CTE Workshop Proceedings, 2022, Vol. 9: CTE-2021, pp. 28-38

Computer modeling of the tournament of game algorithms in the process of learning of basics of algorithmization and programming by pre-service IT-specialists

Liudmyla E. Gryzun[1], *Oleksandr V.* Shcherbakov[1] and *Svitlana H.* Lytvynova[2]

[1]*Simon Kuznets Kharkiv National University of Economics, 9A Nauky Ave., Kharkiv, Ukraine*
[2]*Institute for Digitalisation of Education of the National Academy of Educational Sciences of Ukraine, 9 M. Berlynskoho Str., Kyiv, 04060, Ukraine*

Abstract. The problems of contemporary IT specialists' training in terms of the high requirements to their computational thinking skills as well as the urgency of raising their motivation to mastering algorithmization and programming are discussed in the paper. It is emphasized by the authors that initial university courses should focus pre-service IT-specialists on the deep understanding of an algorithmic nature of any coding task, to realize basic characteristics of the algorithms, to understand their role in modern software development. Due to the contemporary demands, programming should rest on algorithms building and has to be a part of lager scale experiences in order to realize its full potential. One of such experiences offered by the authors in the paper is involving the students into specially arranged activity focused on efficient game algorithms creation and simulation of the tournament between the algorithms. The offered activity is elaborated based on the applying the gamification elements into the learning process. Basing on the core gamifiaction principles, there were thought over and arranged an activity involving the students into the creation of gamified products. In our case, the gamified product which the students had to develop in the process of learning of algorithmization and programming was the software platform which enables a computer simulation of the tournament between the different game algorithms which realize winning strategies. The peculiarities and the stages of the said activity are covered in details along with the description of the final software product. Analyzing the described functionality of the computer simulator of the algorithms tournaments based on the gamification ideas, we can emphasize its significant didactic facilities in the context of its using for IT-specialists training. In particular, the developed gamified product was probed in the process of other students' mastering algorithmization and programming as well as of the schoolchildren training during summer IT schools. The prospects of the research are outlined in the lines of using the obtained results for holding the empirical research for the verification of offered activity impact on the results of IT-specialists training.

Keywords: IT-specialists' training, computer modeling, tournament of algorithms, computer simulator, winning strategy

Lgr2007@ukr.net (L. E. Gryzun); oleksandr.shcherbakov@hneu.net (O. V. Shcherbakov); s.h.lytvynova@gmail.com (S. H. Lytvynova)
https://kafis.hneu.net/grizun-lyudmila-eduardivna/ (L. E. Gryzun); https://kafis.hneu.net/shherbakov-oleksandr-vsevolodovich/ (O. V. Shcherbakov); https://iitlt.gov.ua/structure/departments/technology/detail.php?ID49 (S. H. Lytvynova)
0000-0002-5274-5624 (L. E. Gryzun); 0000-0002-6473-7629 (O. V. Shcherbakov); 0000-0002-5450-6635 (S. H. Lytvynova)

1. Introduction

Rapid growing of IT industries and their spreading in all spheres of social life needs renewing approaches to modern computing education, which would provide national economy with advanced IT-specialists who are able not just to develop efficient software but also have special type of computational thinking.

In the international guidelines shaping the modern paradigms of global IT education, it is emphasized the transformation of computing which nowadays is not a single knowledge domain [2–4, 11, 14, 21]. According to the current state of IT education covered in the latest Computing Curricula Series Report [4, 11], we can conclude that computing area (as a result of rapid evolution) is not an isolated domain, but rather a family of knowledge domains which embraces a set of fundamental and applied study areas. They go on their developing that leads to emerging of new study areas which represent intersection of fundamental science, applied branches of science and traditional computer disciplines. In particular, according to recent studies, the advantages of computer science education nowadays lie in giving the students computational thinking skills which allow them to comprehend a problem in its complexity, to formalize it and develop possible solutions in a way understandable by a machine, a human (or both).

On the other hand, the contemporary evidence of educational practice testifies that IT-specialists' training at universities is facing the problems of raising students' motivation to mastering algorithmic basics of coding, which is considered to be a part of computational thinking [20]. It is defined by the psychologists [13], that computational thinking embraces four cornerstones one of which is and algorithm developing along with decomposition, pattern recognition and abstraction.

In this context it is important to mind that initial university courses should focus pre-service IT-specialists on the deep understanding of an algorithmic nature of any coding task, to realize basic characteristics of the algorithms, to understand their role in modern software development. It is essential at this level to cultivate the students' understanding that coding is not done in a vacuum: just knowing a programming language does not provide you with problem-solving skills necessary for real-life tasks. Programming should rest on algorithms building and has to be a part of lager scale (project-based) experiences in order to realize its full potential.

One of such experiences may be involving the students into specially arranged activity focused on efficient game algorithms creation and simulation the tournament between the algorithms. The purpose of such an activity including the elements of gamification is to give young developers understanding the difference between pure coding knowledge and the algorithmization skills that allow to use computational thinking creatively and effectively.

The aim of the paper is to represent the case of the authors' practice of involving pre-service IT-specialists into computer simulation of the tournament between the different game algorithms which realize winning strategies.

2. Theoretical framework

According to a number of studies, gamification expects applying the game elements to encourage trainees' core innate needs for internal motivation [5, 18, 19]. These needs include, in particular: relatedness as the universal need to interact and be connected with others; competence as the universal need to be efficient and master a problem in a certain environment; autonomy: the universal need to control one's own activity.

The researchers also indicate that gamification methods are successful in fostering collaboration, especially when following the principles of self-determination theory [18]. It is also emphasized that applying only rewarding aspect of gamification (badges, bonuses or other rewards) [15], is not didactically beneficial, and instead gamification is recommended to consider the motivation of the participants, the goals of the course and gameful design together [7].

Gamification nowadays is widely used in educational practice of coding mastering both at school and university levels. Gamification refers to the use of game elements in a non-game context in order to increase communication between people and computers and to solve various problems effectively [8–10, 16, 17]. It is also emphasized that gamification can be described as using pieces of games to motivate learners, but the real definition of gamification involves using game-based mechanics, aesthetics, and game thinking to engage users, motivate action, promote learning, and solve problems. However, according to studies [6, 12], mostly gamification includes renaming grades for all assessments to "experience points" or "badges", a weekly compulsory and non-compulsory activities and a leaderboard, which is often integrated with an online management tools, and including other game elements without deep influence on the content of learning.

Along with such a common use of gamification method in programming learning, there are some studies which demonstrate the experience of a different gamification, involving the students into the creation of gamified products. Basing on the number of papers [6, 17], we learnt basic principles of arrangement of such an activity on design of gamified product which trig gamification mechanisms.

Gamification should have a special meaning for the trainees, the ability to inspire them to master the topic (course) and it should be autonomous providing a feeling of free choice [6]. It should also focus on meaningful victories, a sense of discovery, social interactions and include visually pleasing elements [12].

For the potential users of the gamified product it is essential to have a personal connection to it, they must feel that the game has a purpose for them individually (the final goal must be one that the users wish to achieve). The users must be able to clearly see and track their progress toward long and short term goals, in search of the final goal. To motivate the users to achieve these goals, gamified product should implement an accomplishment-based reward system. The prospect of winning a reward inspires users to work toward aims: the rewards act as self-affirmation symbols as well as allowing users to identify with a group that works toward the same goals [1]. Leader boards should be incorporated to demonstrate the user his achievements compared to others in the same community and also create both competition and a sense of belonging to a similar minded group [1]. The reward system should be transparent and designed properly so as to maximize user's enjoyment.

It is also emphasized in the papers, that creating a gamified product it is important to mind

that gaming is an unrestricted pass-time, and main objective is to motivate the users to go on playing the game and trying to overcome difficulties [6].

The techniques used in said research made the theoretical framework of our practical experience.

3. Computer simulator of the tournament between the different game algorithms

Basing on the covered principles, we thought over and arranged an activity involving the students into the creation of gamified products.

In our case, the gamified product which the students had to develop in the process of learning of algorithmization and programming was the software platform which enables a computer simulation of the tournament between the different game algorithms which realize winning strategies.

The students involved in the activity are pre-service specialists of the "Software engineering" who study the course of fundamentals of algorithmization during one term and the course of programming during two terms. The activity which was organized in its group form was offered to the students in the second term. So, the trainees had an opportunity to master algorithms and coding basics; learn common algorithms properties, methods, algorithmic solutions of classical informatics tasks etc., got familiar with some games and winning strategies for them.

On the preparation stage of the activity on creation the gamified product, the task for the students was specified as following: to develop a computer simulator of the tournament between the different game algorithms which realize winning strategies.

We would like to emphasize that the peculiarity of the task is to model the game process where the competitors are different algorithms but not a computer and humans (or humans controlled by a computer).

After the analysis of the subject domain and minding the principles of gamification (covered above), the use case diagram was built and functional requirements to the simulator were formulated.

In particular, the simulator has to provide users (students or schoolchildren who learn algorithmization and programming) with the opportunities to:

- choose a game (Tic-tac-toe, Battleship or other);
- determine the pitch dimensions and other game parameters;
- add a file with user's own algorithm which realizes the winning strategy of the game and wants to take part in tournament;
- simulate the tournament between available algorithms detecting the winner;
- visualize the progress of the tournament showing the moves of each participating algorithm;
- provide users with the results of the tournament at the leaderboard showing the name of the algorithm-winner;
- provide users with clear and friendly interface encouraging them to go on mastering the algorithms developing and achieving the victory among peers' algorithms.

After that the students were involved into direct activity on the computer simulator design.

Omitting the technical details, we could characterize the developed software as following. As a result, the computer simulator is a cross-platform software built in Python language which was learnt by the students within the course of programming. There were developed the modules and functions which enable to realize all the simulator functionality formulated above.

On the whole, the simulator is composed of three executing files: design.py, scripts.py and index.py.

The file scripts.py stores initial states of the game, default values of variables, and realized checks of winning combinations of the games.

The file design.py contains the interface and provides the proceeding of the simulator interaction with a user.

The file index.py is a main file for running the simulator. In terms of non-functional requirements, the user should make sure that the OS is installed Python programming language translator version not lower than 3.8, PyQt5 modules, Numpy and additional libraries from the list given in the requirements.txt file. For Windows, if you have a translator, just run the file Project121 (win32) .exe.

The executable file will automatically check for availability additional libraries and will install the missing modules. At restart, modules will no longer be loaded. If you are using the Linux family, launch the program is carried out from the window of the terminal emulator.

After the simulator running the user is invited to choose the game for the tournament. So far, it is realized the opportunity to hold the algorithms tournament for Tic-tac-toe and Battleship games (the others are being developed).

After the game choosing, the proper game's window is opened where a user can specify the parameters of the game. For example, after choosing the Tic-tac-toe game (figure 1), it is possible to set:

- the dimensions of the pitch where the algorithms will "play" (width and length which are integers less than 20),
- the number of figures in row to win (an integer that is not greater than minimum value of width and length),
- the number of rounds where a round is made by two games which differ by the first move of the algorithms that are competing.

In order to start the tournament, the user can choose the existing algorithms in the simulator or add his own algorithm realizing his own winning strategy. On this stage of the tournament it is possible to set up the speed of the algorithms visualization and switch on step-by-step mode of their realization.

The tournament on the comparing of winning strategies can start on condition of existing two or more algorithms. In order to add their own algorithm in the simulator to take part in the tournament, a user has to create file *.py with the developed function called *Algorithm*, according to the certain format. In particular, the function must depend on five parameters (*matrix, height, width, player, winCount*), where:

matrix — matrix (height, width), in which each element *matrix[i][j]* takes the value 1, if nobody has taken this cell yet, else this element of *matrix* will take proper symbol ("x" or "o"), latin symbols type;

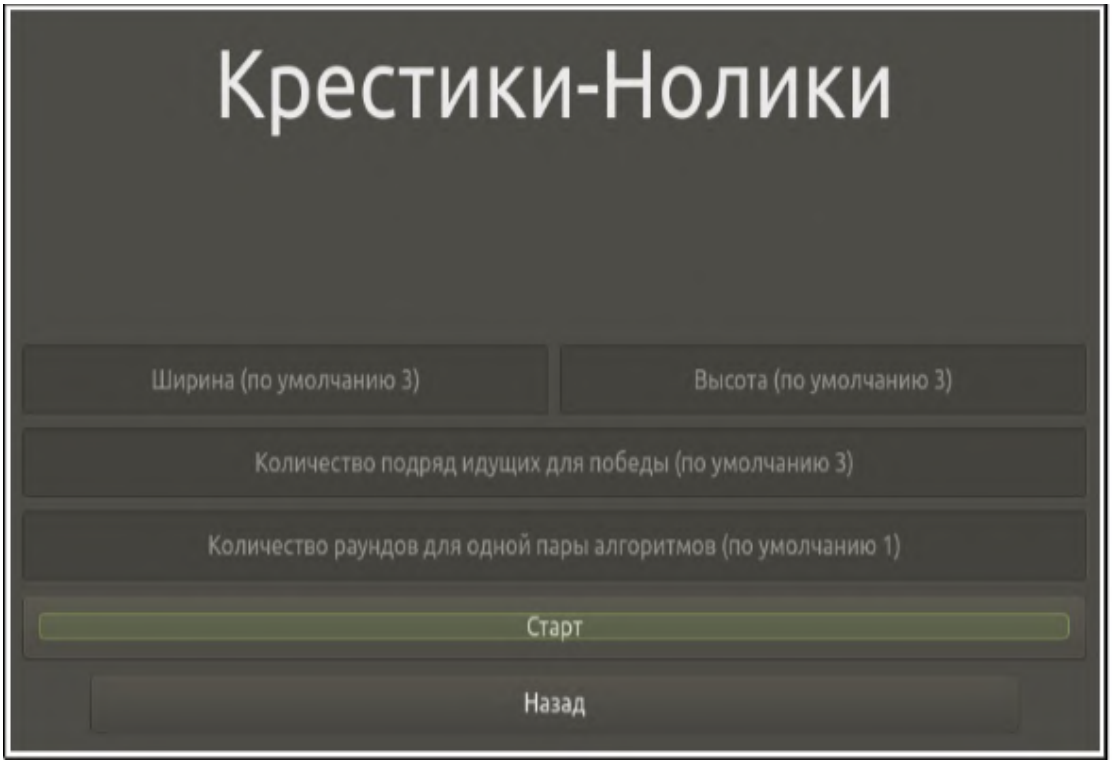

Figure 1: Tic-tac-toe game initial window where a user can set the game parameters.

height — height of the pitch, *int* type;

width — width of the pitch, *int* type;

player — the value which is put in the cell of the pitch by this algorithm, in other words, the value determines the type of this player ("x" or "o");

winCount — the number of figures ("x" or "o") in row which are necessary to win, *int* type.

The user's function contains the algorithm of winning strategy: it has to process the data sending via its parameters, make a move (change one empty cell of the pitch (*matrix*)), and return a new state of the pitch. The format of the function with its example is provided by the simulator in special *Help* section for a user.

When the file *.py with the developed function *Algorithm* is ready, a user can add his algorithm to take part in the tournament via the button Add algorithm (figure 2). The episode of algorithm adding is shown in the figure 3.

Then a user can run the tournament with his algorithm participation.

Thus, the developed software provides the simulation of the algorithms tournament for the chosen game by comparing the algorithms' winning strategies.

The progress of the game is visualized in chosen speed showing the moves of each participating algorithm.

Finally, the simulator provides users with the results of the tournament at the leaderboard showing the name of the algorithm-winner.

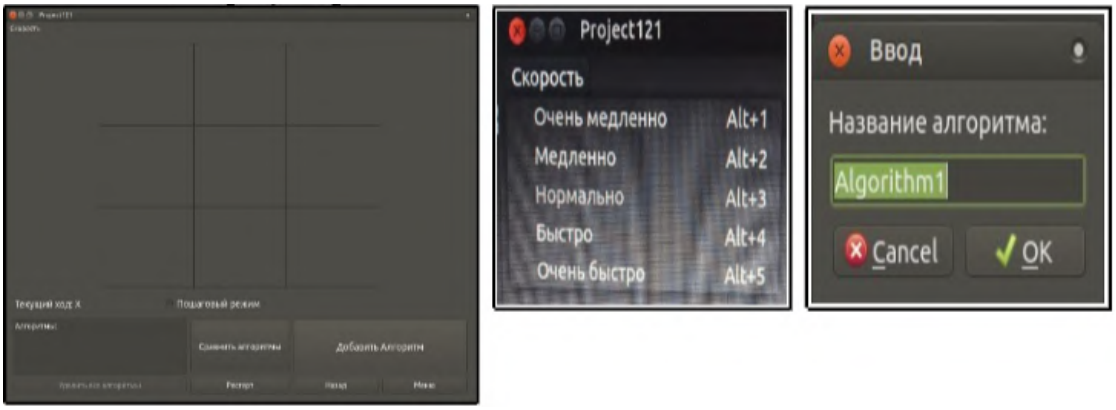

Figure 2: Tic-tac-toe tournament at the stage of choosing the algorithms and parameters of their visualization.

Figure 3: The episode of algorithm adding to take part in the tournament of algorithms.

The simulation of tournaments for Battleship game was realized by the students in similar way which gives potential users similar opportunities as for their algorithmic skills working out (figure 4).

However, the user has to create and add the file with two developed functions: *shipsPlacement* (which determines ships location) and *Algorithm* (which realizes the winning strategy).

It is important to emphasize that during the simulator developing the students had to test the work of their software on their own samples of algorithms as well as on the other their peers' algorithms. It made students themselves be involved into game activity testing gamified product and besides that, allowed them to involve other students into the process of creation of the algorithms which realize winning strategies for famous games. This circumstance showed great benefit of gamification techniques using in the mastering of the said courses in their different variations.

Thus, as a result of the involving the students into specially arranged activity focused on game algorithms creation and development of simulator of the tournament between the algorithms we could conclude the impact of such an activity on the forming trainees' computational thinking.

It was prepared by us special program of monitoring the students' activity during different stages of their work upon the simulator including its testing and preparing the user's guidelines. Our monitoring, according to this program, testified raising the students' motivation and tendency to comprehend the essence of algorithms building. As it was predicted, we could

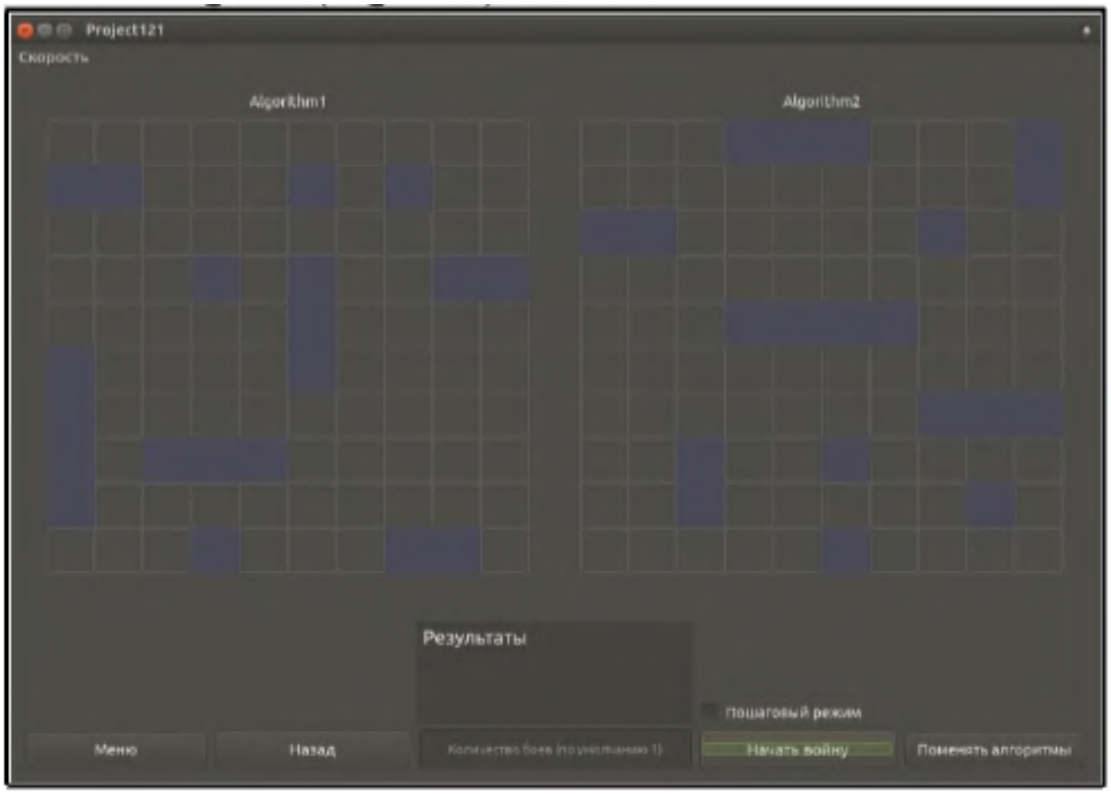

Figure 4: The episode of the tournament of the algorithms for the Battleship game.

notice their growing understanding of the role of algorithmization skills in the process of coding.

In addition, we held the survey which aimed at revealing students' hindsight and reflexion of their work upon the simulator. The survey included the set of questions and tasks, such as:

(1) How do you estimate your degree of attraction of algorithms basics in the progress of developing the said gamified product (according to the given scale)?
(2) Did you change your mind on the value of the fundamental knowledge on algorithms for coding? In which way?
(3) How do you estimate your level of engagement in analyzing feedback of your activities (according to the given scale)?
(4) What experience (besides alorithmization and coding) did you earn during the work upon the gamified product?
(5) How far does this experience seem to be essential for your future career?, and others.

Characterizing in brief the results of the survey, we can emphasize the following. 75% of the trainees agreed that the developing of gamified product encouraged them to attract algorithms basics in more concentrated way. Over the half of the students admitted changes in their understanding of practical value of the abstract fundamental knowledge. They also said that they were more engaged in analyzing feedback for activities in order to improve and earn

more experience in the future. Finally, about 80% of the students admitted that such an activity provided them with precious experience of team work and communication with potential users.

The obtained results of our monitoring and survey may be taken as a basis for holding the empirical research for the verification of the offered activity impact on the results of IT-specialists' training, which makes a prospect of our research.

4. Conclusions

The problems of contemporary IT specialists' training in terms of the high requirements to their computational thinking skills as well as the urgency of raising their motivation to mastering algorithmization and programming are discussed in the paper. It is emphasized by the authors that initial university courses should focus pre-service IT-specialists on the deep understanding of an algorithmic nature of any coding task, to realize basic characteristics of the algorithms, to understand their role in modern software development. Due to the contemporary demands, programming should rest on algorithms building and has to be a part of lager scale experiences in order to realize its full potential.

One of such experiences offered by the authors in the paper is involving the students into specially arranged activity focused on efficient game algorithms creation and simulation of the tournament between the algorithms. The offered activity is elaborated based on the applying the gamification elements into the learning process.

Basing on the core gamifiaction principles, there were thought over and arranged an activity involving the students into the creation of gamified products. In our case, the gamified product which the students had to develop in the process of learning of algorithmization and programming was the software platform which enables a computer simulation of the tournament between the different game algorithms which realize winning strategies.

The peculiarities and the stages of the said activity are covered in details along with the description of the final software product. Analyzing the described functionality of the computer simulator of the algorithms tournaments based on the gamification ideas, we can emphasize its significant didactic facilities in the context of its using for IT-specialists training. In particular, the developed gamified product was probed in the process of other students' mastering algorithmization and programming as well as of the schoolchildren training during summer IT schools.

The prospects of the research are outlined in the lines of using the obtained results for holding the empirical research for the verification of offered activity impact on the results of IT-specialists' training.

References

[1] Antin, J. and Churchill, E., 2011. Badges in social media: A social psychological perspective. *Chi 2011 workshop gamification: Using game design elements in non-game contexts*. Vancouver, Canada, pp.49–54. Available from: http://gamification-research.org/wp-content/uploads/2011/04/03-Antin-Churchill.pdf.

[2] Association for Computing Machinery and IEEE Computer Society, 2015. *Software engineering curricula 2014: Curriculum guidelines for undergraduate degree programs in software engineering*, Computing Curricula. Available from: https://www.acm.org/binaries/content/assets/education/se2014.pdf.

[3] Association for Computing Machinery and IEEE Computer Society, 2016. *Computer engineering curricula 2016: Curriculum guidelines for undergraduate degree programs in computer engineering*. Available from: https://www.acm.org/binaries/content/assets/education/ce2016-final-report.pdf.

[4] Association for Computing Machinery and IEEE Computer Society, 2020. *Computing curricula 2020: Paradigms for global computing education*. Available from: https://www.acm.org/binaries/content/assets/education/curricula-recommendations/cc2020.pdf.

[5] Buzko, V.L., Bonk, A.V. and Tron, V.V., 2018. Implementation of gamification and elements of augmented reality during the binary lessons in a secondary school. *Ceur workshop proceedings*, 2257, pp.53–60.

[6] Deterding, C.S., 2011. Meaningful play: Getting gamification right. Video. Available from: https://www.youtube.com/watch?v=7ZGCPap7GkY.

[7] Deterding, S., 2015. The lens of intrinsic skill atoms: A method for gameful design. *Human–computer interaction*, 30(3-4), pp.294–335. Available from: https://doi.org/10.1080/07370024.2014.993471.

[8] Deterding, S., Dixon, D., Khaled, R. and Nacke, L., 2011. From game design elements to gamefulness: Defining "gamification". *Proceedings of the 15th international academic mindtrek conference: Envisioning future media environments*. New York, NY, USA: Association for Computing Machinery, MindTrek '11, p.9–15. Available from: https://doi.org/10.1145/2181037.2181040.

[9] Deterding, S., Sicart, M., Nacke, L., O'Hara, K. and Dixon, D., 2011. Gamification. using game-design elements in non-gaming contexts. *Chi '11 extended abstracts on human factors in computing systems*. New York, NY, USA: Association for Computing Machinery, CHI EA '11, p.2425–2428. Available from: https://doi.org/10.1145/1979742.1979575.

[10] Fedorenko, E.G., Kaidan, N.V., Velychko, V.Y. and Soloviev, V.N., 2021. Gamification when studying logical operators on the Minecraft EDU platform. *Ceur workshop proceedings*, 2898, pp.107–118. Available from: http://ceur-ws.org/Vol-2898/paper05.pdf.

[11] Geissler, M., Brown, D., McKenzie, N., Peltsverger, S., Preuss, T., Sabin, M. and Tang, C., 2020. *Information technology transfer curricula 2020: Curriculum guidelines for two-year transfer programs in information technology*. New York, NY, USA: Association for Computing Machinery. Available from: https://ccecc.acm.org/files/publications/IT-Transfer2020.pdf.

[12] Hoonhout, J. and Meerbeek, B., 2011. Brainstorm triggers: game characteristics as input in ideation. *Chi 2011 workshop gamification: Using game design elements in non-game contexts*. Vancouver, Canada, pp.49–54. Available from: http://gamification-research.org/wp-content/uploads/2011/04/13-Hoonhout.pdf.

[13] Introduction into computational thinking, 2021. Available from: https://www.bbc.co.uk/bitesize/guides/zp92mp3/revision/1.

[14] Joint Task Force on Cybersecurity Education, 2017. *Cybersecurity curricula 2017: Curriculum guidelines for post-secondary degrees in cybersecurity*. Available from: https://www.acm.org/binaries/content/assets/education/curricula-recommendations/csec2017.pdf.

[15] Kapp, K.M., 2012. *The gamification of learning and instruction: game-based methods and strategies for training and education.* Wiley.

[16] Khaleel, F.L., Ashaari, N.S., Meriam, T.S., Wook, T. and Ismail, A., 2015. The study of gamification application architecture for programming language course. *Proceedings of the 9th international conference on ubiquitous information management and communication.* New York, NY, USA: Association for Computing Machinery, IMCOM '15. Available from: https://doi.org/10.1145/2701126.2701222.

[17] Khaleel, F.L., Ashaari, N.S., Tengku Wook, T.S.M. and Ismail, A., 2015. User-enjoyable learning environment based on gamification elements. *2015 international conference on computer, communications, and control technology (i4ct).* pp.221–226. Available from: https://doi.org/10.1109/I4CT.2015.7219570.

[18] Knutas, A., Ikonen, J., Nikula, U. and Porras, J., 2014. Increasing collaborative communications in a programming course with gamification: A case study. *Proceedings of the 15th international conference on computer systems and technologies.* New York, NY, USA: Association for Computing Machinery, CompSysTech '14, p.370–377. Available from: https://doi.org/10.1145/2659532.2659620.

[19] Seaborn, K. and Fels, D.I., 2015. Gamification in theory and action: A survey. *International journal of human-computer studies*, 74, pp.14–31. Available from: https://doi.org/10.1016/j.ijhcs.2014.09.006.

[20] Seidametova, Z., 2020. Combining programming and mathematics through computer simulation problems. *Ceur workshop proceedings*, 2732, pp.869–880.

[21] Task Group on Information Technology Curricula, 2017. *Information technology curricula 2017: Curriculum guidelines for baccalaureate degree programs in information technology.* New York, NY, USA: Association for Computing Machinery.