Game-based Learning: Enhancing Student Experience, Knowledge Gain and Usability in Higher Education Programming Courses

Dan Zhao, Cristina Hava Muntean, Adriana E. Chis, Gregor Rozinaj,
and Gabriel-Miro Muntean, *Senior Member, IEEE*

Abstract—Contributions: **This paper presents a large-scale study which investigates students' reaction to game-based learning as part of Programming courses. The study focuses on knowledge acquisition, learner experience and game usability.**
Background: **Despite the rapid growth of the Information and Communication Technologies (ICT) sector, the lack of engagement with Science, Technology, Engineering, and Mathematics (STEM) subjects and high dropout rates in computer science and engineering majors is linked directly to the large number of unfilled vacancies in the ICT employment market. To tackle one of the underlying causes for this crisis, (i.e., traditional teaching paradigms struggle to attract students to rather abstract and difficult STEM subjects such as programming), innovative technology-enhanced learning solutions are sought.**
Intended Outcomes: **A set of serious games were proposed and designed to promote students' understanding of programming concepts, improve their confidence, stimulate their interest in STEM and increase engagement with the courses through vivid and appealing scenarios.**
Application Design: **Targeting undergraduate and postgraduate students, the games focused on several key programming topics. They were designed to visualize the programming concepts in illustrative and entertaining scenarios. A comprehensive assessment methodology which includes surveys, observations, and interviews was employed to investigate the impact of the games.**
Findings: **The results show that by using the games in the teaching and learning process all the students have benefited, although differently based on their location, educational backgrounds and game played. The impact of detailed demographic aspects such as participants' use of technology, their initial attitude towards school and learning STEM on the results needs further study.**

*Index Terms—***game-based learning, programming, education**

I. Introduction

STEM-oriented third level education courses enable students to develop important skills that are currently required on the market such as learning to think, employing creativity and making use of critical thinking. The STEM area spans across many disciplines and includes diverse occupations, from software developers, engineers and data scientists to biotechnologists, physicists and chemists. However, according to

Dan Zhao is with Department of Mathematics and Theory, Peng Cheng Laboratory, Shenzhen 518066, China (e-mail: zhaod01@pcl.ac.cn)

Cristina Muntean and Adriana E. Chis are with School of Computing, National College of Ireland, Dublin 1, D01K6W2 Ireland (e-mail addresses: cristina.muntean@ncirl.ie and adriana.chis@ncirl.ie).

Gregor Rozinaj is with Slovak Technical University of Bratislava, Slovakia, (e-mail: gregor.rozinaj@stuba.sk)

Gabriel-Miro Muntean is with Performance Engineering Laboratory, School of Electronic Engineering, Dublin City University, Dublin 9, Ireland (e-mail: gabriel.muntean@dcu.ie).

LinkedIn data, the top ten most demanded skills were all computer-related skills. It is also predicted that by 2024, 73% of STEM job growth will be in ICT occupations, whereas only 3% will be in physical sciences and 3% in life sciences [1]. Recently, ICT has been the fastest growing area among all job categories in the European job market [2]. However, there is still an increasingly large number of unfilled vacancies in the ICT job market, expected to exceed 750 thousands by 2020 [3].

Student enrollments in ICT related courses have increased rapidly over the past 5 years, but the openings still exceed the number of applicants. Despite the high increase in the number of students enrolling into computer science and engineering courses, many students struggle during the first year of their courses and high failing and dropout rates are noticed. For instance, the Higher Education Statistics Agency (HESA) reports that computing courses have the highest first year dropout rate in UK. For example, in 2017 over 10% of students that left higher education before their second year were computer science students [4]. Additionally, almost 40% of the students enrolled in a computer science degree programme drop out from their studies [5]. This percentage includes both students who drop out of university voluntarily and because they have not achieved the required grades to continue.

All computer science, engineering and information technology courses include computer programming-related modules and they are among the most important subjects. Students start learning programming in their early years of studies. As part of a programming module, students are required to demonstrate competencies in the principles of programming (even though some of these concepts are highly abstract and complex), knowledge of programming languages, problem-solving skills, and effective program design and implementation. Research has shown that programming is among the most challenging STEM subjects in the curriculum, and students find it difficult to grasp [6]. In this context, it is important to make use of innovative teaching strategies that provide students with the most efficient learning environment.

Various recommendations have been made to improve teaching and learning of programming. For example, hands-on skills in programming can be developed by combining laboratory practical sessions, projects, seminars and tutorials with lectures [6]. A good approach to teaching and learning programming is to motivate the students by using edutainment-based pedagogy that involves problem solving practical approaches, authentic context showing how the acquired knowl-

edge will be used in real life, conceptual learning and authentic activities [7]. Edutainment is very relevant for education today as it aims to provide education with engagement.

As the current generation of students were exposed to high-end technology at a very early age, they find it difficult to attend and focus on a teaching session delivered in a traditional way where the teacher just talks explaining programming concepts and demonstrates pieces of programming code. Therefore, new teaching approaches that make use of technology should be used when teaching programming concepts. Technology-oriented pedagogies such as Flipped Classroom (FC) and Game-based Learning (GBL) support independent learning, actively engage students in their learning process, and develop problem solving skills.

As an overwhelming number of teenagers play video games almost every day, first year students are familiarized with computer games and therefore using educational games as part of their learning environment becomes something natural to do for them. Educational games engage students, encourage them to get involved in real-time activities, support learning by experimenting and last, but not least important, boost learners' attraction to programming.

This research paper investigates the use of interactive video game-based teaching in computer programming modules, as part of real programming courses. A research study was conducted in the context of the EU-funded NEWTON project[1], which develops innovative technologies to enhance learning process. The paper describes the NEWTON project study performed in three European third level education institutions, and investigates students' perception towards the use of educational games when learning programming-related concepts. Usability, knowledge acquisition and learner experience were analysed through questionnaires and interviews. Furthermore, the influence of students' study phases on their perception towards the games were also studied.

The paper is organized as follows. Section II discusses related works. Section III introduces the background of the study in the context of the NEWTON project and describes the innovative educational games developed and applied in the classroom. Section IV presents an overview of the study and its research methodology. Sections V and VI present overall results and cross analysis, respectively, followed by a section discussing implications and lessons learned. The last section summarizes the paper, draws conclusions regarding the research study performed, and presents future perspectives.

II. RELATED WORK

Educational games have started to become a popular teaching and learning aid in STEM subjects [8], [9], especially in programming-related courses.

In order to effectively employ a game-based learning approach the games have to be well designed so that they can motivate and engage the learners with the game's activities. There are a large number of game motivators such as the challenge, competence, achievement, control, feedback, creativity,

---

[1]EU NEWTON Project Website [Online] Available: http://newtonproject.eu

etc., and game design principles that have to be consider when developing educational games [10].

Miljanovic and Bradbury [11] introduced *Robot ON!*, a serious game, which focuses on a few basic knowledge topics in C++ programming for students with no previous programming experience. An evaluation plan was proposed aiming to investigate the impacts of the game on both learning outcomes and enjoyment, though no results were presented.

Mathrani et al. [12] have analysed the effects of using a game-based learning methodology in a programming course in a computing diploma programme. The findings revealed that the students could easily relate gaming elements to difficult abstract programming constructs. In addition, the results of the study showed that students were highly engaged in learning. Also, some students found the use of gaming elements as a better way to express their program's logic when giving oral presentations for the final assessment.

Schmolitzky and Göttel [13] presented a game entitled *Guess My Object* which helps students understand the concept of objects in programming. Fundamental constructs such as fields and methods are introduced in the context of object state and behavior, respectively. Services are also introduced in two steps through their interfaces and implementations. Finally, despite running a pilot whose participants praised particularly the interactive aspects of the game, the results were not conclusive, as the study had a low number of participants.

The authors of [14] described a curriculum for game development as a mechanism to teach basic programming concepts. The authors conducted a pilot study using *GameMaker*, a game development engine, as part of a high school computer science course run over a six-month semester. The course consisted of three modules. In each module, the students worked in teams of three where each of them played one of the assigned roles of designer, artist or programmer to propose and develop a game. The paper presents both the methods employed and results which demonstrate the efficacy of the proposed approach for learners' comprehension of programming concepts. However, due to the low number of students no general conclusion can be drawn and further studies are required to be performed. In the proposed method, the students rotated their role (i.e. designer, artist and programmer) in each of the three modules, therefore, to effectively apply this approach each student has to be exposed to all the facets of game development in order to acquire the targeted skills.

Zhi et al. [15] investigated the design of instructional support in an educational programming game, BOTS, aimed at teaching secondary school students. The instructional support provided within BOTS consists of three different strategies, namely instructional text, worked examples, and erroneous worked examples (i.e. buggy code). The researchers employed the cognitive load theory when designing the different supports. Preliminary pilot study results showed that using buggy code was the most effective instructional support in teaching the loop programming concept as the students were actively involved in the task. For the other two instructional supports, i.e. instructional text and examples, there was not observed a statistically significant difference in their efficacy.

Dicheva and Hodge [16] presented the *Stack Game*, a programming game focused on the stack data structure. This game covers concepts of conceptualization, application and implementation of the stack data structure. A qualitative survey was run to capture student perceived usefulness, educational value, usage, clarity, support, and enjoyment of the game. The results showed that the *Stack Game* helped most students develop a better conceptual model for stacks, and that most students have a positive attitude towards game-based learning.

López-Fernández et al. [17] conducted two randomized control trials, each addressing one course topic, involving a total of 124 undergraduate students to investigate the effectiveness of traditional teaching and GBL using teacher-authored games in computer science education. The effectiveness of the two pedagogical approaches have been evaluated using pre-test, post-test and a student questionnaire. The empirical results show that while there were no statistically significant differences in the knowledge acquisition when using either of the two teaching and learning approaches, the students' perceived motivation had increased when using GBL.

Miljanovic and Bradbury [18] introduced *GidgetML*, an adaptive serious game for learning debugging, which customizes the game's levels and tasks according to a learner's predicted level of competency. The authors evaluated the adaptive game through a comparative analysis study between the use of the adaptive game and the non-adaptive game, Gidget, in a first-year programming course with 100 participants. The results from the empirical evaluation showed that there was a high variance in the learners' performance in the game both across the different levels of the game and among the different learners for those learners who played the non-adaptive game. The variance of learners' performance in the game reduced for those who played the adaptive game. However, based on the students' answers to a questionnaire on game-playing experience, it seems that the students had a neutral attitude towards the game experience with either of the two games.

In summary, existing studies have shown that educational games are an effective mechanism to increase students' motivation [17], [19] and engagement [8], improve student learning performance [20] and achieve better user experience [21]. However, many studies focus on short tasks, do not allow participant flexibility to explore diverse solutions, lack scale or participant diversity or do not have statistical significance.

In this paper, we study students' learning experience using serious games against various user profile dimensions, including by exploring the evidence in terms of statistical significance. A game-based learning approach, such as the one proposed in our study, exposes the students to critical thinking and problem-solving tasks in an interactive and fun way whilst offering them immediate real-time feedback and rewards to motivate them to discover more solutions and good practice. Solving these problems helps the students to develop skills which may be used to solve future problems in their professional career, as well as motivate and encourage the them to continue studying a particular subject. However, it should be noted that these benefits come with several limitations, mostly associated with the technology used to support game-based learning. For instance, video games may create addiction to gadgets, incorrect body posture and may lower participant interest in other activities. However, various research studies that involved interviews with teachers have shown that many of these limitations can be exceeded by providing additional support from teachers and/or parents.

III. NEWTON PROJECT AND EDUCATIONAL GAMES FOR PROGRAMMING COURSES

A. NEWTON Project

The NEWTON project is a large EU Horizon 2020 Innovation Action project which designs, develops and deploys innovative solutions for Technology-Enhanced Learning (TEL) when delivering of state-of-the-art STEM content to diverse learner audiences. NEWTON proposes innovative technologies for adaptive and personalised multimedia and multiple sensorial media (mulsemedia) delivery, Augmented and Virtual Reality (AR/VR)-enhanced learning, Virtual Teaching and Learning Labs (Virtual Labs), Fabrication Labs (Fab Labs) and Gamification. These technologies are used in conjunction with different pedagogical approaches including self-directed, game-based and problem-based learning methods.

The NEWTON project has deployed its proposed solutions in a new learning management platform, NEWTELP[2], allowing cross-European learner and teacher interaction with content and courses. The platform supports fast dissemination of learning content to a wide audience in a ubiquitous manner and using the latest technological innovations. The NEWTON project has developed proof of concept educational AR/VR applications, games, Virtual Labs and Fab Labs focused on STEM subjects, and has tested these in many small-scale and large-scale pilots in 20 primary, secondary and third level institutions, including in schools with students with special educational needs, across six different EU countries. For example, [22] presented a NEWTON small-scale pilot that utilized Fab Labs to improve students' learning experience. In [23], a small-scale pilot investigating the application of VR and Virtual Labs technology among primary school students was presented. The STEM rich media content and applications were deployed on the NEWTELP platform, including the evaluation procedure. The evaluation follows the methodology resulted from the research performed within the NEWTON project [24] and involves different aspects, including learner experience, knowledge acquisition and usability.

Among the most beneficial aspects of the NEWTON project was the deployment of serious games in real life student education. Serious games in an educational context are games designed specifically for student teaching and learning purposes. Serious games exploit the entertaining and interactive nature of games and integrate educational content with game elements to stimulate students' interests and engagement. Their goal was to increase learning motivation, make learning experience fun and keep the participants engaged. This is as many studies have shown that employing games is a powerful motivator for learning [25], [26]. These are reasons behind developing and deploying serious games that not only teach

---

[2]NEWTELP platform, [Online] Available: http://newtelp.eu

new concepts, but are also very much engaging. The serious games should keep the learners focused and not only make them to learn the provided content, but also determine the students to learn better.

B. Educational Games for Programming Courses

The two games described in this paper were designed and developed as part of the NEWTON project and targeted knowledge aspects related to the programming concepts of *variables* and *loop*, respectively. The reason for the choice of topics covered in this study is the fact that students find these topics most difficult to grasp. An easy to understand alternative approach to teaching them in a game environment was very much welcomed by the academics involved in the initial consultations. The game design style chosen was a "visual novel" style, that mostly consists of graphics elements and gameplay interaction in the context of a real scenario. The game design aimed to suit the objectives of introduction to programming syllabus as it involves providing basic programming information, explaining concepts without interrupting the flow. Academics involved in delivering Programming modules from three European universities were consulted before the design of these games and were asked to identify the key knowledge points and concepts that students struggle with. The games were then designed to target these points. The academics contributed to the design and development of the educational video games by proving regular feedback. They were also involved in the review and improvement process after the game development phase finished in order to validate the accuracy of the content present in the games and to provide feedback on usability. The two games target both C and Java programming languages. The games are 2D games, and have been developed using Unity. Both games provide interactive animations which reinforce students' understanding of the consequences of their actions and hence help acquire related knowledge. After learning programming concepts in the games, students got the chance to further exercise what they have learned in the post-game knowledge tests.

1) NEWTON Project Variable Game: The first game focuses on the concept of variables, and addresses the need for students to understand that a variable is a name given to a contiguous memory location, and the size of the memory allocated for a certain variable is dependent on the variable's data type. The students also get familiar with accessing the variables to either store or read data.

The *Variable* game is placed in a warehouse scenario (see Figure 1) and has three levels. To accommodate the differences between C and Java languages, this game is customised to cater for C and Java differences in its third level, whereas the first two levels are the same in the two versions of the game.

In Level 1, the player is brought to a sea port where several containers of different lengths are being discharged from the ship. The boxes are associated with primitive data types, such as *char*, *int*, *float*. As each box is placed on the ground, a worker character introduces a variable definition and uses the corresponding data type.

In Level 2, the containers are moved inside the warehouse and are ready to be stacked on shelves. The warehouse represents computer memory and the space on the shelves is divided in locations of 1-byte size each. In this game level, the player is directed to carry out tasks including declaring variables and assigning values to variables. Within the game this is achieved by booking shelf space and placing containers in the booked spaces.

In Level 3 of the C game version, the player is asked to repeat again the variable declaration and value assigning tasks, but assign values of unmatched data types to variables (e.g., drag the container that represents a *float* type value to the booked space associated with an *int* variable and vice-versa). Upon player's action of placing a *double* type value into the space declared for the *int* type variable, the double type value box will be automatically truncated to half its size to fit into the booked space, whereas upon player's action of placing an *int* type value into the space declared for the *double* type variable, the *int* value will automatically occupy the double size booked variable space. In Level 3 of the Java game version, the player is asked to perform the same tasks as in the C version of the game. However, upon player's action of placing a *double* type value into the space declared for the *int* type variable, a warning light will be shown and a saw will be given to the player, who should use it to cut the box into half size to fit into the space. This mimics the operation of explicit data type casting in Java. Upon player's action of placing an *int* type value into the space declared for the *double* type variable, an error message will be shown, which is in accordance with the actual operation in Java.

2) NEWTON Project Loop Game: This game introduces the concept of *loop* through interactive coin collection tasks carried out by a mermaid (player) in an undersea scenario (see Figure 2). The *Loop* game focuses on the *for* loop and visualizes repeat activities in three levels. The levels demonstrate in an interactive manner the concepts of the *for* loop, *for* loop with *continue* statement and *for* loop with *break* statement, respectively. This game has a single version that applies to both C and Java programming courses, as the pseudo-code used in the game is the same in both languages.

In a basic *for* loop, 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. In Level 1 of the game, the player needs to control the movement of the mermaid to carry out the tasks "swim to a coin – collect the coin – store the coin in the treasure chest" 5 times, which corresponds to repeating the body of the loop 5 times, as specified by the boolean condition of the loop. The code displayed on the left hand side of the screen changes along every step of the tasks to reflect which line of code is being executed in any step.

In Level 2 of the game, similar tasks are given to the player, however, some of the coins will disappear when touched by the mermaid. When this situation happens, the mermaid skips the remaining steps (i.e., moving towards the treasure chest and storing the coin). This is equivalent to continuing to the next iteration, and the players are able to visualize the operation of the *for* loop with an embedded *continue* statement.

In Level 3 of the game, the operation of a *for* loop with a *break* statement is conveyed. In this level, a Jackpot is hidden
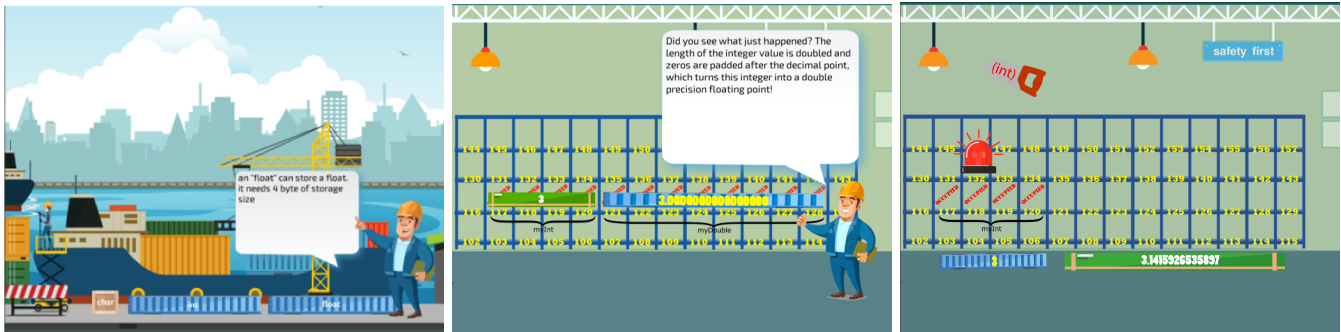
Fig. 1: Screen Captures from the NEWTON Project Variable Game

Fig. 2: Screen Captures from the NEWTON Project Loop Game

behind one of the coins, i.e., one of the coins in the sea will turn into a Jackpot once harvested by the mermaid. The mermaid's tasks in this level is still to repeat collecting and storing coins. However, during this process, once the coin with Jackpot is discovered, the mermaid "breaks" out from the loop of tasks and the level finishes immediately.

IV. THE PROGRAMMING LARGE-SCALE PILOT

A. Pilot Overview

Over 100 students from three institutions: Dublin City University (DCU) - Ireland, Slovak Technical University of Bratislava (STUBA) - Slovakia, and National College of Ireland (NCI) took part in this NEWTON project pilot. 78, 10 and 34 students took part in the *Variable* game and 65, 10 and 30 students participated in the *Loop* game in DCU, NCI and STUBA, respectively[3].

The DCU and STUBA participants are first and second year undergraduate students, respectively, and most of them are under 22 years old. NCI's participants are mature students who are over 23 years old and already hold a third level educational degreein various areas including Humanity and Education. They were enrolled into a conversion course where students were in a process of upgrading their computer science skills. Regarding gender distribution, DCU and STUBA have very high percentage of males (80% and 77%, respectively), whereas about 54% of NCI participants are male. DCU and STUBA's students are from computer science or engineering departments. NCI participants have more diverse backgrounds: 50% of them are computer scientists or engineers, whereas the rest have indicated various other areas, including Humanity and Education.

---

[3]The slight difference in the number of participants in the two games are due to some students being absent.

This pilot was deployed as part of three different Programming modules in DCU, STUBA and NCI, respectively. At DCU and NCI, the Programming modules were the first programming related modules that the students ever took, i.e., most of them have little or no prior programming knowledge/skills. The serious games were integrated as part of the lab sessions associated with the corresponding topics during the academic semester. Each game took around 10-15 minutes (including tests and surveys associated with it).

The STUBA participants were exposed to the serious games during a 30 minutes re-cap session prior to an advanced Programming module at the beginning of the semester. They had already taken the introductory Programming module in the previous academic year and they were using the serious games to refresh their knowledge of the programming concepts they had learned as preparation for the advanced module.

The purpose of testing the games in three different test places was twofold:

1) assess how students from different age groups, with different academic background and various EU countries receive the games. Note that the NCI students were mostly graduate students who previously obtained bachelor degrees in various areas, including Humanity and Education. NCI students had working experience, as they attended a part-time conversion program. DCU and STUBA students, on the other hand, were all full-time undergraduate in computer science related majors and were mostly around 18 years old. DCU and NCI were located in Ireland, whereas STUBA is from Slovakia.
2) investigate whether students using the games for revision react differently from students using them for first-time learning.

TABLE I: Evaluation methodology

Before the pilot	Demographic Questionnaire	
During the pilot	Before each game	Knowledge pre-test
	During each game	Teacher observation
	After each game	Knowledge post-test
		Questionnaire
After the pilot	Interview with teachers	
	Interview with students	

TABLE II: Questions in the post-game questionnaire

Usability
Q1. The game task and levels were properly designed.
Q2. The game user interface design is pleasant.
Q3. I understood all the different parts of the game.
Knowledge Acquisition
Q4. The game could help me achieve better results in this course.
Q5. The game targeted my knowledge gap.
Q6. The game helped me understand the programming concepts.
Q7. This game is a good complement to textbooks and lecture slides on this topic.
User Experience
Q8. This game made me more interested in this Programming course.
Q9. I would prefer to learn without serious games.
Q10. This game was really interesting.
Q11. I was distracted by the game.
Q12. This game was boring.

B. Evaluation Methodologies

In order to fully understand the impact of the serious games on students' learning experience, the following evaluation methodology was adopted in the large scale pilot (see Table I). Before the pilot started, all participants were asked to answer a demographic questionnaire which collects information about their background. During the pilot, knowledge pre- and post-tests were taken by students before and after they played each game to evaluate their learning outcome levels.

Upon finishing each game, students answered a post-game questionnaire, which includes questions related to the following aspects: 1) usability of the game; 2) the game's impact on knowledge acquisition; and 3) user experience. The questions from the post-game questionnaire are listed in Table II. Moreover, teachers were asked to write down their observations of students when they played the game. After the pilot finished, interviews with teachers and several students (volunteered) were also conducted. These enabled to discuss in-depth about students' experience and feelings in relation to the pilot.

This paper focuses on investigating students' subjective feeling and perception towards the games through analyzing the post-game questionnaire results, along with teachers' observation and interviews of teachers and students.

V. STUDY RESULTS

In this section, the overall results of the post-game questionnaires, including the combined results from all three test locations, are presented. From these results, general conclusions regarding the games' benefits and their impact on students' learning experience are drawn.

Fig. 3, Fig. 4 and Fig. 5 illustrate the questionnaire results across all participants concerning usability, knowledge acquisition and user experience, respectively. The results are presented in terms of percentage of the total number of responses for each of the possible answers as follows: SD - *Strongly Disagree*, D - *Disagree*, N - *Neutral*, A - *Agree*, SA - *Strongly Agree*. The white bars are used to represent the results related to the *Variable* game, whereas the dashed bars represent the results associated with the *Loop* game.

Overall, as it can be observed from all the figures, the results of both games are positive. Note that in order to avoid user boredom and any potential bias, some questions have been inverted and a positive result is associated with SD and D answers and not SA and A responses (e.g. Q9 and Q11).

For *Usability* related questions Q1 to Q3, *Agree* is the most popular answer for both games, attracting between 50% to 70% of all feedback, followed by *Neutral* and *Strongly Agree*, each getting around 10% to 30% votes. Negative feedback, i.e., *Strongly Disagree* and *Disagree* only got less than 10% votes in each question in both games. Such results indicate that the usability aspects of both games were well approved by participants.

For *Knowledge Acquisition* related questions Q4 to Q7, the *Variable* game got positive feedback: *Agree* is the dominant answer, attracting up to 60% votes, while *Strongly Agree* won a further 10% votes. On the other hand, negative answers *Strongly Disagree* and *Disagree* added together only account for less than 20%. The *Loop* game, also achieved positive feedback in this category, though not as conspicuous as the first game: Q4 attracted around 45% neutral opinions, though positive answers still got way more votes (40%) than negative answers (15%); Q5 got comparable amounts of neutral and agree answers (around 30% each) while *Disagree* got only 20% votes; Q6 and Q7 both attracted over 65% positive answers (*Strongly Agree* and *Agree*), while negative answers only got less than 20% votes. Overall, the educational aspects of the games achieved considerably positive opinions among participants, especially for the first game.

For *User Experience*-related questions Q8 to Q12, the overall participant feedback are again on the positive side, despite the fact positive answers may not dominant in some questions. For Q9 and Q11, the positive answers are still majority in the results, accounting for more than 50% in both games. For Q8, the *Variable* game got a positive feedback with *Agree* being the most popular answer, while the responses to the *Loop* game is more neutral, with *Neutral* being the most popular answer. For Q10 and Q12, which investigate participants' perception of the interestingness/boringness of the games, *Neutral* is the most popular opinions, however, they still attracted more positive answers than negative ones.

VI. CROSS ANALYSIS

In this section, the questionnaire results are compared among the three locations as well as between games. Moreover, to investigate the influence of study stages, the results are also compared between locations that use the games as a learning tool (i.e., DCU and NCI) and the location that

(a) Q1 - Tasks and levels properly designed

(b) Q2 - User interface design is pleasant
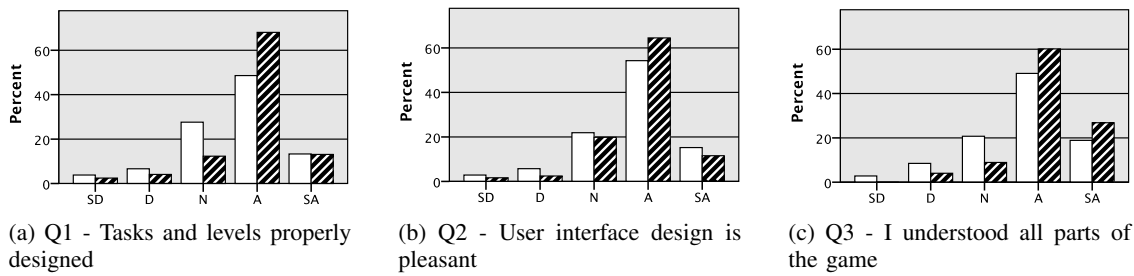
(c) Q3 - I understood all parts of the game

Fig. 3: Usability-related Questionnaire Results: Variable Game (white bars) vs. Loop Game (dashed bars)

used the games as a revision tool (i.e. STUBA). To assist the analysis, the following scoring scheme for the answers was utilized: for positively worded questions (e.g., all questions except Q9, Q11 and Q12), numerical values of 1, 2, 3, 4 and 5 are attributed to the answers *Strongly Disagree*, *Disagree*, *Neutral*, *Agree* and *Strongly Agree*, respectively; for negatively worded questions (e.g., Q9, Q11 and Q12), the numerical scoring runs in the opposite direction, i.e., answers *Strongly Disagree* to *Strongly Agree* are assigned numerical values from 5 to 1.

A. Usability

The results associated with different pilot location responses to each usability-related question are summarized in Table III. Overall, the students' responses from all three locations to all the questions in this category are rather consistent: the same median value of 4 is observed for all locations, questions and games (except the question Q1 in STUBA, which got a median value of 3), while the mean falls within the small range between 3.58 to 4.26 among different locations/questions/games. According to Table III the *Variable* game, in general, got better response than the *Loop* game in terms of usability (i.e., higher mean).

Looking closer to the results of each question, for Q1 (*properly designed tasks and levels*) and Q2 (*pleasant user interface*), the mean of responses only varies slightly among locations and there no one location is associated higher/lower mean values than others in both games. Q3 (*understood all parts of the game*) still attracted relatively close means among locations, although a slightly higher mean was observed in the STUBA results, compared with the other two locations, for both games.

Averaging the responses to Q1-Q3, very similar means among locations can be observed for both games: the average means for the *Variable* game are 3.92, 4.04 and 3.9 in DCU, NCI and STUBA, respectively, whereas the average means for the *Loop* game are 3.66, 3.67 and 3.73 in DCU, NCI and STUBA, respectively.

To consider the impact of student study stages, the means of Q1-Q3 responses were calculated for the combined group of DCU&NCI participants. Results show that the means of DCU&NCI combined group results and those of STUBA are still very close. Furthermore, Mann-Whitney U test was carried out to see whether the means of answers to usability questions (Q1-Q3 average) between the DCU&NCI group and STUBA have any statistically significant difference (see table IV). The

TABLE III: Results of each location's response to usability questions

Question	Location	Mean		Std. Deviation		Median	
		Var.	Loop	Var.	Loop	Var.	Loop
Q1	DCU	3.83	3.62	0.763	0.913	4	4
	NCI	4.20	3.60	0.422	0.843	4	4
	STUBA	3.79	3.60	0.914	1.04	4	3
Q2	DCU	3.91	3.77	0.648	0.844	4	4
	NCI	3.67	3.90	0.866	0.568	4	4
	STUBA	3.65	3.60	0.849	1.07	4	4
Q3	DCU	4.01	3.58	0.742	0.998	4	4
	NCI	4.20	3.82	0.422	0.603	4	4
	STUBA	4.26	4.00	0.710	0.947	4	4
Q1-Q3 average	DCU	3.92	3.66	0.504	0.682	4	4
	NCI	4.04	3.67	0.261	0.373	4	3.7
	DCU&NCI	3.93	3.66	0.484	0.650	4	4
	STUBA	3.90	3.73	0.622	0.814	4	3.8

TABLE IV: Mann-Whitney U test result for usability questions between DCU&NCI and STUBA

Null hypothesis	Game	Sig.	Decision
The distribution is the same in DCU&NCI and STUBA	Var.	0.768	Retain null hypothesis
	Loop	0.857	Retain null hypothesis

results show there was no significant difference between the two groups ($p = 0.768$ in the *Variable* game, $p = 0.857$) in the *Loop* game).

B. Knowledge Acquisition

Table V presents the results for the pilot instances run in each of the three locations. These results include responses to all knowledge acquisition-related questions for both games. In general, questions in this category got positive responses, as most median values are 4 and most means are above 3.5. As can be observed from Table V, different from usability questions which got rather consistent feedback, the responses to knowledge acquisition-related questions are more diverse across locations and games.

Comparing the responses associated to the two games in this category, in general the *Variable* game received better scores than the *Loop* game. For the *Variable* game, except from two medians of 3 observed in STUBA in Q4 and Q5, the medians of all other questions/locations are 4. On the other hand, more values of 3 are observed than 4 in the median values for the *Loop* game. Moreover, the *Variable* game got a higher mean than the *Loop* game for each question in each location.
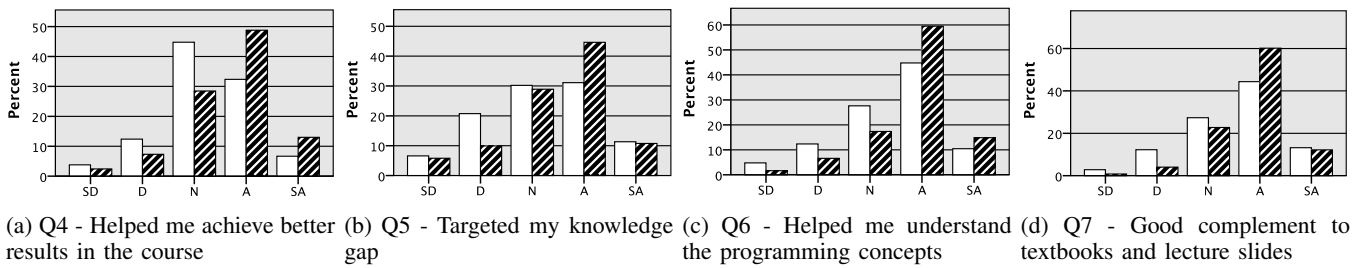
(a) Q4 - Helped me achieve better results in the course

(b) Q5 - Targeted my knowledge gap

(c) Q6 - Helped me understand the programming concepts

(d) Q7 - Good complement to textbooks and lecture slides

Fig. 4: Knowledge Acquisition-related Questionnaire Results: Variable Game (white bars) vs Loop Game (dashed bars)

(a) Q8 - Made me more interested in Programming course

(b) Q9 - Prefer to learn without serious games

(c) Q10 - This game was really interesting

(d) Q11 - I was distracted by the game
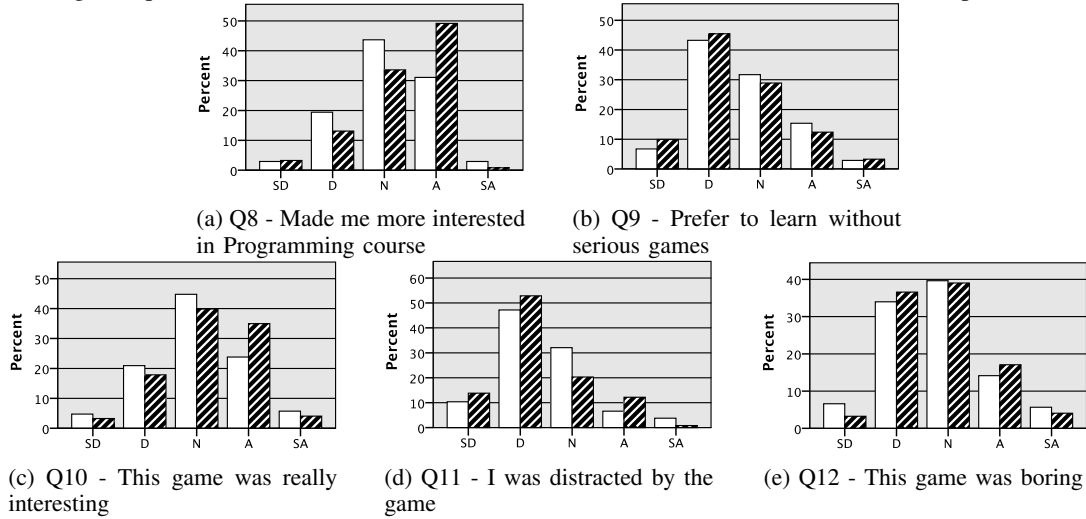
(e) Q12 - This game was boring

Fig. 5: User Experience-related Questionnaire Results: Variable Game (white bars) vs Loop Game (dashed bars)

Location-wise, it is obvious that DCU and NCI students offered better responses than those of STUBA in each question for both games. In Q4 (*game helped to achieve better results*), the mean obtained by STUBA is more than 0.5 and almost 0.8 lower than that those obtained by DCU and NCI, respectively, in the *Variable* game. In the same question, some difference is observed between STUBA's mean (3.03) and the other means (3.33 in DCU and 3.45 in NCI) in the *Loop* game. For Q5 (*game targeted my knowledge gap*), the differences in means are even more significant across locations: STUBA only scored means of 2.74 and 2.63 in the *Variable* game and the *Loop* game, respectively, while the means in DCU and NCI are well above 3. In Q6 (*game helped understand the programming concepts*), DCU and NCI obtained means as high as 3.9 and 4.2, respectively, in *Variable* game, whereas STUBA only got a mean of 3.42 in the same game. In the *Loop* game, both DCU and NCI have means with 0.3 higher than that of STUBA. In Q7 (*game is a good complement to textbooks and lecture slides*), the same phenomenon is observed: DCU and NCI students scored higher than those in STUBA in both games.

Averaging the responses across Q4-Q7, similar patterns can also be observed:

1) DCU and NCI student responses achieved higher means (3.81 and 3.44 in DCU and 4.0 and 3.61 in NCI, for the two games, respectively) than STUBA (only 3.23 and 3.02 in the two games); and
2) the *Variable* game got higher means (3.81, 4.00 and 3.23) than the *Loop* game (3.44, 3.61 and 3.02) in all locations.

These results may be due to the fact that the games are used for review in STUBA or due to the fact that they were using English, a native language for DCU and NCI students. The difference in study stages and language may influence their perception of the games' impacts on their knowledge acquisition. To further investigate such influence, the means for Q4-Q7 responses are calculated for the combined group of DCU&NCI students. Results show the means of DCU&NCI combined group and STUBA students are very different: DCU&NCI achieved means of 3.83 and 3.46 in the two games, whereas STUBA participants achieved 3.23 and 3.02 only. Furthermore, a Mann-Whitney U test was carried out to see whether these differences are statistically significant (see table VI). The result shows there were significant differences between the two groups in both games ($p = 0.000$ in *Variable* game, $p = 0.014$ in *Loop* game).

C. User Experience

The results of each location's response to each user experience related question for both games are summarized in Table VII. Overall, responses varies slightly across locations with no dominate patterns, though STUBA again seem to receive less positive results in most cases. Comparing between the two games, the *Variable* game achieved more positive feedback than the *Loop* game: the former received more means of 4 than 3 for different locations and questions, while the later obviously experienced the opposite; the former got higher means in most locations/questions than the later.

When answering question Q8 (*the game made me more interested in programming*), DCU and NCI students are in

TABLE V: Results of each location's response to knowledge acquisition questions in both games

Question	Location	Mean		Std. Deviation		Median	
		Var.	Loop	Var.	Loop	Var.	Loop
Q4	DCU	3.76	3.33	0.804	0.874	4	3
	NCI	4.00	3.45	0.471	1.036	4	3
	STUBA	3.21	3.03	1.038	0.890	3	3
Q5	DCU	3.72	3.37	0.771	0.977	4	3
	NCI	3.78	3.73	0.667	1.272	4	4
	STUBA	2.74	2.63	1.214	1.098	3	3
Q6	DCU	3.90	3.52	0.713	0.937	4	4
	NCI	4.20	3.55	0.632	1.036	4	4
	STUBA	3.42	3.21	1.032	1.114	4	3
Q7	DCU	3.82	3.60	0.712	0.915	4	4
	NCI	4.00	3.73	0.667	1.009	4	4
	STUBA	3.65	3.30	0.812	1.055	4	3
Q4-Q7 average	DCU	3.81	3.44	0.570	0.748	3.75	3.5
	NCI	4.00	3.61	0.515	0.951	4	3.5
	DCU&NCI	3.83	3.46	0.565	0.776	3.75	3.5
	STUBA	3.23	3.02	0.836	0.566	3.25	3.0

TABLE VI: Mann-Whitney U test for knowledge acquisition questions between DCU&NCI and STUBA

Null hypothesis	Game	Sig.	Decision
The distribution is the same in DCU&NCI and STUBA	Var.	0.000	Reject null hypothesis
	Loop	0.014	Reject null hypothesis

general positive, with means in the 3.17 to 3.56 range, whereas STUBA students gave neutral answers, with means just around 3. The *Variable* game got higher means than the *Loop* game in all locations.

Question Q9 responses (*prefer to learn without serious game*), related to the *Variable* game are positive (with means of 3.47, 3.9 and 3.32 in DCU, NCI and STUBA, respectively), indicating students prefer to learn with this game. The responses related to the *Loop* game from the DCU and STUBA students are also positive, with means of 3.48 and 3.23, respectively, while NCI students gave rather neutral answers (mean was 3).

The answers to question Q11 (*distracted by the game*) from DCU and STUBA students are positive for both games, with means ranging from 3.33 to 3.72. NCI students' responses saw a larger difference between the means calculated for the two games: the *Variable* game got a mean as high as 4.2, which indicates students oppose the statement, whereas the *Loop* game got a mean of 3 only, which indicates students held neutral opinions towards this statement. One possible reason behind such phenomenon may be the fact that, according to interviews, NCI students have different opinions on sound effects than the DCU and STUBA students and some NCI students found the sound effects in the *Loop* game disturbing. During the interviews, DCU students explicitly mentioned that they enjoyed the background sounds and the sounds together with the images contributed to a rich experience, helping them to memorize knowledge better. NCI students, on the other hand, stated that they were distracted by the sounds and some have muted them. This may further be caused by slightly different classroom/laboratory settings (DCU and STUBA had large rooms with many students while NCI had a much smaller and quieter room) and different demographic background, i.e, ages, education/work experience.

TABLE VII: Mean and standard deviation of each location's response to user experience questions

Question	Location	Mean		Std. Deviation		Median	
		Var.	Loop	Var.	Loop	Var.	Loop
Q8	DCU	3.38	3.17	0.756	0.773	4	3
	NCI	3.56	3.36	0.527	0.809	4	3
	STUBA	3.09	2.90	1.026	1.012	3	3
Q9	DCU	3.47	3.48	0.940	0.859	4	4
	NCI	3.90	3.00	0.738	0.894	4	3
	STUBA	3.32	3.23	1.007	1.040	3	3
Q10	DCU	3.06	3.00	0.867	0.854	3	3
	NCI	3.40	3.27	0.699	0.905	3.5	3
	STUBA	3.41	3.07	0.957	1.112	4	3
Q11	DCU	3.65	3.72	0.863	0.857	4	4
	NCI	4.20	3.00	0.919	1.095	4	3
	STUBA	3.56	3.33	0.927	0.844	4	3
Q12	DCU	3.08	3.23	0.844	0.880	3	3
	NCI	3.5	3.09	0.707	1.136	4	3
	STUBA	3.32	3.23	1.036	1.104	3	3
Q8-Q12 average	DCU	3.32	3.33	0.587	0.545	3.4	3.4
	NCI	3.73	3.15	0.547	0.566	3.8	3.2
	DCU&NCI	3.36	3.30	0.594	0.549	3.4	3.2
	STUBA	3.34	3.17	0.649	0.775	3.4	3.2

TABLE VIII: Mann-Whitney U test for user experience questions between DCU&NCI and STUBA

Null hypothesis	Game	Sig.	Decision
The distribution is the same in DCU&NCI and STUBA	Var.	0.781	Retain null hypothesis
	Loop	0.432	Retain null hypothesis

Questions Q10 (*the game was really interesting*) and Q12 (*the game was boring*) are both related to the fun and interestingness of the game. For both questions, responses from the students in all locations regarding both games got means between 3.0 to 3.5, indicating students' opinions were between neutral to positive, slightly appreciated the fun of the games.

Averaging the responses to Q8-Q12, slightly varied means among locations can be observed for both games: the average means for the *Variable* game are 3.32, 3.73 and 3.34 in DCU, NCI and STUBA, respectively; the average means for the *Loop* game are 3.33, 3.15 and 3.17 in DCU, NCI and STUBA, respectively.

In order to consider the impact of study stages, the means of Q8-Q12 were calculated for the combined group of DCU&NCI. Results show how the means of DCU&NCI combined group and STUBA are slightly different, i.e., 3.36 and 3.34 for the *Variable* game and 3.30 and 3.17 for the *Loop* game. To testify whether such differences are statistically significant, a Mann-Whitney U test was conducted (see table VIII). The results show there was no significant statistical difference between the two groups for neither of the two games ($p = 0.781$ in *Variable* game, $p = 0.432$ in *Loop* game).

VII. INTERVIEWS WITH TEACHERS AND STUDENTS

At the end of the pilot, students and teachers were invited for interviews. Their feedback during the interviews are analyzed to further exploit their experience with the games proposed and to provide more insights on the cross analysis results obtained in the previous section.

A. Usability

According to the feedback from students and teachers expressed during the interviews and in-class observations, due to the initial efforts (such as getting familiar with the platform, learning to download and load the games, getting accustomed to the operations of the games) in the first game session, most participants found no obstacles in terms of the usability of the games. Students also mentioned that the in-game instructions of the *Loop* game were less detailed than those for the *Variable* game, which slightly influenced their understanding of the games. However, this did not prevent them from using both games equally well.

B. Knowledge Acquisition

During the interviews, participants mentioned they felt that they had more control of their learning pace while playing educational games, and therefore, could absorb knowledge better. They also claimed their recall of knowledge learned through games seem to last longer and they remembered the knowledge learned from the games better. It is worth noting that STUBA's teacher and students mentioned that since they had already passed the entry-level programming course in the previous year, the knowledge in the games was easy and the pace of games was slow for them as they no longer needed detailed breakdown of knowledge for revision purposes. However, the interviews confirmed that both the details and pace were especially appreciated by the DCU and NCI students who were exposed to programming concepts for the first time. These opinions explain the significant differences observed between the results of the DCU&NCI and STUBA groups, reported in Section VI.B.

C. User Experience

According to the feedback from teachers and students received during the interviews, most participants were very interested and liked playing the games. Participants mentioned that the games added more active elements and diversity to the class. Moreover, the images and sounds helped them to engage better and enjoy the learning process. However, it was also mentioned by students that the games could be more interesting with better user interface designs and include more challenges and game elements. This is a natural user desire following a positive experience and indicates a good level of engagement.

VIII. IMPLICATIONS AND LESSONS LEARNED

Based on the results of the questionnaires and interviews with teachers and students, some implications were identified and several lessons were learned as described next.

First, educational games work better as learning tools (i.e., for students who are learning the knowledge for the first time) rather than revision tools (i.e., for students already exposed to knowledge). This is due to the different needs of the students in different phases of their studies. For first-time learners, their biggest obstacle is understanding the abstract programming concepts. Games, which present the concepts through interactive and visualized environments, are good choices to overcome such obstacles. On the other hand, for advanced learners, the priority is to refresh their memory and reinforce their knowledge, which they have had full grasp of in the past, as fast as possible. The basic concept-oriented games seem not to best match the needs of advanced learners.

Second, serious games could adopt personalization to suit different students' knowledge gaps and/or preferences (i.e. sound) to enhance their experience. Based on the questionnaire results and interviews with students, the obvious differences in students' learning status play vital roles in their experience of the games. Targeting precisely students' knowledge gaps is the key to increasing the effectiveness of games. In particular, personalization could be considered at both content-level (i.e., personalizing levels in each game) and platform level (i.e., recommending different games according to students' learning progress) in the future.

Third, in-game instructions affect users' experience. Lack of instruction may cause confusion and degrade students' understanding of the games. During the interviews with students, it was noted that the *Loop* game had less instructional content and as a result, some students found the operation of the game and the content inside the game took more time to understand than the *Variable* game. Therefore, it is important to ensure detailed instructions are given either before or within games.

IX. CONCLUSIONS AND FUTURE WORK

This paper presented a research case study that introduced two educational games, the Variable game and the Loop game, The two educational games convey key knowledge and abstract concepts from computer programming that most of the students struggle with such as loops, and variables declaration and usage. The extensive computer programming teaching experience of the lecturers involved in this research has shown that these two programming concepts are key concepts when learning programming and many students found them difficult to grasp. The teaching methodology encapsulated in the game design is to introduce the concepts incrementally in 3 stages implemented as game levels. Each level of the game builds on top of the concepts introduced in the previous level. The concepts are explained through a practical, real-life set of interactive tasks that the students have to solve. As the students can play the games multiple times, the games also enable the students to self-assess their knowledge level and to identify those elements of the studied concept they have difficulties to understand and to apply them. The case study deployed in 3 different educational institution aimed to investigate if the designed video game based teaching methodology helps the students in understanding complex and abstract concepts in programming and enhances their learning outcomes. A large-scale pilot involving over 100 students with various age and educational backgrounds from three universities located across Europe was conducted during a semester that included a programming module delivery.

The results of post-game questionnaires, which covered questions related to usability, knowledge acquisition and user experience, as well as feedback from teachers and students received during interviews were analyzed.

The pilot results showed how most students regardless of their locations believed the games were well designed in terms of aesthetics and were very good in terms of operation. The majority of students preferred using serious games, thought the games helped them understand better programming concepts, made them be more interested in the courses and helped them achieve better results. These results indicate that the proposed games served their purposes very well.

Furthermore, following the result cross analysis, it can be concluded that there are statistical significant differences in relation to knowledge acquisition between Irish participants and Slovakian students. These differences were caused by the different learning phases of the participants. No significant differences were observed in relation to usability and user experience related questions between the students in relation to their locations.

Future work includes an analysis of the knowledge test results, and assessment of the game-related knowledge improvements as perceived by students and documented through students' feedback. Additionally, we intend to integrate personalisation within the games to enhance further the students' learning experience by considering dynamically their profile and interactive behaviour.

ACKNOWLEDGMENT

REFERENCES

[1] S. Lohr, "Where the stem jobs are (and where they aren't)," *The New York Times*, Nov 2017.

[2] DIGITALEUROPE, "DIGITALEUROPE and the EC's skills strategy 2016," tech. rep., Brussels, Belgium, https://www.digitaleurope.org/resources/digitaleurope-position-paper-on-digital-skills, January 2016.

[3] E. Commission, "A new skills agenda for EUROPE," tech. rep., Brussels, Belgium, https://cor.europa.eu/en/engage/studies/Documents/New-Skills-Agenda-Europe.pdf, June 2016.

[4] G. Lee, "Which universities have the highest first year dropout rates?," *Channel 4 News*, Aug 2017.

[5] M. W. Ohland, S. D. Sheppard, G. Lichtenstein, O. Eris, D. Chachra, and R. A. Layton, "Persistence, engagement, and migration in engineering programs," *Journal of Engineering Education*, vol. 97, no. 3, pp. 259–278.

[6] K. Adu-Manusarpong, J. K. Arthur, and P. Y. O. Amoako, "Causes of failure of students in computer programming courses: The teacher learner perspective," *International Journal of Computer Applications*, vol. 77, no. 12, p. 27–32, 2013.

[7] A. E. Chis, A. Moldovan, L. Murphy, P. Pathak, and C. H. Muntean, "Investigating flipped classroom and problem-based learning in a programming module for computing conversion course," *Educational Technology & Society*, vol. 21, no. 4, pp. 232–247, 2018.

[8] J. Hamari, D. J. Shernoff, E. Rowe, B. Coller, J. Asbell-Clarke, and T. Edwards, "Challenging games help students learn: An empirical study on engagement, flow and immersion in game-based learning," *Computers in Human Behavior*, vol. 54, pp. 170–179, 2016.

[9] C. H. Muntean, J. Andrews, and G.-M. Muntean, "Final frontier: an educational game on solar system concepts acquisition for primary schools," in *Proc. IEEE 17th Int. Conf. Adv. Learning Technol. (ICALT)*, pp. 335–337, Jul. 2017.

[10] T. H. Laine and R. S. N. Lindberg, "Designing engaging games for education: A systematic literature review on game motivators and design principles," *IEEE Transactions on Learning Technologies*, vol. 13, no. 4, pp. 804–821, 2020.

[11] M. A. Miljanovic and J. S. Bradbury, "Robot on!: A serious game for improving programming comprehension," in *2016 IEEE/ACM 5th International Workshop on Games and Software Engineering (GAS)*, pp. 33–36, May 2016.

[12] A. Mathrani, S. Christian, and A. Ponder-Sutton, "Playit: Game based learning approach for teaching programming concepts," *Educational Technology & Society*, vol. 19, pp. 5–17, 04 2016.

[13] A. W. Schmolitzky and T. Göttel, "Guess my object: an'objects first'game on objects' behavior and implementation with bluej," in *Proceedings of the 2014 conference on Innovation & technology in computer science education*, pp. 219–224, ACM, 2014.

[14] K. Seaborn, M. Seif El-Nasr, D. Milam, and D. Yung, "Programming, PWNed: Using Digital Game Development to Enhance Learners' Competency and Self-Efficacy in a High School Computing Science Course," In Proceedings of the 43rd ACM technical symposium on Computer Science Education (SIGCSE '12), (New York, NY, USA), p. 93–98, Association for Computing Machinery, 2012.

[15] R. Zhi, N. Lytle, and T. W. Price, "Exploring instructional support design in an educational game for k-12 computing education," in *Proceedings of the 49th ACM Technical Symposium on Computer Science Education*, pp. 747–752, ACM, 2018.

[16] D. Dicheva and A. Hodge, "Active learning through game play in a data structures course," in *Proceedings of the 49th ACM Technical Symposium on Computer Science Education*, pp. 834–839, ACM, 2018.

[17] D. López-Fernández, A. Gordillo, P. P. Alarcón, and E. Tovar, "Comparing traditional teaching and game-based learning using teacher-authored games on computer science education," *IEEE Transactions on Education*, vol. 64, no. 4, pp. 367–373, 2021.

[18] M. A. Miljanovic and J. S. Bradbury, "Gidgetml: An adaptive serious game for enhancing first year programming labs," in *2020 IEEE/ACM 42nd International Conference on Software Engineering: Software Engineering Education and Training (ICSE-SEET)*, pp. 184–192, 2020.

[19] M. Zapata-Cáceres and E. Martín-Barroso, "Applying game learning analytics to a voluntary video game: Intrinsic motivation, persistence, and rewards in learning to program at an early age," *IEEE Access*, vol. 9, pp. 123588–123602, 2021.

[20] J. C. Burguillo, "Using game theory and competition-based learning to stimulate student motivation and performance," *Computers & Education*, vol. 55, no. 2, pp. 566–575, 2010.

[21] R. B. Hijon-Neira, Á. Velázquez-Iturbide, C. Pizarro-Romero, and L. Carriço, "Game programming for improving learning experience," in *Proceedings of the 2014 conference on Innovation & technology in computer science education*, pp. 225–230, ACM, 2014.

[22] M. Togou, C. Lorenzo, E. Lorenzo, G. Cornetta, and G. Muntean, "Raising students' interest in stem education via remote digital fabrication: An irish primary school case study," in *the 10-th annual International Conference on Education and New Learning Technologies (EDULEARN)*, (Palma De Mallorca, Spain), pp. 2835–2840, 2018.

[23] D. Bogusevschi, I. Tal, M. Bratu, B. Gornea, D. Caraman, I. Ghergulescu, C. Muntean, and G. Muntean, "Water cycle in nature: Small-scale stem education pilot," in *EdMedia World Conference on Educational Media and Technology*, (Amsterdam, The Netherlands), pp. 2835–2840, 2018.

[24] L. Montandon, J. Playfoot, I. Ghergulescu, M. Bratu, D. Bogusevschi, R. Rybarova, and N. El Mawas, "Multi-dimensional approach for the pedagogical assessment in stem technology enhanced learning," pp. 378–383, June 2018.

[25] M. Oren, S. Pedersen, and K. L. Butler-Purry, "Teaching digital circuit design with a 3-d video game: The impact of using in-game tools on students' performance," *IEEE Transactions on Education*, vol. 64, no. 1, pp. 24–31, 2021.

[26] C. Muntean, D. Bogusevschi, and G.-M. Muntean, "Innovative technology-based solutions for primary secondary and tertiary stem education," *Paragon Publishing, UK*, 2019.

Dan Zhao is currently an Assistant Researcher with Peng Cheng Laboratory, Shenzhen, China. She was a Postdoctoral Researcher with School of Electronic Engineering, Dublin City University and with School of Computing, National College of Ireland when she worked on the EU Horizon 2020 NEWTON project. Dr Dan Zhao obtained her bachelor's degree from Beijing University of Posts and Telecommunications in 2011, majored in Telecommunications. In 2015, she graduated from Queen Mary University of London with a Ph.D. degree. She joined the NEWTON Project in 2016. Current research interests of Dr Zhao include the impact and benefit of technology-enhanced learning on students' learning outcome and experience.

Cristina Hava Muntean is a Senior Lecturer with School of Computing, National College of Ireland. She holds a B.Eng. degree in Computer Science awarded by Politehnica University of Timisoara, Romania in 2000 and a Ph.D. degree awarded by Dublin City University, Ireland in 2005. Dr Cristina Muntean has been constantly involved in various research related activities over the past 16 years fostering and promoting research, leading research projects, supervising PhD and MSc students and publishing over 100 publications in international peer-reviewed books, journals and conferences. Adaptive and personalised e/m- learning, user modelling, technology enhanced learning, game-based learning, self-directed learning, consumer behaviour, end-user Quality of Experience, adaptive multimedia, energy saving solutions are the main research areas she is involved in. Dr. Muntean chaired or served as technical program committee member for top international conferences and acted as reviewer for several journals. She was the NCI coordinator of the EU Horizon 2020 project NEWTON.

Adriana E. Chis is a Lecturer in the School of Computing at National College of Ireland. She received a Ph.D. in Computer Science from University College Dublin, Ireland in 2013. She received her Diploma-Engineer (Honors) title in Computer Science and Engineering from the Faculty of Automation and Computers, "Politehnica" University of Timisoara, Romania in 2007. Her research interests include program analysis, compilers, runtime systems, cloud computing, data-intensive systems, and computer science education.

Gregor Rozinaj is a Professor and Director at the Multimedia ICT Institute of FEI Slovak University of Technology (STU) in Bratislava, Slovakia. He graduated from STU in 1981 and started to work as an assistant professor there. In 1990 he was awarded a PhD degree, in 1998 he became an associate professor and since 2014 he is a full professor, all at STU. Since 2015 he has been a director of the Institute for Multimedia ICT at STU. He has spent two years in the Research Centre Alcatel in Stuttgart and two more years at the University of Stuttgart working in the research area of Speech and Image Processing. He was also as a vice-dean for international relations at the FEI STUBA. His research interests include telereality, multimedia and speech processing, HCI (Human Computer Interaction) based on Multimodal Interface. He published more than 130 research articles (chapters in books, papers in journals, international conferences). He is an author of four patents, three of them patented worldwide. He has participated in several educational projects under TEMPUS PHARE, LdV, ERASMUS+ and other EC programmes and as a trainer in several LLP EC and national projects. He is experienced as a coordinator and manager of many national/international research development projects oriented to multimedia processing and multimedia services. He was a technical coordinator of FP7 project HBB-NEXT. He was the local coordinator of the EU Horizon 2020 project NEWTON.

Gabriel-Miro Muntean is a Professor with the School of Electronic Engineering and co-Director of the Performance Engineering Laboratory at Dublin City University, Ireland. He obtained his Ph.D. degree from Dublin City University, Ireland for research in quality-oriented adaptive multimedia streaming in 2004. Prof. Muntean's research interests include quality and performance-related issues of adaptive multimedia streaming, performance of content delivery over wired and wireless networks and with various devices, and energy-aware networking. He is an author of 4 books, 6 edited books and over 450 peer-reviewed international journal and conference papers. Prof. Muntean is an Associate Editor of the IEEE Transactions on Broadcasting and the Multimedia Communications Area Editor for the IEEE Communications Surveys and Tutorials. He was coordinator of the EU Horizon 2020 project NEWTON and is DCU principal investigator of the EU project TRACTION.