interested in having a job in computer science one day?
8. What steps do you think are involved in the development of a video game?

The post-test includes most of the items in the pre-test, and add some further questions:

-
1. Does Kodu remind you of any subjects at school?
 - a. Math
 - b. Science
 - c. History
 - d. Art
 - e. Other (please explain)

 2. How much of your time have you spent developing the videogame and how much playing it in Kodu? (check one)
 - a. All my time has been spent for developing; I didn't play games at all.
 - b. Most of my time has been spent developing rather than playing.
 - c. I have spent equal amounts of time for developing and playing.
 - d. Most of my time has been spent developing rather than creating.
 - e. All of my time has been spent playing; I have not developed games at all.

 3. Developing games in Kodu has been fun
 4. Developing games in Kodu has been easy
 5. Playing games in Kodu has been fun
 6. Playing games in Kodu has been easy
 7. I know a lot about computer programming.
 8. I know a lot about History.
 9. Do you think computer programming is:
 - a. creative
 - b. difficult
 - c. easy
 - d. funny
 - e. interesting
-

- f. only for computer guru
 - g. Boring
 - h. Frustrating
 - i. Cool
 - j. Something I'd be good at
10. If your school offered a computer science class next year, would you take it?
 11. Would you be interested in having a job in computer science one day?
 12. What steps do you think are involved in the development of a video game?
 13. What did you like most about Kodu?
 14. What could be improved?

4. Results

The data collected during the project confirm the widespread idea that boys tend to spend more time on videogames than girls. In our sample, 55.2 % of boys spend between 1 and 2 hours versus 18.2% of girls; 17.2 % of males spend between 3 and 4 hours versus 4.5% of females). 36.4% of females do not play videogames and 9.1% of them does not own a digital device (figure 5).

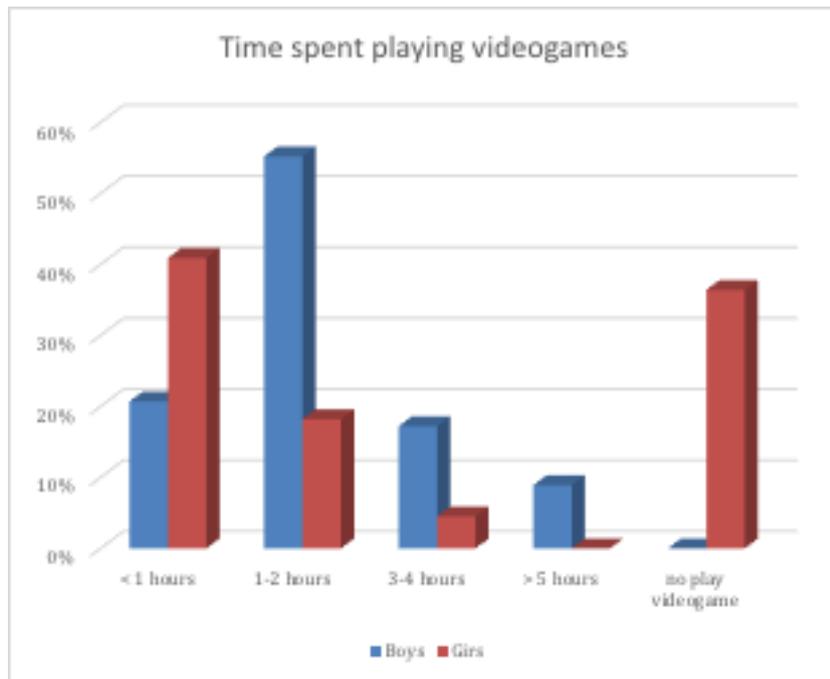


Figure 5. Time spent by girls and boys playing videogames

In order to evaluate if there is a correlation between time spent on videogames and individual aptitude to programming, we have measured students' perception of computer programming, and analyzed

their behavior and actions during the different activities of the project.

A Wilcoxon signed-rank test has been applied to measure if the learning activities have modified the students' perception of computer programming. The test has highlighted a significant difference ($p < .05$) in items d), i), a), j) of question "Do you think computer programming is" between pre and post tests (respectively $Z = -3.666$, $p = .000$; $Z = -2.041$, $p = .041$; $Z = -2.060$, $p = .01$; $Z = -3.267$, $p = .001$. Figure 6).

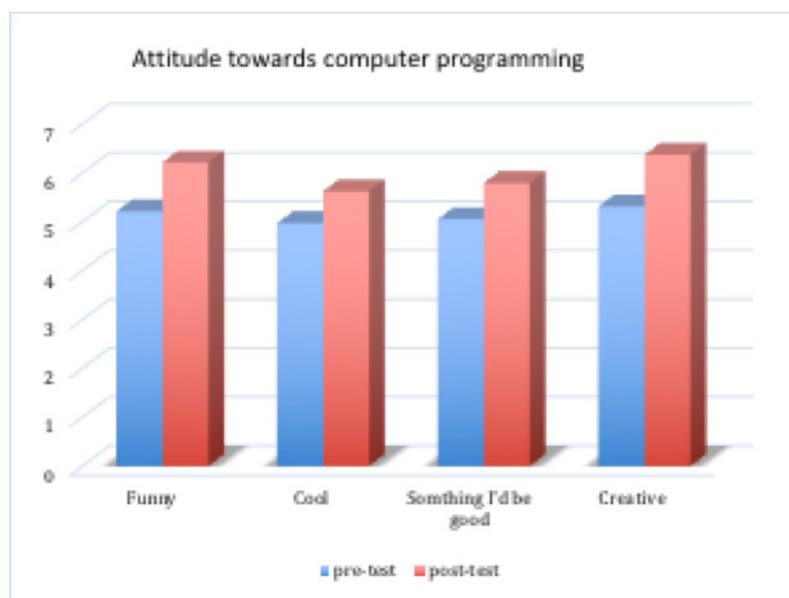


Figure 6. Attitude towards computer programming has increased during the project

In order to assess the awareness of programming processes, we asked students to describe the steps for creating their favorite videogame. Two external experts evaluated the quality of responses in terms of actions described (references to characters, game elements, actions to programs, steps order), giving a score (from 0 to 4) according to the completeness and accuracy of the answers. At the beginning of the project, students were unable to go beyond mentioning their favorite video game, without any further description related to design. This limited awareness appears to be quite transversal among students, given the reduced average variance. This result has changed in a significant way in the post-test, with a substantial increase in mean scores, ranging from 0.9 to 2.1, and a parallel increase in average variance, from 0.6 to 1.8 (figure 7).

The increase in the level of awareness among students appears quite obvious at the end of the project. However, it raises some specific questions on the categories of the students who have most benefitted from the project activities. Can these improvements be linked to the gender or socio-economic conditions of the students' family? Furthermore, higher variance of the results must be explored. Can the variance increase be connected to the level of participation in the project?

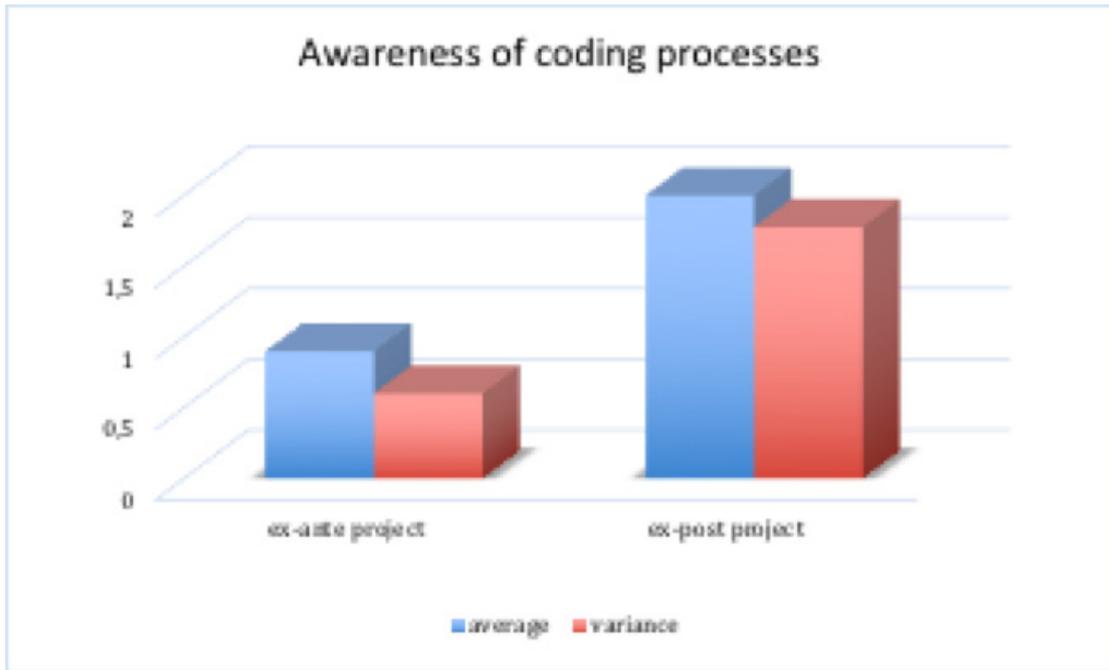


Figure 7. Awareness of coding processes

Girls are the group that gets the most positive outcomes from the project activities (boys got a mean score increase of 0.8, girls got an increase of 1.1; figure 8). This confirms that the assumption that boys are better than girls at programming is a stereotype, as already proved by other authors (Master et al., 2017). Furthermore, this result indicates that programming activities can be strongly stimulated through specific learning paths.

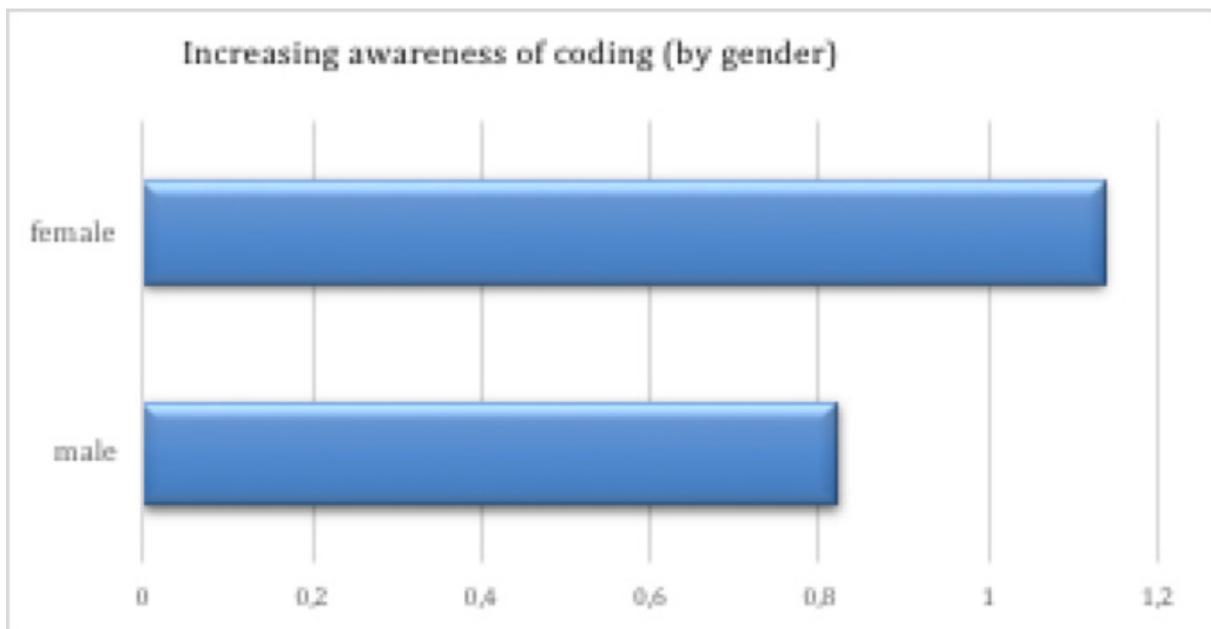


Figure 8. Increasing awareness of coding (by gender)

The data shows a clear prevalence of positive results among children born in Italy (figure 9). It is useful to underline that the distinction between born in Italy and born abroad, in the case of Mazara del Vallo, does not coincide with a distinction between Italians and foreigners. In fact, Tunisians living in the town are mostly second and third generation immigrants. Rather, the distinction aims at capturing a proxy of the “economic, social and cultural status” of the families of children born abroad. Actually, many empirical studies highlight a high correlation between students’ abilities and socio-economic and cultural life of their parents (Haveman & Wolfe, 1995; Sirin, 2005; Becker & Tomes, 1986). Partly, this depends on the fact that families with better “status” can stimulate higher competencies in their children, by providing a greater number of educational resources and teaching tools and providing behavioral patterns conducive to learning and consolidation of skills (Xia, 2010).

Furthermore, the distinction between born in Italy and born abroad aims to capture another dimension of the learning process: the language used for learning. In fact, as reported in literature, lower scores of new immigrants is strictly related to the difficulty to learn using a foreign language (Schnepf, 2007).

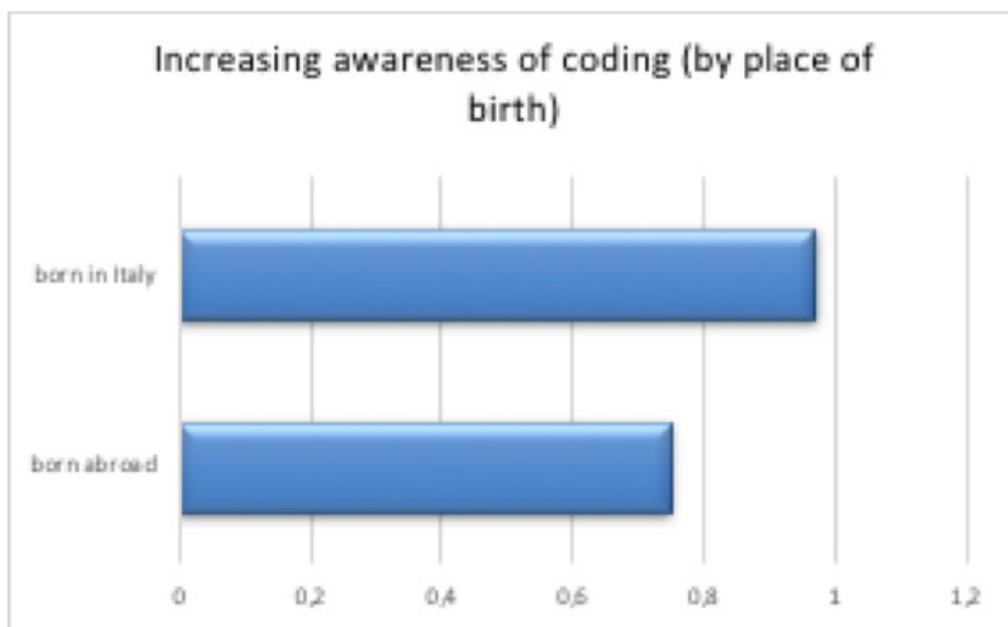


Figure 9. Increasing awareness of coding by place of birth

The increase in the awareness of programming processes has been correlated with pupils’ school achievements (measured through the grades received at the end of the year in Italian language and in Math). This analysis has highlighted that the pupils that have most benefited from the projects are the ones that do better at school (figure 10). This seems to indicate a strict correlation between school proficiency and Computational Thinking skills.

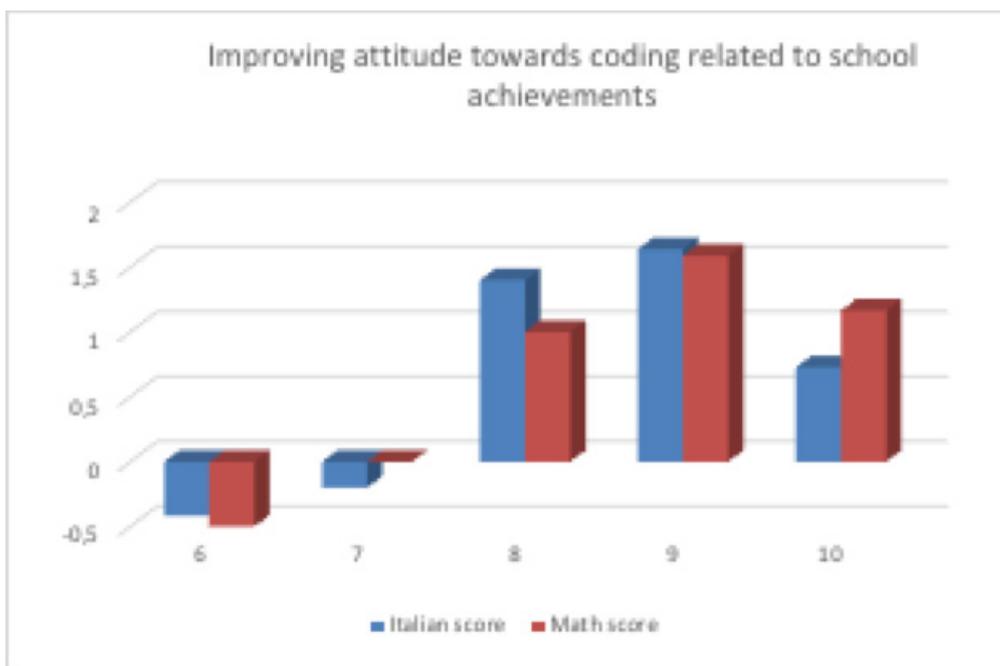


Figure 10. Improving attitude towards coding related to school achievements

In order to verify how participation in the project activities have influenced on the pupils' learning outcomes, we have distinguished between regular students (pupils with an attendance rate higher than 80%) and occasional students. Then we have analyzed and compared how the two groups have answered to the questions reflecting the exercise comprehension (e.g. the correct identification of: the protagonists of the game, the antagonists, the objects, the actions, the scoring rules and the final objectives).

The results have shown that regular attendance significantly improves Computational Thinking skills (figure 11). This improvement becomes more and more significant as the training path proceeds.

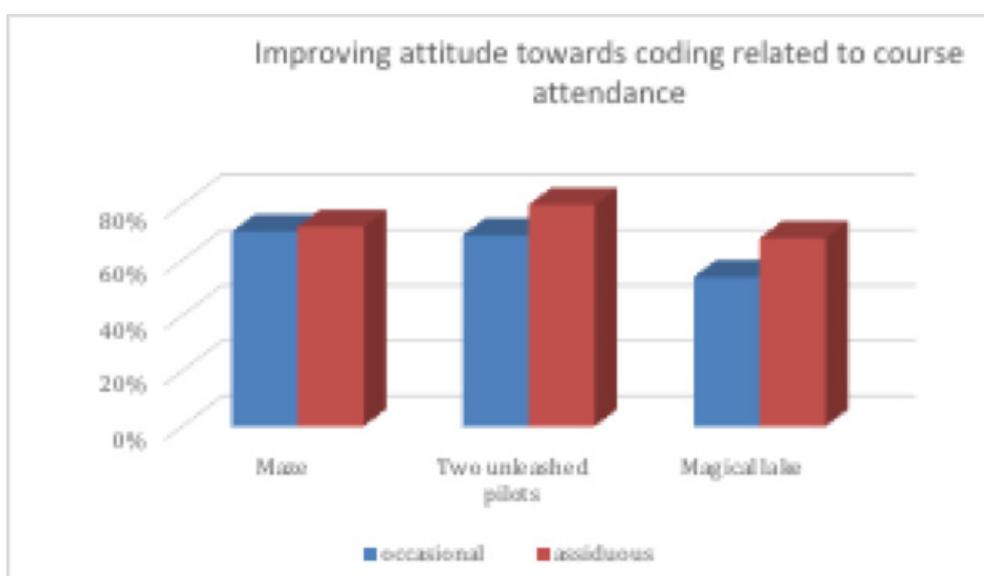


Figure 11. Improving attitude towards coding related to course attendance

5. Conclusions

Computer programming is most frequently considered a significant enabler for Computational Thinking skills, even in primary school students. The results discussed in this study provide evidence about the effects of practical coding sessions on the development of Computational Thinking skills. The experimentation has been conducted by adopting a narrative approach in the Kodu environment. The analysis of pre- and post-tests has pointed out an improvement on the perception of computer programming in primary school children, especially with respect to the items *funny*, *cool*, *exciting*, *creative* and *"Something I'd be good at"*.

Furthermore, we have observed that high school achievements and the family context positively influence coding abilities and, in turn, Computational Thinking skills. In particular, students with higher marks in Math and Italian language have achieved higher educational outcomes than their peers with lower marks.

Our research has already confirmed that coding is not a discipline for males only, as a stereotyped vision of the society has suggested in the past; on the contrary, although girls do not use video games as frequently as boys do, they have a higher attitude for programming than boys, and have achieved higher learning outcomes from the project. Finally, the coding ability is strictly related to the attendance frequency, and this relationship appears increasingly stronger as students' progress through the learning path.

The results presented in this paper are to be considered as the preliminary outcomes of an in-progress research study. Subsequent works will allow the authors to further investigate the learning activities and identify the project phases that have better contributed to increase computational skills in the pupils.

6. Acknowledgments

All primary school children participating to this learning experience, their teachers and the principal of the school "Daniele Ajello".

7. References

Akinola, O. S., Akinkunmi, B. O., & Alo, T. S. (2012). A Data Mining Model for Predicting Computer Programming Proficiency of Computer Science Undergraduate Students. *African Journal of Computing & ICT*, 5(1), 43-52.

-
- 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:<https://doi.org/10.1145/1929887.1929905>
- Bau, D., Gray, J., Kelleher, C., Sheldon, J., & Turbak, F. (2017). Learnable programming: blocks and beyond. *Communications of the ACM*, 60(6), 72-80. doi:<https://doi.org/10.1145/3015455>
- Becker, G., & Tomes, N. (1986). Human capital and the rise and fall of families. *Journal of Labor economics*, 4(3), S1-S39. doi: <https://doi.org/10.1086/298118>
- Chiazzese, G., & Laganà, M. R. (2011). Online learning with virtual puppetry. *Journal of e-Learning and Knowledge Society* 7(3), 121-129. doi:<https://doi.org/10.20368/1971-8829/557>
- Chiazzese, G., Cafari, A. M., Taibi, D., & Fulantelli, G. (2015). EduCodeGame - Risolvere problemi di matematica con Kodu: uno studio Pilota. *Proceedings of Teach Different! EMEMITALIA conference*.
- Code. (2017). Hour of Code website. Retrieved 26 April 2018 from <http://code.org/learn>
- De Beni, R., & Gruppo MT. (1995). "Q1 ELEMENTARI" - Prove per la compilazione del profilo iniziale del nuovo documento di valutazione. Organizzazioni Speciali.
- Fowler, A., Fristoe, T., & MacLauren, M. (2012). Kodu game lab: A programming environment. *The computer Games Journal* 1(1), 17-28. doi:<https://doi.org/10.1007/BF03392325>
- García-Peñalvo, F. J. (2016). What computational thinking is. *Journal of Information Technology Research*, 9(3), v-viii.
- García-Peñalvo, F. J., Reimann, S., Tuul, M., Rees, A., & Jormanainen, I. (2016). An overview of the most relevant literature on coding and computational thinking with emphasis on the relevant issues for teachers. Belgium: TACCLE3 Consortium. doi:<https://doi.org/10.5281/zenodo.165123>
- García-Peñalvo, F. J., & Mendes, J. A. (2018). Exploring the computational thinking effects in pre-university education. *Computers in Human Behavior*, 80, 407-411. doi:10.1016/j.chb.2017.12.005
- Harms, K. J., Balzuweit, E., Chen, J., & Kelleher, C. (2016). Learning programming from tutorials and code puzzles: Children's perceptions of value. In *Proceedings of the 2016 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)* (pp. 59-67). USA: IEEE. doi:<http://doi.org/10.1109/VLHCC.2016.7739665>
- Haveman, R., & Wolfe, B. (1995). The determinants of children's attainments: a review of methods and findings. *Journal of Economic Literature* 33(4), 1829–1878.
- IWG, (2010). Computational Thinking for Youth, Education Development Center, Inc. Newton, MA.

-
- Lye, S. Y., & Koh, J. H. L. (2014). Review on teaching and learning of computational thinking through programming. *Computer Human Behaviour*, 41, 51-61. doi:<https://doi.org/10.1016/j.chb.2014.09.012>
- Master, A., Cheryan, S., Moscatelli, A., & Meltzoff, A. N. (2017). Programming experience promotes higher STEM motivation among first-grade girls. *Journal of Experimental Child Psychology*, 160, 92-106. doi:<https://doi.org/10.1016/j.jecp.2017.03.013>
- Perkovi, L., Settle, A., Hwang, S. and Jones, J. (2010). A Framework for Computational Thinking across the Curriculum. In *Proceedings of the 2010 Conference on Innovation and Technology in Computer Science Education*, (pp. 123-127). doi:<https://doi.org/10.1145/1822090.1822126>
- Pinto-Llorente, A. M., Casillas-Martín, S., Cabezas-González, M., & García-Peñalvo, F. J. (2018). Building, coding and programming 3D models via a visual programming environment. *Quality & Quantity*, In Press doi:10.1007/s11135-017-0509-4.
- Repenning, A., Basawapatna, A. R., & Escherle, N. A. (2017). Principles of Computational Thinking Tools. In P. Rich, & C. Hodges (Eds.), *Proceedings of Emerging Research, Practice, and Policy on Computational Thinking* (pp. 291-305). Cham: Springer International Publishing. doi:https://doi.org/10.1007/978-3-319-52691-1_18
- Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., Silver, J., Silverman, B., & Kafai, Y. (2009). Scratch: programming for all. *Communications of the ACM* 52(11), 60-67. doi: <https://doi.org/10.1145/1592761.1592779>
- Schnepf, S. V. (2007). Immigrants' educational disadvantage: an examination across ten countries and three surveys. *Journal of Population Economics*, 20(3), 527-545. doi:<https://doi.org/10.1007/s00148-006-0102-y>
- Sirin, S. (2005). Socioeconomic status and academic achievement: A meta-analytic review of research. *Review of Educational Research*, 75(3), 417-453. doi:<https://doi.org/10.3102/00346543075003417>
- Szurmak, J., & Mindy, T. (2013). Tell me a story: The use of narrative as a tool for instruction. In *Imagine, Innovate, Inspire: The Proceedings of the ACRL 2013 Conference* (pp. 546-552). Indianapolis, IN, USA: ACRL.
- Touretzky, D. S. (2014). Teaching Kodu with physical manipulatives. *ACM Inroads* 5(4), 44-51. doi: <https://doi.org/10.1145/2684721.2684732>
- Touretzky, D. S., Marghitu, D., Ludi, S., Bernstein, D., & Ni, L. (2013). Accelerating K-12 computational thinking using scaffolding, staging, and abstraction. In *Proceeding of the 44th ACM technical symposium on Computer science education, SIGCSE '13 (Denver, Colorado, USA – March 06 - 09, 2013)* (pp. 609-
-

614). USA: ACM. doi:<https://doi.org/10.1145/2445196.2445374>

Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33-35. doi:<https://doi.org/10.1145/1118178.1118215>

Wing, J. M. (2008). Computational thinking and thinking about computing. *Philosophical Transactions of the Royal Society a-Mathematical Physical and Engineering Sciences*, 366(1881), 3717-3725.

Xia, N. (2010). Family factors and student outcomes. Ph.D. Dissertation. Pardee RAND Graduate School, Santa Monica, CA. Retrieved 26 April 2018 from https://www.rand.org/pubs/rgs_dissertations/RGSD256.html