International Conference Cybernetics & Informatics 2022
Visegrád, Hungary, September 11th - September 14th, 2022

Application for Python Programming Language Education Developed by Unity Engine

Michal Hlavatý, Alena Kozáková and Oto Haffner
Faculty of Electrical Engineering and Information Technology
Slovak University of Technology in Bratislava
Bratislava, Slovakia
Email: michal_hlavaty@stuba.sk

***Abstract*— Programming is not easy to learn or teach because different students need different explanations to understand such a complex concept. Traditional teaching methods and learning approaches are no longer appropriate for many kids. According to studies, such approaches impair students' motivation and interest in studying programming. According to research, games can help learn Python programming. The aim of this paper is to present our educational game that we developed in the Unity 3D game engine. The game focuses on pure basics of programming in Python language. Before and after completing the game the player has to fill out a survey that helped us analyze the project.**

Keywords—programming, Python, game, education

I. Introduction

The fast acceptance of gamification and the usage of game features in education has a wide range of implications on student outcomes and engagement. Gamification is defined as the intentional use of game design features in nongame contexts to promote desired behavior or solve problems. The concept of gamification has inspired a plethora of insightful studies attempting to comprehend its applicability in education and all aspects of life [1].

The quick development of technology always results in novel and exciting ways to engage students in learning and meet the expanding demands of education. However, traditional teaching strategies or even current software programs are consistently shown to be either ineffective or insufficient [2]–[4]. According to the National Research Council [5], science education is widely regarded as one of the most essential components of modern education because it is in charge of preparing citizens who are scientifically literate and encouraging essential 21st-century skills like adaptability or problem-solving. As a result, there has been a lot of interest in tools and methods that encourage scientific thinking as well as the educational theories that are incorporated into them [6]–[9].

One of the most well-known definitions of a game comes from game designer and producer Sid Meier, who says „a game is a series of interesting and meaningful choices made by the player in pursuit of a clear and compelling goal". A game is typically characterized as an activity aimed at achieving a specified state of things using only methods authorized by specific rules, where the means specified by the rules are more limited in scope than they would be in the absence of the rules, and where the primary purpose for accepting such limitation is to enable such activity [10].

Traditional teaching methods and learning approaches are no longer appropriate for many kids. According to studies, such approaches impair students' motivation and interest in studying programming. Several serious games or educational games have been created in an attempt to address these issues and promote the learning of computer programming [4], [11]–[13].

II. Python in Education

"Python is an interpreted, object-oriented, high-level, open-source programming language with dynamic sematic" [14]. It is widely used both in science and in R&D by large companies. This programming language is designed to be easy to learn which makes it an extremely good choice for beginners or elementary/high schools.

Python has various qualities that make it a suitable language for programming learning. According to [15], [16] its syntax is straightforward and objective; its dynamic typing determines variables automatically on execution time; and its interpreter offers quick feedback due to potential errors.

Python syntax's compactness allows to dedicate more energy to application and problem solving, significantly reducing development complexity and enhancing development efficiency. Python, being a lightweight-grammar programming language, has well adapted to the modern technology period making it suited for computer programming courses for logical syntax, often similar to English language. It also forces the programmer to write code neatly which is an important habit to learn early on.

Another advantage is versatility of this programming language. Python can be used for a range of tasks from web design to machine learning and is therefore excellent for education. This is because many young people do not know in what area they will be programming. Programming in Python is a great skill even for people who will never work as programmers. This is because of the scripts they can write making their daily work easier.

Open-source projects have huge amount of resources and materials for students and Python is no different. Official sites are written with great care and are often enough to solve most of a basic problems. The size of the Python community is also one of the assets. If a student has a problem, there is always someone who has encountered a similar problem or someone who can give advice.

The aforementioned advantages of Python are far from being all of them. So why mostly other languages are taught in pre-university environment and not Python? According to [17] it is because of Python is an interpreted and not compiled language which affects performance. In interpreted languages the code is easier to test or change, but those languages perform slower compared to their compiled counterparts. But the hardware is getting cheaper and faster on a daily basis and performance, in a smaller, educational scale, ceases to be a problem[18].

The aim of this work was to develop a game to be used for teaching purposes to explain fundamentals of programming in

978-1-6654-8775-7/22/$31.00 ©2022 IEEE

Python to absolute beginners. Programming is a difficult process to understand and many students may struggle with the basic principles, that are being taught in school by theoretical explanations. The game-like environment of this project should help these students to learn those basics with hands-on approach.

Most pupils in the traditional educational system sit and listen to a teacher's oral presentation. However, increasing student enthusiasm in studying through this strategy is frequently tough. During the sessions, most pupils grow preoccupied. To improve their learning motivation, a collection of board games was devised and used in after-class study. Students conducted chats with team members and learned from each other while playing the activities. As a result, pupils learned and taught one another. According to a research, board games can help you learn Python programming [19], [20].

When student learns by playing, he has an additional motivation, not only learning something new but the player also wants to know what new mechanics or puzzles the game introduces next. While playing the game, the player will also receive an instant feedback which can prevent learning bad coding habits and keeps the player captivated.

III. Content of the Educational Game

The main goal of the game is to explain and teach the players fundamental basics of programming in Python. Player can interact from first person view, with the 3D environment where information panels are placed.

The game starts with a basic explanation of **Boolean** and **variables**. The player needs to understand those principles, because the concept of *True* and *False* will be used throughout the whole game and without understanding of variables it is not possible to write a proper code. The second and third obstacle/exercise is about the **conditions** and the **if-else** statements which are used for decision making and directly correlate with the previous challenge. The if-else statement is the basic *control structure* and the player can learn this first step of how to control the program flow.

The fourth exercise is like a summary of the previous exercises and introduction of the **print** command to print messages or variables to a console or any output device.

In fifth exercise, the player will encounter the **list** data structure. It is crucial to understand this part because lists are most common ways to store data multiple times, so players can create basic databases. Lists are also used in last two exercises.

The sixth and seventh exercises are focused on the **for** loop and the **range** command. The loops are one of the most useful tools in programming and it is important to understand how to control them properly so that the loop stops exactly when programmer wants it.

IV. Structure of the Educationl Game

The player can move in the game world freely, however there is only one way through the exercises which means the game is divided in seven parts:

1. Boolean and variables
2. Conditions
3. If-else statement
4. Print
5. Lists
6. Loops
7. For loop /with range

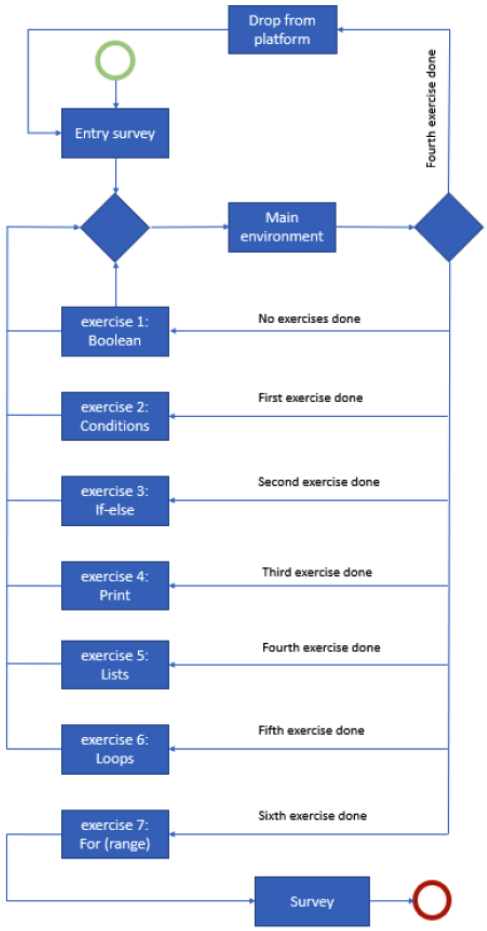


Fig. 1. Structure of the educational game

The game has only one level and at the beginning the player has only one little room to explore; however, the more exercises the player completes the bigger part of the level opens for him.

While designing the map of this game (Fig. 2) we tried to encourage the player to explore the level and reward his exploration by gaining access to more areas and to more abilities, e.g. jumping. After completing the fourth exercise the map shows a new area, but player cannot jump yet and it builds up expectations and desire to continue playing and learning.

To monitor the improvement of the players we created two surveys. The player takes one survey before the game and the other one after completing the game, so we can decide if this

project serves the expected teaching purposes. The whole structure of the game can be seen in Fig. 1.

With every exercise, there is a table explaining the topic to the player. We tried to keep the explanation as brief and clear as possible. We want the player to understand a problem quickly, so that he needs not to read lot of text and can get to the entertaining part as soon as possible. An example of such an information table is in Fig.3 – it is one of the longest in the game.

Though, according to the feedback, some students prefer a more detailed explanation. We tried to cover these needs in the game too, by introducing the advanced help option while solving the exercise. Players can access the advanced help by clicking on the Python symbol (Fig. 4) during the exercise and they will be redirected to *w3school* documentation for the relevant command.

The exercises are in the form of puzzles (Fig. 4), where the player can drag and drop commands/numbers from the box to the code. After the check button is pressed, the code is evaluated by the Python interpreter. If the code works as intended, the exercise is considered complete and the level opens a new path for exploration. If the compilation of the exercise fails, the player has to try again.

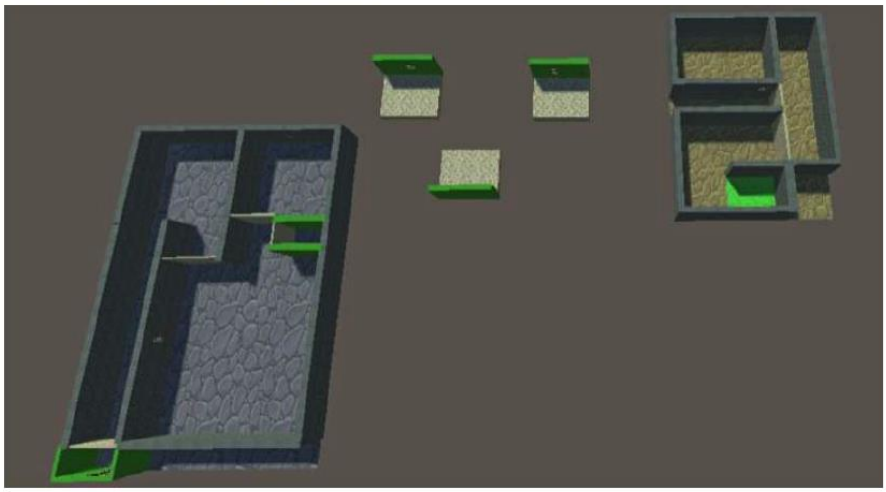


Fig. 2. Map of the level

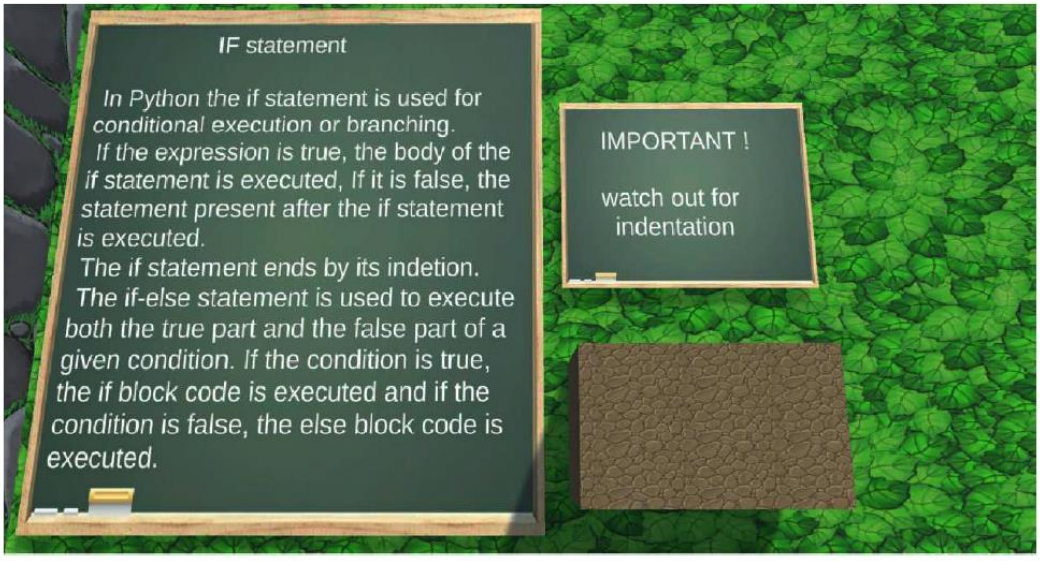


Fig. 3. Information table

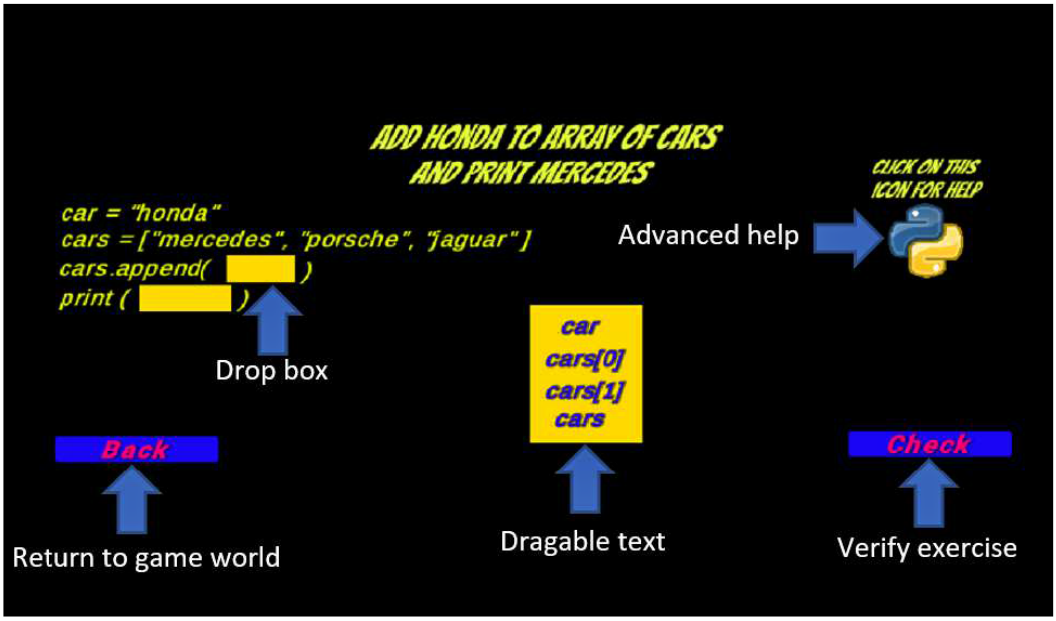


Fig. 4. Exercise example

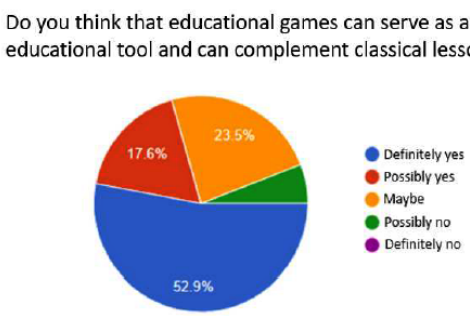


Fig. 5. Opinion of players on games in education before playing our game

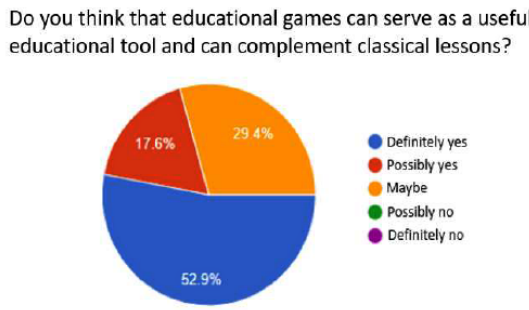


Fig. 6. Opinion of players on games in education after playing our game

V. FEEDBACK AND ANALYSIS

We tested our game directly in schools. Most of our participants were from high schools (70.06%), the others were from a university (29.04%). We were not able to fund any elementary school that would be willing to participate in the survey. The aim of the survey was to identify the skills of the participants before and after the playthrough by proposing some basic questions about programming in Python.

We also wanted to know how the players feel about educational games before and after their experience with our application. A comparison of users' opinion before and after they played the game is shown in Fig. 5 and Fig. 6. We managed to improve their opinion a little bit.

Another question was about programming skills (Fig.7 and Fig.8) and skills with Python (Fig. 9 and Fig. 10). Evidently, around a quarter of all players have evaluated that they improved their skills.

Next, we analyzed improvement of the participants in specific commands. In Fig. 11 and Fig 12 we can see assessment of improvements. Before the game, 35.3% of participants knew the right answer and after the game it was 68.8%.

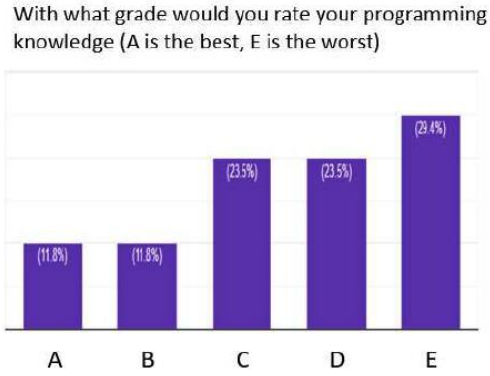


Fig. 7. Programming skill before the game

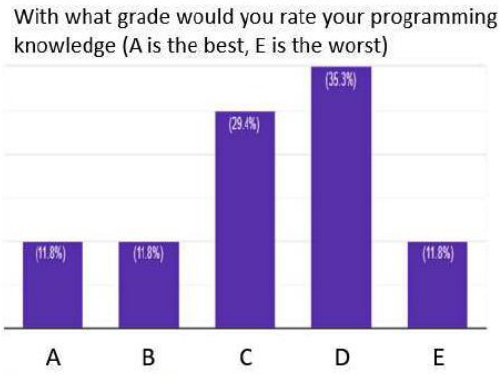


Fig. 8. Programming skill after the game

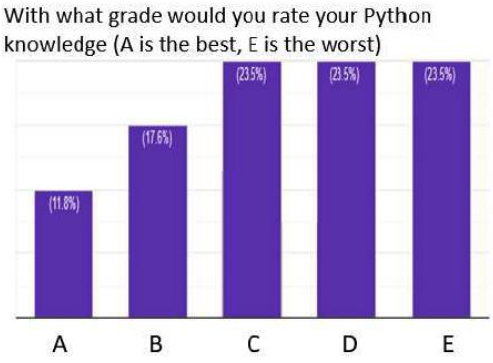


Fig. 9. Python knowledge before the game

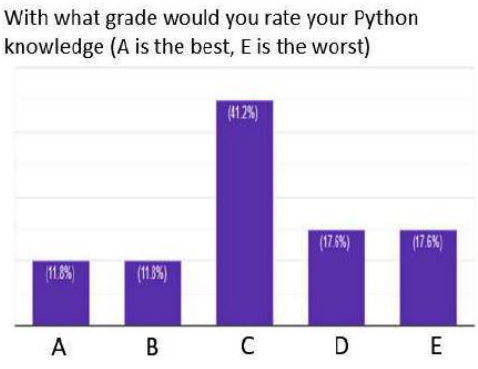


Fig. 10. Python knowledge after the game

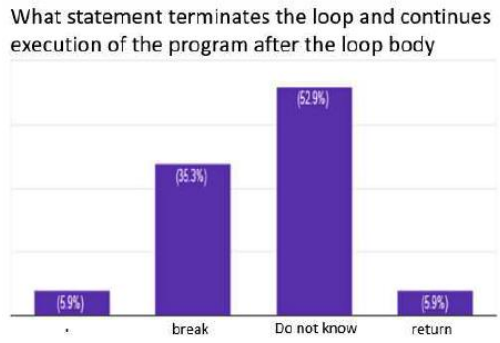


Fig. 11. Basic command knowledge before the game

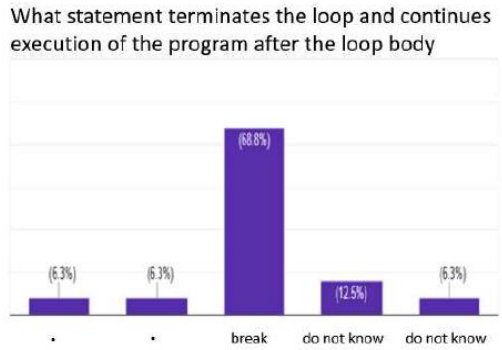


Fig. 12. Basic command knowledge after the game

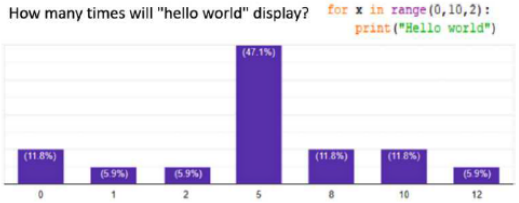


Fig. 13. Knowledge of range and for cycle

Fig. 13 shows the assessment of the answers to the last question, which was only presented in the exit survey, because we had many participants that never used any programming language, and we found this question too hard for such people. Only 47.1% answered correctly, it may be because the information needed to answer this question correctly was given as text table and students have not tried it on their own.

VI. Conclusion

The aim of this paper was to introduce interactive gaming environment for education to younger students and make it easier for them to learn Python. The Python is a great tool, and students are likely to encounter it in their professional or academic life. We created entry and exit surveys to help us analyze and evaluate our application. Based on our analysis, we can call our project a success. Players who played the game reported improvement in their programming skills and Python knowledge. Based on feedback we could improve graphics and make more use of special powers to make game more fun. Players also asked for mor content so in future, we can add more levels and explain another important subject related to programming like functions or object-oriented programming.

Acknowledgement

The paper was supported by the SRDA grant APVV-2021-0121 and by the Scientific Grant Agency of the Ministry of Education, Science, Research and Sport of the Slovak Republic under the grant KEGA 016STU- /2020 . We would also like to thank to Cicka for helping with programming the implementation.

References

[1] K. Ofosu-Ampong, "The Shift to Gamification in Education: A Review on Dominant Issues," *J. Educ. Technol. Syst.*, vol. 49, no. 1, 2020, doi: 10.1177/0047239520917629.

[2] M. Baeten, K. Struyven, and F. Dochy, "Student-centred teaching methods: Can they optimise students' approaches to learning in professional higher education?," *Stud. Educ. Eval.*, vol. 39, no. 1, 2013, doi: 10.1016/j.stueduc.2012.11.001.

[3] E. Kucera, O. Haffner, and R. Leskovsky, "Interactive and virtual/mixed reality applications for mechatronics education developed in unity engine," in *Proceedings of the 29th International Conference on Cybernetics and Informatics, K and I 2018*, Apr. 2018, vol. 2018-January, pp. 1–5. doi: 10.1109/CYBERI.2018.8337533.

[4] E. Kučera, O. Haffner, R. Leskovský, J. Matišák, and P. Drahoš, "The Use of Modern Multimedia Trends for Popularization of Mechatronics by Presenting Its Present and Future Application," in *Communication Papers of the 2019 Federated Conference on Computer Science and Information Systems*, Sep. 2019, vol. 20, pp. 49–54. doi: 10.15439/2019f182.

[5] M. H. Rapporteur, *Exploring the Intersection of Science Education and 21st Century Skills: A Workshop Summary*. 2010.

[6] R. N. Landers, K. N. Bauer, R. C. Callan, and M. B. Armstrong, "Psychological theory and the gamification of learning," in *Gamification in Education and Business*, 2015. doi: 10.1007/978-3-319-10208-5_9.

[7] M. Kalogiannakis, S. Papadakis, and A. I. Zourmpakis, "Gamification in science education. A systematic review of the literature," *Educ. Sci.*, vol. 11, no. 1, 2021, doi: 10.3390/educsci11010022.

[8] M. Huba and Š. Kozák, "From e-Learning to Industry 4.0," in *ICETA 2016 - 14th IEEE International Conference on Emerging eLearning Technologies and Applications, Proceedings*, Dec. 2016, pp. 103–108. doi: 10.1109/ICETA.2016.7802083.

[9] S. Kozak, E. Ruzicky, J. Stefanovic, and F. Schindler, "Research and education for industry 4.0: Present development," in *Proceedings of the 29th International Conference on Cybernetics and Informatics, K and I 2018*, Apr. 2018, vol. 2018-January, pp. 1–8. doi: 10.1109/CYBERI.2018.8337556.

[10] S. Kim, K. Song, B. Lockee, and J. Burton, *Gamification in Learning and Education*. 2018. doi: 10.1007/978-3-319-47283-6.

[11] S. Papadakis and M. Kalogiannakis, "Using gamification for supporting an introductory programming course. The case of classcraft in a secondary education classroom," in *Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering, LNICST*, 2018, vol. 229. doi: 10.1007/978-3-319-76908-0_35.

[12] M. A. Miljanovic and J. S. Bradbury, "A review of serious games for programming," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2018, vol. 11243 LNCS. doi: 10.1007/978-3-030-02762-9_21.

[13] C. Kroustalli and S. Xinogalos, "Studying the effects of teaching programming to lower secondary school students with a serious game: a case study with Python and CodeCombat," *Educ. Inf. Technol.*, vol. 26, no. 5, 2021, doi: 10.1007/s10639-021-10596-y.

[14] Python Foundation, "About Python™ | Python.org," *About Python*, 2016.

[15] L. Grandell, M. Peltomäki, R. J. Back, and T. Salakoski, "Why complicate things? Introducing programming in high school using Python," *Conf. Res. Pract. Inf. Technol. Ser.*, vol. 52, 2006.

[16] Z. Xu, X. Shen, S. Lin, and F. Zhang, "Using Visualization to Teach an Introductory Programming Course with Python," in *Proceedings - 11th International Conference on Information Technology in Medicine and Education, ITME 2021*, 2021, pp. 514–518. doi: 10.1109/ITME53901.2021.00109.

[17] F. Georgatos, "How applicable is Python as first computer language for teaching programming in a pre-university educational environment, from a teacher's point of view?," Sep. 2008, [Online]. Available: http://arxiv.org/abs/0809.1437

[18] M. Pajpach, O. Haffner, E. Kučera, and P. Drahoš, "Low-Cost Education Kit for Teaching Basic Skills for Industry 4.0 Using Deep-Learning in Quality Control Tasks," *Electron.*, vol. 11, no. 2, Jan. 2022, doi: 10.3390/electronics11020230.

[19] L. W. Huang, L. W. Chen, and P. H. Cheng, "Board Game Design for Python Programming Education," 2019. doi: 10.1109/TALE48000.2019.9225988.

[20] O. Haffner, E. Kucera, D. Rosinova, A. Kozakova, P. Drahos, and M. Kocur, "Multimedia support for education of mechatronics," in *Proceedings of the 29th International Conference on Cybernetics and Informatics, K and I 2018*, Apr. 2018, vol. 2018-January, pp. 1–5. doi: 10.1109/CYBERI.2018.8337537.

Authors



Ing. Michal Hlavatý
Institute of Automotive Mechatronics,
Faculty of Electrical Engineering and Information Technology,
Slovak University of Technology in Bratislava, Slovakia
michal.hlavatý@stuba.sk
He completed his master's degree in the field of automotive mechatronics in 2017. He is currently studying for the third year of doctoral studies dedicated to artificial intelligence and neural networks with a focus on reinforcement learning.



prof. Ing. Alena Kozáková, PhD.
Institute of Automotive Mechatronics,
Faculty of Electrical Engineering and Information Technology,
Slovak University of Technology in Bratislava, Slovakia
alena.kozakova@stuba.sk
From 1985 to 2013 she was with the Department of Automatic Control Systems (later Institute of Control and Industrial Informatics). From 2018 she is a full professor at the Institute of Automotive Mechatronics, Faculty of Electrical Engineering and Information Technology, STU in Bratislava. Her research interests include robust, optimal and decentralized control.



doc. Ing. Oto Haffner, PhD.
Institute of Automotive Mechatronics,
Faculty of Electrical Engineering and Information Technology,
Slovak University of Technology in Bratislava, Slovakia
oto.haffner@stuba.sk
Research activities in applications of computer and machine vision in mechatronic systems using modern ICT such as microcomputers, cloud, computational intelligence, etc. Current focus also on the use of deep learning in practical applications for Industry 4.0 such as visual quality control. He is also involved in modelling and simulation of production processes in manufacturing.