Investigating the Learning Impact of Autothinking Educational Game on Adults: A Case Study of France

Nour El Mawas[1], Danial Hooshyar[2] and Yeongwook Yang[2]
[1]*CIREL (EA 4354), University of Lille, Lille, France*
[2]*Institute of Education, University of Tartu, Tartu 50103, Estonia*

Keywords: Technology Enhanced Learning, Computational Thinking, Educational Game, Adaptive Learning, Adult Learning.

Abstract: Adults have different needs for education and training throughout their lives in order to maintain and progress in their job or find a new one. Nowadays, Computational Thinking is one of the 21st century skills that adults must acquire and develop. In this context, some adults have difficulties to find new teaching and learning methodologies that help them learn Computational Thinking. Technology Enhance Learning and specifically Educational Games give the opportunity to learners to enhance their Computational Thinking skills and conceptual knowledge. This paper presents a research study on the learning impact of an adaptive educational game, called AutoThinking, developed for promoting Computational Thinking skills and conceptual knowledge. The game was used by adults in a Master class at the Université de Lille in France. Pre- and Post-tests results analysis has shown that the game helped the adults to acquire knowledge on the Computational Thinking: 92% of adults have answered correct at least 4 questions out of 7 in the post-test versus only 34% of learners in the pre-test.

1 INTRODUCTION

These days, an individual will have a wide range of employment opportunities during his/her lifetime. Lifelong learning is becoming a central asset, beginning with the university and continuing through the professional career with different jobs. Adults have different needs for education and training throughout their lives (El Mawas et al., 2017). Many job seekers/employees find themselves in need of acquiring or improving their technology skills to maintain and progress in their jobs or find new career opportunities. Computational Thinking (CT) skills are among those skills (El Mawas et al., 2018) that adults need to keep up-to-date according to the OECD (2013).

CT is defined as the mental ability enabling learners to develop a computational solution for a problem in hand (Wing, 2006). In other words, CT is a cognitive ability reflecting the application of key reasoning process and concepts of computer science into science, technology, engineering, and mathematics (STEM) domains, as well as wide range of problems and activities in everyday life (Wang, 2016).

As a practical skill computer programming shares common and similar ideas with CT's construct as a cognitive ability, such as concept of sequence, loops, conditionals, and parallelism. Additionally, CT involves some key cognitive counterparts of computer programming concepts, namely algorithmic thinking, decomposition, conditional logic, pattern recognition, debugging, simulation, and generalization. As stated by the founder of CT, Wing (2006), CT is not computer programming (coding in particular), and instead it refers to problem solving by way of computing. More specifically, CT's products are ideas and concepts used to approach and solve problems, and it starts before writing the code. Given the fact that CT denotes a general and applicable problem solving strategy for wide range of domains, it has been highlighted as one of the main and fundamental 21st century skill (Wing, 2008).

Several research have shown that learners' analytical skills could potentially be improved by teaching CT concepts and skills, and possessing such abilities could possibly be seen as indication of

El Mawas, N., Hooshyar, D. and Yang, Y.
Investigating the Learning Impact of Autothinking Educational Game on Adults: A Case Study of France.
DOI: 10.5220/0009790301880196
In *Proceedings of the 12th International Conference on Computer Supported Education (CSEDU 2020) - Volume 2*, pages 188-196
ISBN: 978-989-758-417-6

learners' academic success (Haddad & Kalaani, 2015). Thus, similar to numeracy and literacy, CT is considered as a vital competence for everyone, not just computer scientists, that should be acquired and taught early in education. Recently, several reformations and adaptations of educational programs have taken place in different education level all over the world as both cognitive and non-cognitive benefit of integration of CT into educational curricula is indicated by many research (e.g., Brown, Sentance, Crick, & Humphreys, 2014; Repenning et al., 2015). For instance, several recent references related to governmental institutions and educational programs have highlighted that CT is being added to primary, secondary, higher educational programs, and adult learning all over the world (European School Network, 2020) (OECD, 2013). However, there exists two major challenges in fostering CT which are lack of motivation and opportunities to improve learners' CT skills. To this end, some research show that school learners usually show negative attitude toward learning CT, hindering proper development of CT skills (e.g., Yardi & Bruckman, 2007). To approach these issues, different methods have been employed to make CT more accessible to learners, educational games among them.

Educational games have gained a lot of attentions lately as they have proven to be effective learning tools engaging and motivating learners (El Mawas et al., 2019). Findings from several research show that educational games are capable of bringing about improvements in both learners' motivation and learning achievements (Hooshyar et al., 2018a). Although there exists several educational games for fostering CT, they chiefly ignore promoting CT skills (as such) and providing adaptivity in game-play and teaching process (Kazimoglu et al., 2012). Instead, they reinforce CT's theoretical knowledge while promoting learners' motivation. What's more, they mostly follow predefined and rigid computer-assisted instruction concepts (ignoring adaptivity which considers individual needs and characteristics) making them fall short when it comes to different player's needs. Regarding the former issue (ignoring CT skills), while educational games developed for promoting CT indeed improve abstract and theoretical knowledge, they do not provide learners with opportunities to develop their CT skills (Kazimoglu et al., 2012).

Basically, in games with focus on improving CT abstract and theoretical knowledge, contextual relationship between the focus of the game and the knowledge being acquired is of less importance and may even be completely abstract, providing less opportunities to develop CT skills. On the other hand, games that aim to teach CT skills offer opportunities to practice the conceptual knowledge through game-play. Thus, we must distinguish between games that target teaching applied knowledge and skills, and those that aim reinforcing theoretical knowledge. In terms of the latter issue of CT games (ignoring adaptivity), despite several calls urging researchers and practitioners to pay more attention to adaption and personalization to the individual needs, existing CT games mainly follow unadaptable and rigid computer-assisted instruction concepts, resulting in plaguing the full educational potential of computer games (e.g., Kickmeier-Rust et al., 2011; Hooshyar et al., 2018b).

Given the societal relevance and importance of CT, and the existing gaps in CT game research that undermine their educational potential, we developed an adaptive game for teaching both CT concepts and skills, engaging learners with individually tailored gameplay (called AutoThinking) (Hooshyar et al., 2019). To evaluate the effectiveness of our proposed game, in this study, we design and conduct a study to investigate possible effect of AutoThinking on adults. This research work is dedicated to Education and Computer Science active communities and more specifically to directors of training centres / CT teachers, and lifelong learners who meet difficulties to learn CT concepts.

The outline of this paper is as follows: Section 2 reviews the related studies in the area of educational games research aimed at fostering CT. Section 3 presents our AutoThinking game, while Section 4 illustrates our case study and the results analysis. Section 5 offers conclusion of this study.

2 RELATED WORK

Because computer programming shares common and similar ideas with CT's construct as a cognitive ability, several learning environments use programming, coding in particular, to teach CT to learners (Grover & Pea, 2013). Most of these environments use block-based and visual programming environments, or adapt game design principles to reduce the complexities associated to programming languages syntax by simplifying it down to drag-and-drop interactions. Some example of such environments are Scratch (Resnick et al., 2009), Snap! (Harvey & Mönig, 2010), and Blockly (Fraser, 2013). Even though these approaches have shown some success in improving learners' motivation in programming activities and CT, they fall short when it comes to promoting deeper learning (e.g., Brennan & Resnick, 2012; Meerbaum-Salant, Armoni, & Ben-Ari, 2011). One reason is that even though CT's main focus is conceptualization and underlying taught processes of solving a problem not coding, using these environments learners still get distracted and overwhelmed by syntax of programming languages presented to them in different forms (e.g. blocks). In other words, alignment of these environments with CT skills is incomplete. Furthermore, though such environments rely on game design principles and are often named as games for fostering CT, they cannot be considered as educational games as they lack several essential elements of educational games, such as timely feedback, encouraging engagement, improving retention, and incentives.

Educational games which are well-known vehicles for developing many different skills in education and proven to be effective learning tools have also been developed and used for developing learners' CT knowledge (e.g. Weintrop & Wilensky, 2012). Usually, educational games aimed at fostering CT use motivating context to engage learners in process of developing solutions to solve a problem (e.g. Kazimoglu et al., 2012). Compared to block-based or visual programming environments (or designed-based learning environments), such educational games have a capacity to foster more purposeful learning with richer learning support through different game elements (e.g., Land, 2000). For instance, Eagle and Barnes (2009) developed an educational game called Wu's Castle; Esper, Foster, Griswold, Herrera, and Snyder (2014) developed CodeSpell; and Ayman, Sharaf, Ahmed, and Abdennadher (2018) developed MiniColon for teaching programming and promoting CT. Even though these games are reported to be useful for developing learners' CT and a number of studies on these games found their positive impact on learners programming and CT learning, they are not aligned with CT as they employ a text-based programming language that begs a substantial attention of learners to syntax details (Zhao & Shute, 2019).

On the other hand, such educational games still mostly suffer from two issues: ignoring development of CT skill of learners and adaption to each learners' need. In regards to the former, educational games aimed at fostering CT mainly reinforce CT's theoretical knowledge while promoting learners' motivation, providing less opportunities to develop CT skills. Concerning the latter issue, games for fostering CT mainly ignore adaption and personalization to the individual needs. In other words, such games follow unadaptable and rigid computer-assisted instruction concepts, resulting in plaguing the full educational potential of computer games. In brief, research has shown promising results concerning application of educational games to CT among learners. However, there still exist some room for improvement of such games. To improve the existing games, we developed an adaptive CT game engaging users with individually tailored gameplay and learning process that helps to foster both learners' CT concepts and skills.

3 THE AutoThinking GAME

3.1 Overview of the Game

AutoThinking (http://www.autothinking.ut.ee/) is an adaptive educational game developed for promoting CT skills and concepts (Hooshyar et al., 2019). It uses icons rather than syntax of computer programming languages in order to exclude syntactical errors, reducing the cognitive load of learners (see Figure 1). AutoThinking, to the best of our knowledge, is the first adaptive educational game developed for promoting CT that includes adaptivity in both game-play and learning process. It, in a novel way, promotes four CT skills, namely problem identification and decomposition (algorithmic thinking), algorithm building (pattern recognition and generalization), debugging, and simulation. What's more, it fosters three CT concepts, including sequence, conditional, and loop.

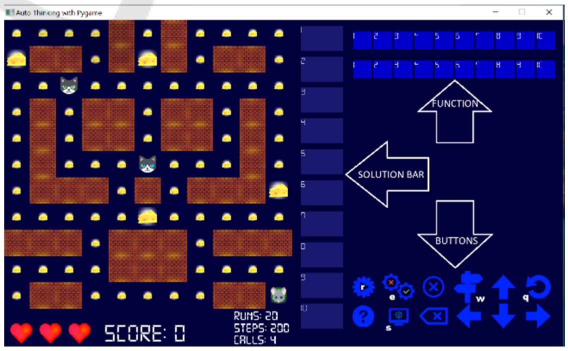


Figure 1: AutoThinking's interface.

In brief, AutoThinking includes three levels where players should, in a role of a mouse, develop different types of strategies or solutions to—collect as many cheese and score as possible, and scape from two cats in the maze—complete or win the level. Players are given opportunity to develop up to 20 solutions for clearing all 76 cheeses on the maze. During the game-play, players receive more score for solutions that involve various CT concepts or skills compared to traversing empty tiles, or only using simple solutions. Note that players are provided with various options in the game to develop different types of solutions, for example, they can use the “function bar”, see Figure 1, to save various patterns, and if necessary apply or generalize them in different situation of the game. What’s more, before developing or running solutions, players should thoughtfully and carefully observe the movement of both cats and consider the risk of running their solution for the current state of the maze. Note that one cat moves randomly through the maze according to the number of commands placed by the player in the “solution bar” (e.g., a solution that is appropriate for the current state of the maze might be inappropriate for another situation), whereas the other cat moves intelligently according to the number of tiles traversed by the mouse and the quality of the developed solutions (skill of players). According to the suitability of solutions for the current state of the maze, players are adaptively given various type of feedback (textual, graphical, or video) and hints.

Several activities and features in AutoThinking game are designed and embedded to target and promote different CT skills and concepts. These include “function bar” to encourage players to construct generalizable patterns where they can be used in different situations of the game (targeting algorithmic thinking and pattern recognition skill); “debug” button enabling players to monitor their solution algorithm and possibly detect any potential errors in their logic (practicing debugging skill); “simulation” button to allow players to simulate their solution before actually executing it to observe the outcome of their solution regardless of intervention of other variables in the game, such as cats movements and cheeses (practicing run time mode or simulation skill); “solution bar” to help players to develop different solutions for different situations of the maze, or different problems, using sequence of proper actions (targeting both problem-solving and sequence); “loop” button to run the same sequence of actions multiple times (practicing loop concept); and finally “conditional” button to enable player to make decisions based on certain decisions that supports expression of multiple outcomes (practicing conditional concept).

3.2 Adaptivity in Game-play

During the game-play, one of the cats moves intelligently according to the quality of the developed solution by the player. To do so, it considers whether the solution has the potential to gain enough score, whether it is risky and the mouse might get caught by cats, and whether players used proper CT skills or concepts in their developed solution according to the current state of the maze. Accordingly, a decision-making technique used in the game—provided by a probabilistic model, Bayesian Network, that automatically assesses player’s skills—regulates the movement of the cat by switching between the following algorithms:

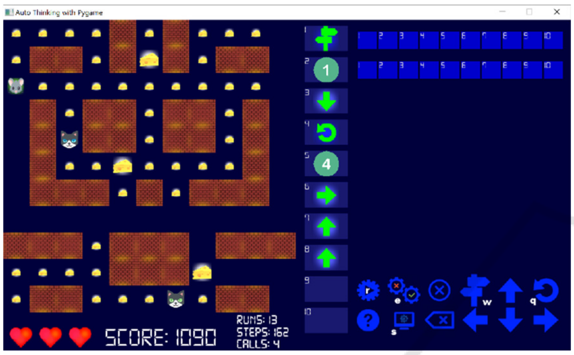
- The cat decides to move randomly without iteration through the maze.
- The cat decides to move aggressively aimed at catching the mouse (by finding the shortest distance from the mouse).
- The cat decides to move provocatively by going close to the mouse (up to one tile away), not to catch it, and come back.
- The cat decides not to get closer than 6 tiles away from the mouse.

Observe that the cat decides to choose a more appropriate algorithm to use for its movements according to both short term and long term solution of the player. In other words, it considers both the current solution developed and also previous solutions developed by the player. However, another cat still moves randomly with repetition according to the number of commands used in the solution, making AutoThinking an unpredictable and never-ending game that always provides player with a new situation that might have never happened for previous players.

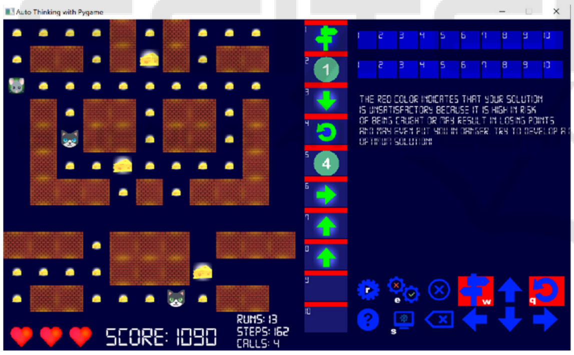
3.3 Adaptivity in Learning

While playing the game, the automatic short and long term assessment of the players enables the game to provide them with timely feedback and hints.

According to the skill level of the players and current status of the maze, the game offers textual, graphical, or video feedback about CT concepts and skills that are embedded in the game-play. It also highlights some of the game features or buttons as a hint, enabling players to improve their solutions according to both the hint and feedback (see Figure 2). This phase of adaptivity takes place in two different timings, before or after running the solution. Regarding the former one, after players have developed their solution they can use the “debug” button—which activates the probabilistic model used for decision-making—to see the estimation of the suitability of their solution in a form of timely adaptive feedback or hints. Doing so provides player with a chance to, if necessary, change and improve their solution so as to have a more optimum solution. Alternatively, concerning the latter situation, player can skip using the "debug" option and directly "run" the game. This results in timely adaptive feedback or hint, after running the game, which would help players know about their shortcomings and mistakes in previous solutions and possible ways to overcome them. Such adaptivity—which aim to foster both learners' problem-solving (algorithmic thinking) and pattern recognition skills—individually support learners in developing the most optimum solution for the problem in hand.



(a)



(b)

Figure 2: (a) A solution developed by a player, (b) Textual feedback and hint generated by the game.

4 CASE STUDY

The goal of the research study was to investigate the learning impact of the AutoThinking game in the class to teach CT skills and concepts.

This section presents the evaluation methodology applied, case study set-up, and results analysis of the collected data.

4.1 Research Methodology

As this research study is focused on knowledge acquisition aspect of game-based learning in the AutoThinking game, pre- and post-test assessments were run before and after the use of the game.

Table 1: The pre-test questions deployed before the game.

Question	Answer
Q1. What is a sequence?	- A piece of code that is written over and over again - The small shiny things sewn onto clothes for a fancy effect - An error in the coding language - The order of events that the computer will complete
Q2. _____ is about analyzing and identifying repeated sequences	- Type answer: ____________ - I don't know
Q3. The action of doing something over and over again is "conditionals".	- True - False - I don't know
Q4. The if, elif, else statement is used for _____	- Selection. - Iteration. - Indentation. - Printing. - I don't know.
Q5. _____ is a named group of programming instructions. They are reusable abstractions that reduce the complexity of writing and maintaining programs.	- Function - Loop - Repeat - Algorithm - I don't know
Q6. Debugging is	- What an exterminator does. - Rewriting code to make it less complex. - Finding and fixing problems in an algorithm or program. - A girl named Dee annoying everyone. - I don't know.
Q7. Simulation is	- A model that's used to see how a specific process will work. - A full-scale working model used to test a design to see if it solves the problem it was created to. address - A graph that uses vertical or horizontal bars to show comparisons among two or more items. - A graph that uses line segments to show changes that occur over time. - I don't know.

Table 2: The post-test questions deployed after the game.

Question	Answer
Q1. A sequence is the order in which the commands are given.	- True - False - I don't know
Q2. Define Pattern Recognition.	- A sequence of instructions. - Looking for similarities and trends. - Breaking a task into smaller tasks. - Focusing on what is important and ignoring what is unnecessary - I don't know
Q3. _____ is the action of doing something over and over again	- Type answer: _____________ - I don't know
Q4. Which of the following instructions allows a program to search a list of options and make a decision?	- If - Select - Function - Choose - I don't know.
Q5. A piece of code that includes the steps performed	- Command - Execute - Function - Iteration - I don't know
Q6. Finding and fixing problems in an algorithm or program.	- Sequencing - Debugging - Conditionals - Behavior - I don't know
Q7. Simulation is, essentially, a program that allows the user to observe an operation through simulation without actually performing that operation	- True - False - I don't know

The research methodology applied in this case study involved 12 students from the Digital Learning Management Master. Note that students in this Master class are adults and they do not have any course about CT. All students learned about the CT by playing the educational game. The learning process took place during the university study hours. All the tests were implemented in the online survey tool, Lime Survey, and provided to learners online via Moodle. The case study consisted of several phases which cover the collection of assent and consent forms, description of the realised course, special pre-questionnaires, knowledge pre-test, learning experience, knowledge post-test, and other post-questionnaires. In this paper, we are interested in the knowledge pre- and post-tests.

Each learner played the game individually in the computer room with a teacher present in the room, but the teacher did not answer any question related to the subject. In order to evaluate learners' level of knowledge on the subject prior the particular pedagogical approach all students did the same pre-test. Similarly, the same post-tests were provided to all students to analyse and evaluate level of acquired knowledge. Tables 1 and 2 show questions of pre- and post-test applied during the experimentation. The pre- and post-tests creation followed requirements such as they should last max. 10 minutes, both tests should have very similar content (Table 3) and identical concept. These tests consist of a single choice and simple answer questions.

Based on knowledge tests results an average score can be calculated for students. By comparing average pre-test and post-test scores a knowledge gain can be calculated.

Table 3: The addressed concept in each question.

Question (pre- and post-test)	Concept
Q1	Sequence
Q2	Pattern recognition
Q3	Loop
Q4	Conditional
Q5	Function
Q6	Debugging
Q7	Simulation

4.2 Results Analysis

The research focuses on the knowledge acquisition while students play the game. The evaluation was based on the results of knowledge tests (pre- and post-tests).

Final results showing the level of learner' knowledge in percentage are depicted in Figure 3. The AutoThinking game increases knowledge level of learners by 21.4%. More specifically, we can notice that the game improves the sequence concept by 33%, the pattern recognition concept by 25%, the conditional concept by 16%, the function concept by 42%, and the simulation concept by 50%. However, the knowledge about the loop and debugging concepts were slightly decreased by 8%. One possible explanation could be that some students could not properly read or understand the feedback and hints provided by the game due to several reasons, e.g., language barrier. Additionally, some students may have ignored using the “debug” button, which offers chance to monitor solution algorithms and detect any potential errors in their logic, as the game does not enforce using this option and players can run their solution even without debugging. This results in not receiving some useful feedback or hints related to different concepts or skills, among them loop logic.

The pre-test and post-test results are displayed in Table 4, were the percentage of correct answers and the corresponding number of learners are provided. Regarding the pre-test, no learners answered correctly all pre-test questions or 6 questions out of 7 questions. 17% of learners provided correct answers to 5 questions out of 7 and 4 questions out of 7 in the pre-test. 50% of learners provided correct answers to 3 questions out of 7. 16% of learners provided correct answers to either all questions or answered correctly only 1 or 2 questions out of 7.

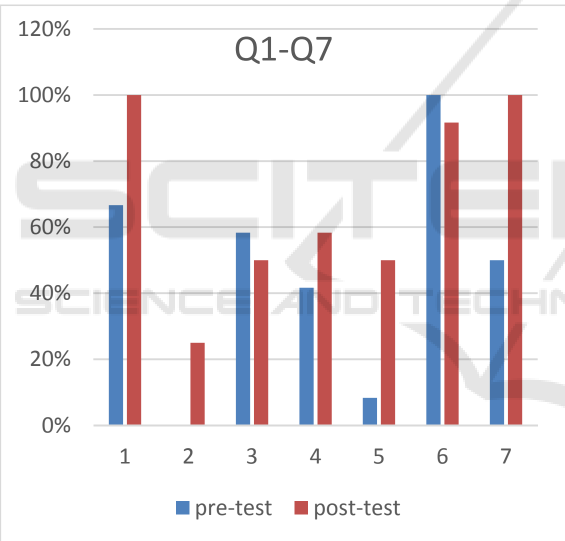


Figure 3: Average of pre- and post- test scores.

Table 4: Number of questions correctly answered by learners.

	Pre-test	Post-test
7 out of 7	0%	17%
6 out of 7	0%	0%
5 out of 7	17%	33%
4 out of 7	17%	42%
3 out of 7	50%	8%
2 out of 7	8%	0%
1 out of 7	8%	0%
none	0%	0%

Regarding the post-test, 17% of learners answered correctly all post-test questions and no learners provided correct answers to 6 questions out of 7 in the post-test. 33% of learners provided correct answers to 5 questions out of 7. 42% of learners provided correct answers to 4 questions out of 7. 8% of students have answered at most 3 questions out of 7.

An analysis of the results shows that AutoThinking game increases the learning outcomes for the learners. 92% of learners have answered correct at least 4 questions out of 7 in the post-test versus only 34% of learners in the pre-test.

In general, students' answers revealed the positive effect of the CT game and the fact that how an adaptive educational game could successfully engage learners in an interactive learning environment for promoting their CT skills. Findings of this preliminary study also unveiled that without highly complex learning environments, it is still possible to encourage students to produce some appropriate computational problem-solving practices, thereby fostering their CT concepts and skills. One possible reason for this encouraging findings is the adaptivity feature improvised in the game which enables the game to treat each learner according to his/her skill level. Such claim is in line with previous findings reported by other researchers. For instance, both Kickmeier-Rust et al. (2011) and Hooshyar, Yousefi, and Lim. (2018c) concluded that a meaningful personalization and adaptivity (individual support) are among crucial factors leading to the success of educational games which eventually result in improving learning performance.

5 CONCLUSION

The paper presented a case study that investigated the learning impact of an adaptive educational game called AutoThinking on adults. The educational game is about promoting CT skills and concepts where players should, in a role of a mouse, collect cheese and scape from two cats in the maze in order to complete or win the level. The game offers adaptivity in terms of game-play and learning. Pre- and Post- tests results analysis has shown that the game helped the adults to acquire knowledge on the CT especially for the sequence, the pattern recognition, the conditional, the function, and the simulation concepts.

As a future work, we plan to design and carry out a number of experimental studies with larger sample size so as to more accurately measure the effect of

AutoThinking game on learning gain of players. The experimental studies will include interviews that can be in the focus group mode. What's more, we aim to investigate the effect of adaptivity in the game by running a study between two different versions of the game, adaptive versus non-adaptive in different European countries.

REFERENCES

El Mawas, N., Gilliot, J.-M., Garlatti, S., Serrano Alvarado, P., Skaf-Molli, H., Eneau, J., Lameul, G., Marchandise, J. F. & Pentecouteau, H. (2017): Towards a Self-Regulated Learning in a Lifelong Learning Perspective. *Proceedings of the 9th International Conference on Computer Supported Education* , CSEDU (1) pp. 661-670, 2017.

El Mawas, N., Bradford, M., Andrews, J., Pathak, P. & Hava Muntean, C. (2018). A Case Study on 21st Century Skills Development Through a Computer Based Maths Game. *In Proceedings of EdMedia: World Conference on Educational Media and Technology* (pp. 1160-1169). Amsterdam, Netherlands.

Organisation for Economic Co-operation and Development [OECD]. *OECD skills outlook 2013. First results from the survey of adult skills*. Retrieved February 19, 2020 at http://www.oecd.org/ site/piaac/publications.htm

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

Wang, P. S. (2016). *From computing to computational thinking*. Chapman and Hall/CRC..

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.

Haddad, R. J., & Kalaani, Y. (2015). Can computational thinking predict academic performance?. In *2015 IEEE Integrated STEM Education Conference* (pp. 225-229). IEEE.

Brown, N. C., Sentance, S., Crick, T. & Humphreys, S. (2014). Restart: The resurgence of computer science in UK schools. *ACM Trans. Comput. Educ. TOCE*, vol. 14, no. 2, p. 9, 2014.

Repenning, A., Webb, D. C., Koh, K. H., Nickerson, H., Miller, S. B., Brand, C., ... & Gutierrez, K (2015). Scalable game design: A strategy to bring systemic computer science education to schools through game design and simulation creation. *ACM Trans. Comput. Educ. TOCE*, vol. 15, no. 2, p. 11, 2015.

European School Network Homepage: https://www.esnetwork.eu/. Accessed 9 Jan 2020.

Yardi, S., & Bruckman, A. (2007). What is computing? Bridging the gap between teenagers' perceptions and graduate students' experiences. In *Proceedings of the third international workshop on Computing education research* (pp. 39-50).

El Mawas, N., Trúchly, P., Podhradský, P., Hava Muntean, C. (2019). The Effect of Educational Game on Children Learning Experience in a Slovakian School. *In Proceedings of the 12th International Conference on Computer Supported Education*. CSEDU (1) 2019: 465-472

Hooshyar, D., Yousefi, M., Wang, M., & Lim, H. (2018a). A data□driven procedural□content□generation approach for educational games. *Journal of Computer Assisted Learning*, 34(6), 731-739.

Kazimoglu, C., Kiernan, M., Bacon, L., & Mackinnon, L. (2012). A serious game for developing computational thinking and learning introductory computer programming. *Procedia-Social and Behavioral Sciences*, 47, 1991-1999.

Kickmeier-Rust, M. D., Mattheiss, E., Steiner, C., & Albert, D. (2011). A psycho-pedagogical framework for multi-adaptive educational games. *International Journal of Game-Based Learning (IJGBL)*, 1(1), 45-58.

Hooshyar, D., Yousefi, M., & Lim, H. (2018b). Data-driven approaches to game player modeling: a systematic literature review. ACM Computing Surveys (CSUR), 50(6), 1-19.

Hooshyar, D., Lim, H., Pedaste, M., Yang, K., Fathi, M., & Yang, Y. (2019). AutoThinking: An Adaptive Computational Thinking Game. In *International Conference on Innovative Technologies and Learning* (pp. 381-391). Springer, Cham.

Grover, S., & Pea, R. (2013). Computational thinking in K–12: A review of the state of the field. *Educational researcher*, 42(1), 38-43.

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.

Harvey, B., & Mönig, J. (2010). Bringing "no ceiling" to Scratch: Can one language serve kids and computer scientists. *Proc. Constructionism*, 1-10.

Fraser, N. (2013). Blockly: A visual programming editor. URL: https://code. google. com/p/blockly.

Brennan, K., & Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. In *Proceedings of the 2012 annual meeting of the American educational research association*, Vancouver, Canada (Vol. 1, p. 25).

Meerbaum-Salant, O., Armoni, M., & Ben-Ari, M. (2011, June). Habits of programming in scratch. In *Proceedings of the 16th annual joint conference on Innovation and technology in computer science education* (pp. 168-172).

Weintrop, D., & Wilensky, U. (2012). RoboBuilder: A program-to-play constructionist video game. In *Proceedings of the constructionism 2012 conference*. Athens, Greece.

Land, S. M. (2000). Cognitive requirements for learning with open-ended learning environments. *Educational Technology Research & Development*, 48(3), 61–78.

Eagle, M., & Barnes, T. (2009). Experimental evaluation of an educational game for improved learning in introductory computing. *ACM SIGCSE Bulletin*, 41(1), 321-325.

Esper, S., Foster, S. R., Griswold, W. G., Herrera, C., & Snyder, W. (2014). CodeSpells: bridging educational language features with industry-standard languages. In *Proceedings of the 14th Koli calling international conference on computing education research* (pp. 05-14).
Ayman, R., Sharaf, N., Ahmed, G., & Abdennadher, S. (2018). MiniColon; teaching kids computational thinking using an interactive serious game. In *Joint International Conference on Serious Games* (pp. 79-90). Springer, Cham.
Zhao, W., & Shute, V. J. (2019). Can playing a video game foster computational thinking skills?. *Computers & Education*, 141, 103633.
Hooshyar, D., Yousefi, M., & Lim, H. (2018c). A procedural content generation-based framework for educational games: Toward a tailored data-driven game for developing early English reading skills. Journal of Educational Computing Research, 56(2), 293-310.