International Journal of Digital Society (IJDS), Volume 10, Issue 2, June 2019

Computer Programming: A Case Study of Teaching Loop Statement by Using an Interactive Educational Game

Eleni Makri, Neeraj Choudhary, Cristina Hava Muntean
National College of Ireland, School of Computing, Ireland

Abstract

any European countries reported a decline in the number of students who choose STE sub ects as their future career. This is due many to the fact that STE sub ects are perceived as being difficult and boring. This paper presents findings of a case study that made use of an educational game called Loop to teach the loop statement in a Software Development module. The game was deployed as part of the NE T N pro ect large scale Programming pilot run at National College of Ireland NCI as part of igher Diploma in Computing a level conversion degree during the academic year 2 1 2 1 . 34 mature students over 23 years old that already hold a third level educational degree took part in the study. Pre-and post-game assessments and post-game uestionnaire were employed while the Loop game was used as part of the lab sessions. Research findings indicate that ma ority of the students achieved improved knowledge performance in ava programming language coding of the loop statement. The positive assessment outcomes are discussed together with students feedback in terms of game design and game usability provided in the uestionnaire.

1. Introduction

The latest development of technologies and computers have triggered the adoption of various technology based pedagogies by various educational institutions. One of the application of technology aided education is educational computer game that was introduced in both formal and informal educational curricula for teaching computer programming concepts [1], [2]. Educational games are regarded as not only developing computational thinking [3] and comprehension [4] in computer programming but also as a means to advance students' immersion in Science, Technology, Engineering and Mathematics (STEM) subjects. In that sense, educational games assist and motivate both teachers and students in experiential learning of computer programming (i.e. learning through refection on doing, hand-on learning,) [5], as they deal with abstract and complex concepts, difficult to teach in an enjoyable classroom exercises. However, educational games do demand complicated procedures and effort to design, implement and test them. Further, few studies have been reported to evaluate the game's cognitive and attitudinal outcomes as well as their effects on learners' (i.e. students') performance and experience [6].

In this context, this paper aims to present the results of a case study run at the National College of Ireland (NCI) that explores the students' knowledge achievement and their usability feedback when the *Loop* educational game was used to teach the *loop* statement principles as part of the Software Development module. The case study was run with two different cohorts of students during the academic year 2018/2019. The Loop game was developed as part of EU Horizon 2020 NEWTON project[1]. NEWTON project designs, develops, deploys and evaluates diverse innovative technology based educational solutions that make use of adaptive multimedia, multiple sensorial media delivery, Virtual and Augmented Reality (VR/AR) learning, Virtual Teaching and Learning Labs (Virtual Labs), Fabrication Labs (Fab Labs), personalization and gamification in the teaching and learning process [7-12]. These solutions are applied in conjunction with different pedagogical approaches including self-directed, game-based, flipped classroom, and problem-based learning methods to different learner audiences, including students with special educational needs [13-17]. The NEWTON project has also created and deployed an innovative learning management platform, NEWTELP[2], which embeds the NEWTON educational content and applications.

The remainder of this paper is organised as follows. **Section 2** presents a comprehensive review of various educational games based approaches applied for teaching and learning computer programming in *elementary*, *secondary* and *higher education*. **Section 3** provides a description of the

[1]NEWTON project website, [Online] Available: http://newtonproject.eu
[2]NEWTELP platform, [Online] Available: http://newtelp.eu

research methodology applied in this case study. the including the description of the *Loop* educational game. **Section 4** provides an analysis of the knowledge test and of the game usability survey. **Section 5** synopsizes the findings, their implications and concludes this paper with new avenues for further research.

2. Literature review

The integration of educational computer games into different educational curricula has shown to advance students' learning, knowledge and skills' enrichment and commitment [18], [19]. Successful educational computer games are claimed to expose players (i.e. students) to critical thinking and problem-solving skills whilst offering them immediate real-time feedback and rewards to motivate them to discover more solutions and good practice [20]. In addition, when a student interacts either individually or collectively with his or her peers in the game, he/she is required to provide input and generate its' own learning. That input combined with gameplay feedback per se, seems to assist the student in understanding the game-associated concepts addressed each time [21].

Teaching computer programming across different educational levels and curricula (i.e. *elementary, secondary* and *third level education*), tends to be a really challenging issue: diverse computer knowledge and abilities exist, various kinds of applications are used, lengthy amount of time is required to teach different programming concepts and difficulties are faced in keeping young students engaged in their gameplay learning experience [22] [23]. As regards the learning effectiveness of educational computer games employed in teaching programming concepts at *elementary* and *secondary* levels, it tends to be rather equivocal findings as far as programming knowledge and competencies, motivation and learner's experience aspects are concerned. In synopsizing some indicative examples of them, are discussed next.

Gomes, Pontual Falcão and Tedesco [24] present a preliminary study of the assessment of several games that targeted teaching and learning of programming concepts at a private *elementary* school. 42 young children aged 5-7 years old played educational games in 4 teams as a learning activity in their computer science classrooms, over a 9-week period in 40 minute sessions. The data collected as part of semi-structured interviews, observations and children feedback after their gameplay experience, indicated that they met a range of difficulties in playing the games, handling the programming language and understanding the programming concept of repetition, in particular.

Giannakoulas and Xinogalos [25] elaborate on their study of evaluating a role-playing game aimed at teaching fundamental programming concepts to last grade *elementary school* children during their information and communication technologies (ICT) course. The children had prior knowledge of basic programming concepts. During their gameplay experience, they had to help the main character of the game to meet his friends in a wild natural environment. The execution of the game included several levels and sections complemented by game-related tasks and questions. Quantitative and qualitative data collected by both teachers and students during the case study, revealed that the game was perceived to be effective in teaching introductory programming concepts. Most of the children (over 60%) completed successfully almost all of the designed game-associated exercises and answered correctly the addressed questions. Almost half of the children reported they needed assistance during game play while 63.16% were comfortable in following the sequence of instructions they had to complete. Half the children reported that it was a nice idea to use the game in teaching and learning programming concepts, while 21.05% of them were not confident about that. More than half of the students, indicated favorable views for introducing the computer game in the curriculum of the ICT course. The vast majority of children believed that the game supported them in understanding the programming concepts.

Shim, Kwon and Lee [26] present a study involving 48 *primary school* students who appraised 2 games as a learning approach in teaching programming module. Pre-test measures assessed the attitudes of students regarding computer programming and their comprehension of programming concepts. Post-test measures of usability (enjoyment and memory skill), pleasure in the gaming activity and attitudes towards and understanding of programming concepts, were employed after the students spent overall 8 hours in lab sessions playing the games. Overall, the students scored high in usability and enjoyment with the programming activity features, but indicated the lowest score in the question whether it was straightforward to resolve the problems or difficulties encountered in their gameplay experience, reflecting the complex nature of programming concepts. In total, the students perceived the gaming activity as positive in supporting them to view programming learning as helpful, while their understanding of the programming concepts increased significantly after they played the games.

Papastergiou [27] describes the design and evaluation of an educational game aimed at teaching computer memory concepts against another non-gaming application by 88 *high school* students from two different Greek educational institutions attending the computer science module. The game directed the students as "protagonists" through 3 rooms within each one the players had to resolve a number of problems by answering storyline-related questions using information spots and appropriate feedback, overcoming "hurdles" and achieve the final mission of collecting an "end" banner in each room. When starting the game, the students were provided with a number of lives and zero marks and each time they gave a response to an answer in the room they were located, they received descriptive feedback regarding their answers. If their response wrong, they experience reduced number of lives and were offered help to give a correct answer. While situated in the room, the players had further the opportunity to extend their number of lives and get extra marks. If their lives were finally lost, then the game was finished and the students could either terminate playing the game or begin to play again from the number of room (1, 2 or 3) they had passed before completing the game. The students experienced both the game and the non-gaming application in their school's lab classroom during their computer science module for 2 hours, supported by the researcher and their school teachers. Overall, the students who played the game indicated they were more attracted to their gameplay experience, reported increased engagement and higher post-test scores in computer memory acquisition, in relation to their peers who used the non-gaming application.

Theodoropoulos, Antoniou and Lepouras [28] elaborate on the evaluation of a game for teaching basic computational thinking and programming concepts to *high school* students. The game consists of puzzles in each of them the students needed to "write" a sequence of instructions that dragged a character (actor) through a maze or built a number of trigonometrical forms. 77 *high school* students with basic and not advanced programming knowledge, played the game during their computer labs within a 4-week time period for 7 hours in total, outside of their regular school classes. The user feedback questionnaires returned positive views, as follows: 81.8% of the students indicated their gameplay experience as enjoyable and not difficult to use the game. 65% reported they had gained a certain knowledge of programming concepts by using the game, it was a gaming activity they would like to do again in a future experience and 62.4% of them were highly motivated to be taught by games. Introverted cognitive style students were more effective in playing the game, their peers with high intuition (i.e. information they like to attend to) perceived the gaming activity as easier and those classified as judging ones (i.e. in treating the wider world), achieved better gaming performance and indicated they were able to learn introductory programming knowledge and skills by gaming.

On the other hand, *higher education* students enrolled in computer science degree, seem to fall short of prior experience in employing programming concepts and practicing programming knowledge and skills leading to their lowered engagement in learning programming [29]. In helping them to learn and exercise the corresponding concepts more effectively, computer programming games were designed, developed and disseminated predominantly over the last years. However, as with other educational curricula, a limited number of studies are reported to evaluate whether these games seem to support or advance programming learning in effect [30]. As regards the learning effectiveness of programming games used in teaching programming modules at *university level*, there seem to be rather inconclusive findings as far as programming knowledge and skills, motivation and user experience features are concerned. In summarizing a few indicative examples of them, are discussed next:

Dolgopolovas, Jevsikova and Dagiene [31] indicated in their study evaluating the learning experience of 24 first year engineering students attending an introductory course in C programming language that more than half of the students were successful in providing at least one correct answer out of 5 questions introduced for appropriate matching between game application and programming language code concepts, after their gameplay exercise.

Muratet, Torguet, Viallet, and Jessel [32] in their study involving 15 first year students of computer science playing a programming game, indicated that applying programming in the game did advance students' enjoyment, with more than half of them indicating their motivation to continue playing the game. Overall, in terms of the game's usability features explored, the students did demonstrate moderate levels of satisfaction.

Adamo-Villani, Oania, and Cooper [33] in their study concerning 17 undergraduate students registered in computer science, computer graphics technology and electrical engineering courses playing a role-playing game for teaching secure coding and information assurance concepts, indicated mixed findings, as follows: 16 out of 17 students were not able to finish level 1 of the game (error rate average: 26.9%), in contrast to level 2 where all participants completed the corresponding level (error rate average: 36.10%). 15 out of 17 students reported they encountered level 1 of the game as difficult to complete, although it was motivating and posed a challenge to them. Level 2, on the other hand, was perceived by them as easy to finish. Game and time controls were indicated as not satisfactory in level 1, game screen appeared with low visibility, the most enjoyable aspect of level 1 was graphics and sound effects. Observation, video recordings and screen captures' data revealed that students became initially confused about the gameplay levels, but as they were moving to the next level, they appeared to be motivated and enjoyed their overall gameplay experience, irrespective of their gender, time spent in finishing the levels and error rates obtained.

The majority of research papers exploring motivation and learning performance of educational games for computer programming as well as their effects on students' cognitive (i.e. knowledge achievement), attitudinal and behavioral (i.e. user experience) seem to yield contradictory findings. In that respect, the use of games in teaching programming concepts across diverse curricular modules in higher education involving students from different cultural and educational background, needs to be further explored.

Following the rationale above, the research work reported in this paper, aims to explore the impact of the *Loop* computer game that teaches *loop* statement principles on the knowledge acquisition as part of a case study involving computing students. Game usability is also investigated through the use of a questionnaire.

3. Case Study Description

The *Loop* game was developed as part of the NEWTON project and aims to help students understand *loop* principles and operations in Java programming language. The game was used as part of lab sessions in Software Development module delivered at National College of Ireland. The case study was run with two different cohorts of students enrolled into a one-year computing conversion course Higher Diploma in Science in Computing during the academic year 2018/2019.

The *Loop* game is part of the NEWTON Programming Large-Scale pilot [17] which investigates the impact of various NEWTON technologies. This pilot was deployed in 3 European Universities during a full semester programming module and game based learning, flipped classroom, and combined FC and problem based learning innovative pedagogies were applied. Programming code examples were designed and developed for both C and Java programming languages.

This paper explores only the game-based learning aspect of the course where programming statements and code examples were provided through the Java programming language.

A total of 34 mature students (over 23 years old) that already hold a third level educational degree took part in the study. Half of the students had computer science/engineering as prior education background while the other half of the students had humanity and education. The gender distribution of the participants is 34.29% females and 67.71% males.

3.1 *Loop* Game

Lecturers involved in the delivery of the computer programming modules were consulted before the design of the game to identify the key knowledge points and concepts that students struggle with. The lecturers were also involved in the review and feedback process after the game development in order to evaluate the educational content provided by the game. The game is a 2D educational computer game developed in Unity.

The *Loop* game introduces the programming concept of *loop* statement through the completion of game activities such as tasks and quizzes. The activities correspond to coins' collection carried out by the player who acts as a mermaid within an underwater storyline context (i.e. role-playing game).

The game has 3 levels that cover the following concepts:

a) basic *for* loop (Level 1), where the coins need to be collected and stored (first level). A group of statements located within the body of the *for* loop are executed multiple times while the boolean condition of the loop statement is true. The player controls the movement of the

Figure 1. Loop Game – Level 1

Figure 2. Loop Game – Level 3

mermaid that has to execute the one at a time coin collection task for 5 times. Five corresponds to repeating the body of the loop 5 times as specified by the boolean condition of the loop (Figure 1).

b) *for* loop with *continue* statement (Level 2). The player has the same task (i.e. coins' collection) but some of the coins disappear once they are touched by the mermaid. The coin disappearance indicates the mermaid skips the remaining steps of the loop (i.e., moving towards the treasure chest and storing the coin). and continues with the next iteration.

c) *for* loop with *break* statement (Level 3), visualized by letting one of the coins becomes a jackpot coin (reward) once the mermaid collects them (third level). When the player comes across a jackpot coin, the mermaid "breaks" out from the loop of tasks and the level finishes immediately (Figure 2).

Once a level is completed, the student is asked to complete a quiz that requires them to answer a number of knowledge game-related questions.

3.2 Study Design

The case study adopted the pre- and post-assessment tests approach when the *Loop* game was used during the lab sessions part of the Software Development module. The *Loop* statement principles were exemplified in the game using the Java programming language.

The style of the questions used in the pre- and post-tests were multiple choice. The students were asked to answer pre-and-post knowledge tests corresponding to their pre-and-post play game experience with the *Loop* game. Each knowledge test consists of 3 multiple choice questions with their corresponding answer options. The pre-test and post-test were designed in such a way that each question from the two tests covers the same topic (i.e. question 1 in pre-test and post-test assess the knowledge level on the same topic) at the same difficulty level. Each correct answered question was marked 1 point.

Upon finishing all the three levels of the game, students were asked to answer a post-game questionnaire, which includes questions related to game design and game usability (e.g. support for knowledge acquisition and learning experience) The questions are all single choice questions with 5-level Likert scale answer options such as strongly disagree, disagree, neutral, agree, and strongly agree (Table 1).

Table 1. Post-game Questionnaire

Category	No	Questions
Game Design	Q1	I understood all the different parts of the game.
	Q2	The game task and levels were properly designed.
	Q3	The game user interface design is pleasant.
Game Usability	Q4	I would prefer to learn without educational games.
	Q5	The game helped me understand the programming concepts.
	Q6	This game is a good complement to textbooks and lecture slides on this topic.

4. Case Study Data Analysis

4.1 Knowledge Assessment

Pre-and-post knowledge test scores were compared in terms of the mean value of overall students' score and the percentage of the correct answers per question given by the students. The findings are presented in Table 2 and Figure 3.

The analysis indicates that the mean value of the overall scores for the 3 questions in the post-test was

Table 2. Mean value of the marks in the pre- and post-game assessments

Overall Knowledge *Pre-test result*		Overall Knowledge *Post-test result*	
Mean (out of 3)	% correct answers	Mean (out of 3)	% correct answers
1.29	43%	2.11	71%

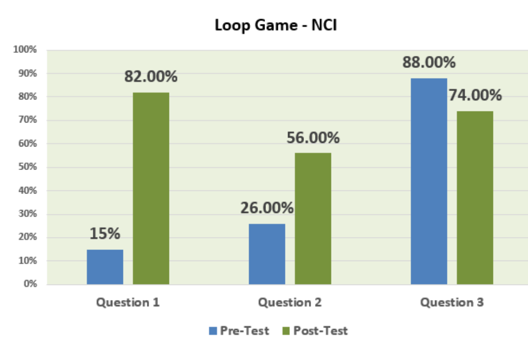

Figure 3. Percentage of correct answers per question in the pre- and post- game assessments

Table 3. Distribution of students' answers (numbers and percentage) provided for the survey's questions.

	Question	**Strongly Disagree**	**Disagree**	**Neutral**	**Agree**	**Strongly Agree**
Game Design	Q1	1 (3%)	2 (6%)	5 (16%)	18 (58%)	5 (16%)
	Q2	1 (3%)	3 (10%)	8 (26%)	15 (48%)	4 (13%)
	Q3	1 (3%)	1 (3%)	6 (20%)	17 (57%)	5 (17%)
Game Usability	Q4	3 (9%)	13 (41%)	11 (34%)	4 (13%)	1 (3%)
	Q5	1 (3%)	7 (22%)	5 (16%)	15 (47%)	4 (13%)
	Q6	1 (3%)	3 (9%)	7 (22%)	16 (50%)	5 (16%)

Table 4. Students answers distribution across positive, neutral and negative feedback.

Category	No	Questions	Positive	Neutral	Negative
Game Design	Q1	I understood all the different parts of the game.	74%	16%	10%
	Q2	The game task and levels were properly designed.	61%	26%	13%
	Q3	The game user interface design is pleasant.	73%	20%	7%
		Overall Feedback	*69.33%*	*20.66*	*10%*
Game Usability	Q4	I would prefer to learn without educational games.	16%	34%	50%
	Q5	The game helped me understand the programming concepts.	59%	16%	25%
	Q6	This game is a good complement to textbooks and lecture slides on this topic.	66%	22%	13%
		Overall Feedback (Q4 inverted)	*58.33%*	*24%*	*18%*

higher than the corresponding one in the pre-test (2.11 and 1.29, respectively). In addition, the percentage of the correct answers given by the students in the post- test for questions 1 and 2 was significant higher than the corresponding ones in the pre- test (82%, 56% and 15%, 26%, respectively). The percentage of the correct answers given by the students in all 3 questions overall in the post-test was higher than the corresponding one in the pre-test (71% and 43%, respectively). These results indicate that the students have achieved better results for the Java programming *loop* concept after employing the *Loop* game. Worth to be mentioned is that an important number of correct answers were provided in the pre-test due to the fact that 50% of the students had a computer/engineering background and therefore most probably they have studied in the past some programming concepts, not necessary through Java programming language.

4.2 Post-Game Questionnaire Analysis

The answers provided in the questionnare that was run after the use of the *Loop* game were analised interms of distribution of the answers across the 5 answer options in order to understand the students learning experience with the game. The findings are been presented in Table 3 and Table 4. The analysis was performed based on the two categories of questions: game design and game usability (e.g. knowledge acquisition and learning experience)

The answers provided by the students for each question were grouped into positive feedback (strongly agree and agree answers), neutral feedback (neutral answer) and negative feedback (disagree and strongly disagree). Overall, the results show that majority of the students provided positive feedback (69.33%) in terms of game design and only 10% did not appreciate the game interface and the design of the game levels.

In terms of game usability, worth to be mentioned is that Q4 is an inverted question where a negative answer rejects the statement of the question (e.g. “I would prefer to learn without educational games”). Converting question 4 and the provided answers into a positive statement, the overall feedback across Q4, Q5 and Q6 indicates that almost 60% of the students considered that the game helped them to learn the *loop* statement and only 18% did not consider that.

5. Discussion and Conclusion

The results presented in this paper indicate that the *Loop* game helped majority of students to learn the loop statement principle and coding related aspects in the Java programming language, thus achieving high scores in the post-test. Based on their limited prior knowledge on the *loop* statement concept in Java, the *Loop* game was found to be useful as a particular exercise and learning tool for teaching and learning programming coding objectives.

The slightly more favorable percentage of correct answers provided by the participants for question 3 from the pre-test than the corresponding one in the post-test, may indicate the need for the *Loop* game to be additionally improved in terms of captivating the programming coding procedure learning.

However, the high result in the overall post-test result (71%) indicates that the game was perceived as a supportive application for computing students with less advanced knowledge and skills in programming language coding concepts. This affirmation is also supported through the feedback received in the post-game questionnaire.

These findings tend to lend support to prior programming game-based learning ones obtained in higher education domain [31], [33].

As the game was applied in a module delivered as part of Level 8 Computing conversion course to mature students that already had a 3rd level qualification in different areas, it would be fruitful to explore additionally an appraisal of the *Loop* game in terms of programming knowledge acquisition across experienced and non-experienced users and (or) students' prior qualification. These seem to be interesting questions and issues for further research in **educational games** for computer programming curricular.

Acknowledgements

This work is supported by the European Union's Horizon 2020 Research and Innovation Programme via Grant Agreement no. 688503 for NEWTON Project (http://newtonproject.eu). The authors thank the NEWTON's Project participating partners, teachers and students that made this study possible.

References

[1] C.H. Muntean, N. El Mawas, M. Bradford, and P. Pathak, "Investigating the impact of an immersive computer-based math game on the learning process of undergraduate students", *Proceedings of the 48th Annual Frontiers in Education Conference (FIE)* (pp. 1-7), San Jose, CA: IEEE, 2018

[2] C. H. Muntean, J. Andrews and G.-M. Muntean, "Final Frontier: An Educational Game on Solar System Concepts Acquisition for Primary Schools," in *17th IEEE International Conference on Advanced Learning Technologies (ICALT)*, Timisoara, Romania, July 2017

[3] J. M. Wing, "Computational thinking", *Communications of the ACM, 49, 3,* 2006, pp. 33-35

[4] J. Wakefield, "BBC gives children mini-computers in Make it Digital scheme", *BBC News*, Retrieved 3rd May 2017 from http://www.bbc.co.uk/news/technology-31834927

[5] K. Kiili, "Digital game-based learning: Towards an experiential gaming model", *The Internet and Higher Education*, *8*, 2005, pp. 13-24

[6] J. Cheon, S. Chung, and S. Lee, "The roles of attitudinal perceptions and cognitive achievements in a serious game", *Journal of Educational Computing*, *52,1*, 2015, pp. 3-25

[7] I. Ghergulescu, T.Lynch, M. Bratu, A.N. Moldovan, C.H.Muntean andG.M. Muntean, "STEM Education with Atomic Structure Virtual Lab for Learners with Special Educational Needs", *10th International Conference on Education and New Learning Technologies (EDULEARN)*, Palma de Mallorca, Spain, pp. 8747-8752, 2018

[8] N. El Mawas, I. Ghergulescu,A.-N. Moldovan and C. H. Muntean, "Pedagogical based Learner Model Characteristics", *Ireland International Conference on Education (IICE)*, Dublin, Ireland, April, 2018

[9] N. El Mawas, I. Tal,A. N. Moldovan,D. Bogusevschi,J. Andrews, G.-M.Muntean and C.H. Muntean,"Final Frontier Game: A Case Study on Learner Experience", *Proceedings of the 10th International Conference on Computer Supported Education (CSEDU)*, Madeira, Portugal, pp. 122-129, 2018

[10] A.-N. Moldovan, I. Ghergulescu, C. H. Muntean, "VQAMap: A novel mechanism for mapping objective video quality metrics to subjective MOS scale", *IEEE Transactions on Broadcasting*, 62, 3, pp. 610-627, 2016

[11] D. Bogusevschi, I. Tal, M. Bratu, B. Gornea, D. Caraman,I. Ghergulescu, C. H. Muntean, C. and G.-M. Muntean, "Water Cycle in Nature: Small-Scale STEM Education Pilot". *Proceedings of EdMedia: World Conference on Educational Media and Technology* (pp. 1496-1505), Amsterdam, Netherlands: Association for the Advancement of Computing in Education (AACE), 2018

[12] C. H. Muntean and J. Andrews, "Final Frontier: An Educational Game on Solar System Concepts Acquisition for Primary Schools", *17th IEEE International Conference on Advanced Learning Technologies (ICALT)*, Timisoara, Romania, pp. 335-337, 2017

[13] N. El Mawas,M. Bradford, J. Andrews, P. Pathak and C.H. Muntean, "A Case Study on 21st Century Skills Development Through a Computer Based Maths Game", *Proceedings of EdMedia: World Conference on Educational Media and Technology* (pp. 1160-1169), Amsterdam, Netherlands: Association for the Advancement of Computing in Education (AACE), 2018

[14] M. Bradford, C. H. Muntean and P. Pathak, "An Analysis of flip classroom pedagogy in first year undergraduate mathematics for computing", *IEEE Frontiers in Education Conference (FIE)*, Madrid, Spain, pp.1-5, 2014

[15] C.H. Muntean, N. El Mawas, M. Bradford, and P. Pathak, "Investigating the impact of an immersive computer-based math game on the learning process of undergraduate students", *Proceedings of the 48th Annual Frontiers in Education Conference (FIE)* (pp. 1-7), San Jose, CA: IEEE, 2018

[16] T. Bi, F. Silva, G. Ghinea, and G. M. Muntean, "Improving Learning Experience by Employing DASH-based Mulsemedia Delivery", *IEEE Games, Entertainment, Media Conference (GEM)*, Galway, Ireland, August, 2018

[17] D. Zhao, A. E. Chis, G.M. Muntean, and C.H. Muntean, "A Large-Scale Pilot Study on Game-based Learning and Blended Learning Methodologies in Undergraduate Programming Courses", *10th International Conference on Education and New Learning Technologies (EDULEARN)*, Palma de Mallorca, Spain, July, 2018

[18] T.M. Connolly, E.A. Boyle, E. MacArthur, T. Hainey and J.M. Boyle, ″A systematic literature review of empirical evidence on computer games and serious games″, *Computers & Education, 59*, 2012, pp. 661-686

[19] I. Ghergulescu and C. H. Muntean, "Assessment of motivation in games based e-learning", in *IADIS International Conference Cognition and Exploratory Learning in Digital Age (CELDA 2010)*, Timisoara, Romania, pp.71-78, 2010

[20] M. Olsson, and P. Mozelius, ″Learning to Program by Playing Learning Games″, *Proceedings of the European Conference on Games Based Learning* (ed.) Maja Pivec and Josef Gründler, Reading, UK, Academic Conferences and Publishing International Limited, 11, 498-506, 2017

[21] W.Y. Seng, and M.H.M. Yatim, ″Computer game as learning and teaching tool for object oriented programming in higher education institution″, *Procedia-Social and Behavioural Sciences, 123*, 2014, pp. 215-224

[22] S. Combéfis, G. Beresnevičius, and V. Dagiene, "Learning programming through games and contests: Overview, characterization and discussion", *Olympiads in Informatics*, 10, 2016, pp. 39-60

[23] N.Iten and D. Petko, "Learning with serious games: Is fun playing the game a predictor of learning success?", *British Journal of Educational Technology, 47, 1,* 2016, pp. 151-163

[24] T.C.S. Gomes, T. Pontual Falcão, and P.C.A.R. Tedesco, "Exploring an approach based on digital games for teaching programming concepts to young children", *International Journal of Child-Computer Interaction, 16,* 2018, pp. 77-84

[25] A. Giannakoulas and S. Xinogalos, "A pilot study on the effectiveness and acceptance of an educational game for teaching programming concepts to primary school students", *Education and Information Technologies, 23, 5,* 2018, pp. 2029-2052

[26] J. Shim, D. Kwon, and W. Lee, "The effects of robot game environment on computer programming education for elementary school students", *IEEE Transactions on Education*, 60, 2, 2017, pp. 164-172

[27] M. Papastergiou, "Digital game-based learning in high school computer science education: Impact on educational effectiveness and student motivation", *Computers & Education, 52,* 2009, pp. 1-12

[28] A. Theodoropoulos, A. Antoniou, and G. Lepouras, "How do different cognitive styles affect learning programming? Insights from a game-based approach in Greek schools", *ACM Transactions on Computing Education, 17, 1,* 2016, pp. 320-325

[29] M. Muratet, P. Torguet, J.-P. Jessel, and F. Viallet, "Towards a serious game to help students learn computer programming", *International Journal of Computer Games Technology*, 2009, pp. 470-590

[30] L. Sung, C. Hillyard, R.L. Angotti, M.W. Panitz, D.S. Goldstein, and J. Nordlinger, ″Game-Themed Programming Assignment Modules: A Pathway for Gradual Integration of Gaming Context into Existing Introductory Programming Courses″, *IEEE Education Society, 54, 3*, 2010, pp. 416-427

[31] V. Dolgopolovas, T. Jevsikova, and V.Dagiene, "From Android games to coding in C-An approach to motivate novice engineering students to learn programming: A case study", *Comput Appl Eng Educ.*, *26*, 2017, pp. 75-90

[32] M. Muratet, P. Torguet, F. Viallet, and J.-P. Jessel, "Experimental feedback on ProgPlay, a serious game for programming practice", *EUROGRAPHICS (European Association for Computer Graphics)*, Norrköping, Sweden, 2010

[33] N. Adamo-Villani, M. Oania, and S. Cooper, "Using a serious game approach to teach secure coding in introductory programming: Development and initial findings", *Journal of Educational Technology Systems, 41, 2,* 2012, pp. 107-131

[34] E. Makri, D. Koryzis, D. Spiliotopoulos, and V. Svolopoulos, "Metalogue's virtual agent for negotiation: Its' effects on learning experience, metacogntive and individual-and-community-level attitudes pre-and-post interaction", *11th International Conference on Education and New Learning Technologies (EDULEARN)*, Palma de Mallorca, Spain, 1st -3rd July, 2019

[35] I. Ghergulescu and C.H. Muntean, "Motivation Monitoring and Assessment Extension for Input-Process-Outcome Game Model", *International Journal of Game-Based Learning*, 4, 2014, pp. 15-35

[36] I. Ghergulescu and C. H. Muntean, "MoGAME: Motivation based Game Level Adaptation Mechanism", in *The 10th Annual Irish Learning Technology Association Conference (EdTech 2010)*, Athlone, Ireland, 2010