

MakeCode Arcade Platform for Game-Based Learning

Patrik Voštinár

Department of Computer Science, Tajovského 40
974 01 Banská Bystrica, Slovakia
patrik.vostinar@umb.sk

Abstract. Currently, there is an increasing demand for specialists in the field of computer science. However, for many children computer science is not as interesting as it could be. They learn many topics in a very short time and usually do not have the motivation to continue their studies in Computer Science. The solution could be teaching through games. In this paper, we describe our research into whether the MakeCode Arcade environment is suitable for teaching programming through games (game-based learning platform). In the research, we compare the responses of 315 male and female students and the responses of elementary and secondary school students. The research showed that boys liked the MakeCode environment more than girls and also that programming games in this environment were more meaningful to them as the girls. In this contribution, we describe also our experiences with using this environment for teaching in extracurricular activity at our university, in normal lessons in primary schools, as a motivational tool for distance workshops and also for workshops for teachers.

Keywords: MakeCode Arcade, children education, game-based learning, teaching programming.

1. Introduction

For the successful structuring of students' educational processes, the system of education necessitates the employment of new teaching technology. In today's digital era, smart education is a form of education which is becoming increasingly generally recognized and enthusiastically embraced adopted by younger generations [1].

We are currently facing technology all around us. More and more programmers are in demand to acquire, develop, and work with these technologies. Therefore, it is important that our education system produces more students who are able to program and think algorithmically. One possibility could be to use various types of computer games to motivate and also teach algorithm thinking.

Computational thinking (CT) has been actively promoted in schools in an integrative way or as a part of computer science subject, as it addresses concepts and learning goals of computer science. We may notice that interest in teaching computer science in primary school has increased during the last decade [2]. As recommended for computer science education in the report by the Committee on European Computing Education, "All students must have access to ongoing education in computer science in the school system. Computer science teaching should preferably start in primary school. . ." [3, p. 3]. Dagienė et al. [4] did research, about the worldwide tendencies of teaching CT through computing in primary education and primary school. The results of the quantitative study had shown

that CT and concepts related to computer science subject are introduced in the majority of surveyed countries (83%) in primary education. However, there are a lot of differences in the level of its implementation.

Computer games have a huge impact on a child's development. Actually, recent educational neuroscience researches show that the best way to teach children is through playing, getting their attention, their engagement, receiving feedback and consolidating their skills [5].

A special category are games that could also be used in the educational process in schools. It is important to select appropriate games that allow them to develop their personality. Examples of such games are Roblox and Minecraft Education. The authors of *Minecraft Research: A Game as an Education and Scientific Learning Tool* [6] examines how Minecraft can help teach different areas. They state that the diversity of biomes and terrain in Minecraft's world can help students understand ecology and geology. They say that the behavior of objects in Minecraft is similar to that in the real world. For example, trees need light, water, and space to grow, or sugar cane must live close to water [7]. Redstone logic uses the principles of logic gates (OR, AND, NOR, and XOR). For example, this simple logic allowed players to create a working computer in the game that could perform simple mathematical operations. The use of Redston is very easy for new players and it can be used to explain the basics of electrical engineering and logic circuits. With *Minecraft: Education Edition*, it is also possible to teach mathematics, chemistry, biology, history, art and design, English, and programming. Han, Liu, and Gao [8] focused in their research on the positive influence of using Roblox during the teaching process.

2. Gamification, Game-Based Learning and Serious games

There exist special terms when computer games are used in the educational process: gamification, game-based learning (GBL) education, and serious gameplay. The term gamification is generally used to denote the application of game mechanisms in non-gaming environments, such as earning points, levels, or badges. Gamification aims to transform systems, activities, organizations, and services to a system that uses the characteristics of game elements [9]. Gamification is a well known technique in education, organization engagement, crowdsourcing, commerce, information retrieval, and so on [10,11]. Gamification appears to be an emerging trend in many sectors: health, social policy, business, education [12]. Garcia-Iruela and Hijón-Neira in their study [13] focused on the perception that students have of the different elements of gamification in three different experiences, after which students were asked to answer a survey about the gamified elements of the course in which they participated.

Game-based learning is to present the instruction by games in learning, with the main purpose of triggering learners' motives instead of instructing the courses. GBL has been used to teach programming, any other topic, in a way that engages learners [14-16]. Husain et al. [17] did research focusing on digital game-based learning methods to improve students' critical thinking skills in elementary science. The outcome of their study may help researchers, teachers, and practitioners in the educational sector to evaluate the effectiveness of serious games in facilitating students' critical thinking. Many researchers show that digital game-based learning is becoming a powerful tool in education, making learning more enjoyable, easier and efficient [18,19].

Edutainment means educating children using methods and tools that are traditionally associated with entertainment. Examples of this type of education are movies, television shows, and game consoles. Edutainment was popular during the 1990s with its growing multimedia PC market [20].

Serious games encompass the same goals as education but extend far beyond teaching facts and rote memorization, and instead include all aspects of education, teaching, training, and informing, and for all ages. Serious games can be applied to a broad spectrum of application areas, e.g. military, government, educational, corporate, and healthcare [21]. Christinaki et al. [22] did research, that serious games can be very effective in the areas of therapy and education for children with autism. Alamri et al. [23] created and used a serious game focused on problem of obesity.

Many other researchers are not only using Roblox and Minecraft, but they teach programming through many other games or game environments. Jesus and Silveira in their research [22] used the collaborative learning concept in secondary school students through a digital games programming workshop. Many types of research [24-29] show that teaching programming games for students could be motivating for students. It is also very good for the imagination, for developing children's computational thinking.

Another interesting environment for game-based learning is the MakeCode Arcade environment. Siever and Rogers conducted research [30] about this interesting environment, which can create games, that can be deployed to lowcost consoles, comparable to Nintendo's Gameboy. The camp used MakeCode Arcade as the programming language while still focusing on game development. Brown et al. [31] conducted a survey of interface representations in visual programming language environments for children's physical computing kits that shows that MakeCode Arcade is a suitable environment for teaching programming. Begel et al. [32] developed instructional materials to teach computer programming, video game design, and communication and teaming skills through the MakeCode Arcade environment. Students used the MakeCode Arcade development environment to build their games and Zoom to remotely collaborate with their teammates. In summative interviews, students reported improved programming skills, increased confidence in communication, and better experiences working with others. They found that students valued the opportunity to practice teamwork, such as being more vocal in expressing ideas to their peers and working out differences of opinion with their teammates. We described our pilot research and experience with MakeCode Arcade in conference Edunine [33].

3. MakeCode environment for teaching programmings

The Microsoft company started developing the MakeCode Arcade environment in 2018. The first beta version appeared in 2019 and the official version started in 2020. Microsoft MakeCode is an online web environment that offers visual programming language, JavaScript, and Python language to program educational board micro:bit, robots Lego EV3 and Cue, and the game Minecraft: Educational Edition, MakeCode Arcade, and Adafruit Circuit Playground. The environment is available at <https://arcade.makecode.com/>

MakeCode is not a universal programming platform, but an extensible framework to create any number of MakeCode based experiences customized to a specific target. These

targets are typically makerstyle or educational singleboard computers, but could also be a simulated only 'soft' target.

The concept of each device, that provides the possibility to program in the MakeCode environment, is the same - it offers many tutorials on the main page, the essentials of a MakeCode editor consist of switching buttons between programming languages, an interactive simulator, target code button, and blocks/language editor. The only difference is in the visual site (different colors for each device).

Fig. 1 shows the MakeCode Arcade page, which contains the simulator on the left side, under the simulator is the target code button (after pressing this button a popup window will appear with a list of available hardware devices, consoles), next to the simulator is a palette of blocks, and on the right side is the editor of visual code programming.

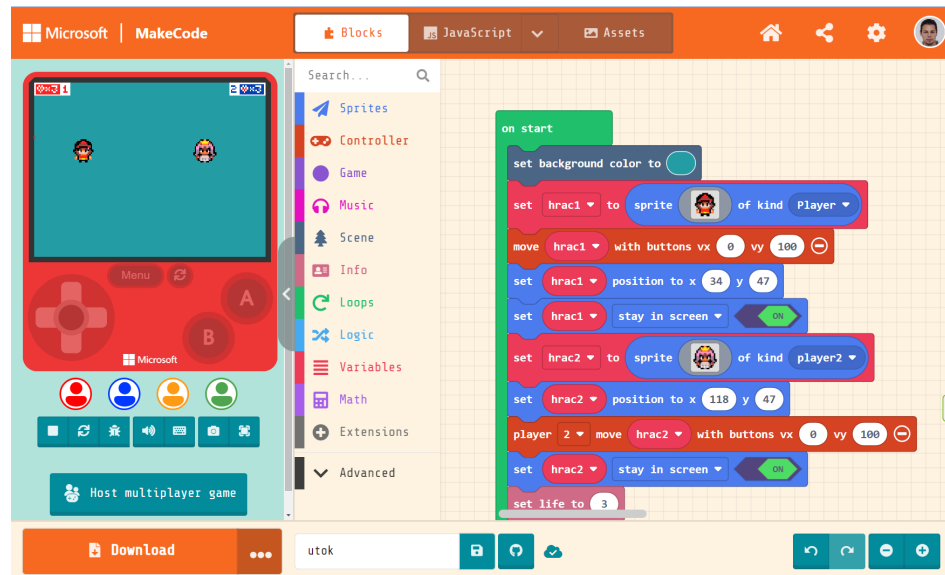


Fig. 1. The MakeCode Arcade environment

The advantage of this environment is the ability to test the created 2D games in the simulator, on a special game console (approx. 35 €), and on mobile devices. The interactive simulator provides students immediate feedback on how their program is running and makes it easy to test and debug your code. The simulator can be controlled with its buttons, with the computer keyboard keys W, S, A, D and space, or with a gamepad connected to the computer (e.g. Xbox gamepad). After pressing the download button, located under the simulator, it is possible to choose from many types of game consoles intended for MakeCode Arcade. Fig. 2 shows multiple game consoles for MakeCode Arcade games.

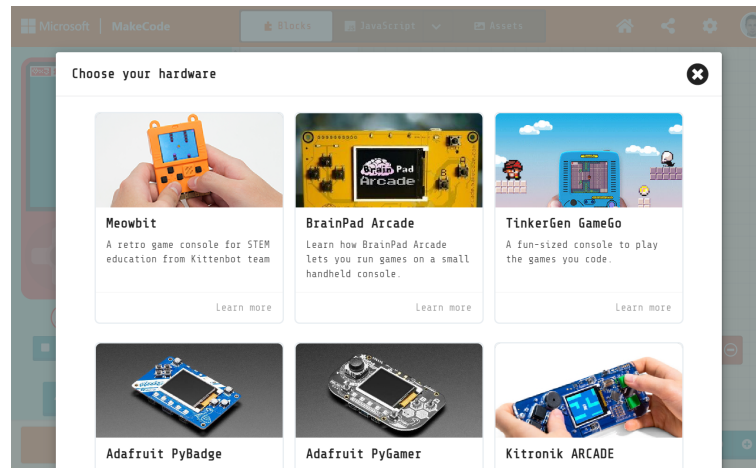


Fig. 2. The MakeCode Arcade – a hardware for testing

The third option is to test the game on a mobile device by pressing the share button (on the orange top panel, the third button). It will create a unique URL, which can be copied to the mobile browser (through QR or just through the URL link). A screen with a game console (buttons, display) will appear on the mobile device. This screen looks and behaves exactly like a game console.

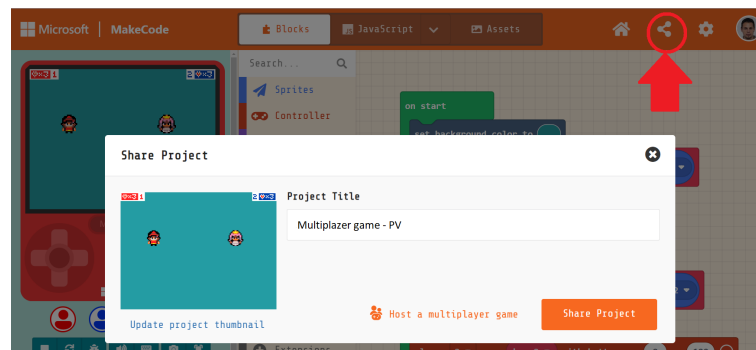


Fig. 3. The MakeCode Arcade – sharing a project for mobile devices

The MakeCode Arcade environment contains many instructions for creating different types of games, but also for teaching programming. Compared to the Scratch environment, MakeCode Arcade focuses more on game principles (velocity, status bar or health bar of sprites, working with scenes, connection to physical game consoles) than Scratch. Scratch is a more universal environment. Compared to other game development environments, this environment is best suited for primary school teaching (it uses block programming that students are used to). Another advantage is that we don't have to log in or

install to computers. Other environments for teaching game programming, e.g. Unity, Unreal Engine, Godot Engine require a more advanced level of programming (more suitable for secondary schools or universities). The environments eBlock, Scratux, Code Kit are based on the Scratch environment and, like Scratch, are more universal environments for teaching programming than for programming games. Another advantage of MakeCode Arcade is that Microsoft created a curriculum, a full academic year course designed to be taught for introductory programming classes in secondary schools. The curriculum was adapted as a core teaching platform using a game-based learning approach to engage a diversity of students. The original TEALS course has been successfully implemented in hundreds of secondary schools. The course is divided into a series of units which successively introduces new concepts and reinforces those learnt previously. The course starts with teaching programming using visual drag-and-drop blocks and then JavaScript or Python. During the course, students learn basic principles of games, programming concepts, creating various types of biomes, working with coordinates, and advanced graphics techniques. Students work in teams (collaboration). The course is available on the official website <https://arcade.makecode.com/courses/intros>.

3.1. Examples of educational games

On the official website of MakeCode Arcade, there are several games that we can use for educational purposes. Creating educational games in this environment is quite easy, so it is possible to create different types of educational games.

Educational games - we can create different educational games for geography (different biomes), create history (prehistoric period - etc.), math coordinates/examples, practice language learning, observation, plays focused on significant works of literature (e.g. Hamlet), etc. We can also create different games for multiplayer cooperation (competition for button pressing speed, who gets clovers first, etc.). Games aimed at explaining game design concepts, explaining the principles of various graphic algorithms, fibonacci tiles, arts, etc. Fig. 4 shows existing games for game design concepts and graphic (Setting the Scene, Tale of Talagron, Fibonacci Tiles, etc.).

We can run games created on the MakeCode Arcade platform on a computer, mobile device, specialised game consoles, or we can create our own game console according to the instructions (basics of electrical engineering). Fig. 5 shows tutorials for DIY (Do it yourself) hardware. In the Shoebox Controller tutorials, you can make a game console from a shoebox, Arcade cabinet - make a game console from wood, etc.

3.2. Programming games

If the created games are not enough for us, we can simply create our own game. Creating games is very easy even for non-programmers. We have tutorials for each type of game. We can create games in blocks, JavaScript or Python.

We will explain the principle of creating a simple game for two players in a game where two players will throw fruit at each other. Figure 6 shows the code blocks for creating two characters. The picture shows a code block without the principles of object-oriented programming. A more suitable version is if we divide the creation of characters into several functions (one function is for one character). The students themselves will understand that object-oriented programming is easier to read the code.

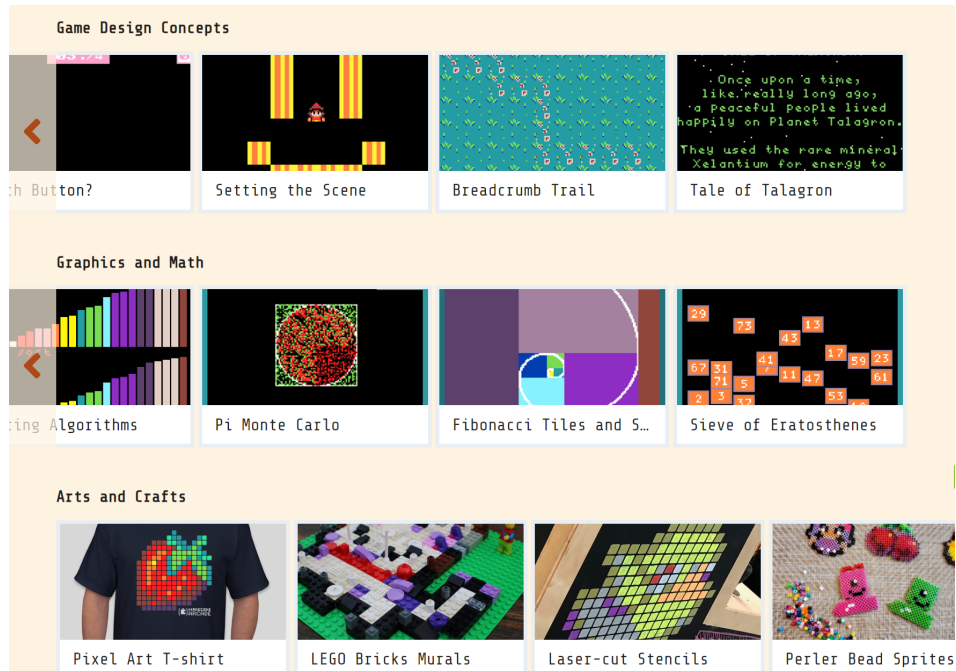


Fig. 4. Games for game design concepts and graphic

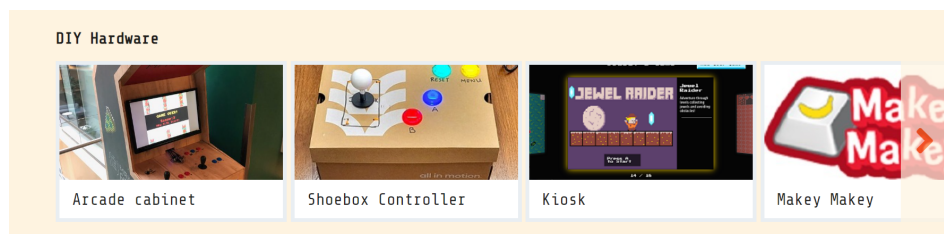


Fig. 5. DIY hardware

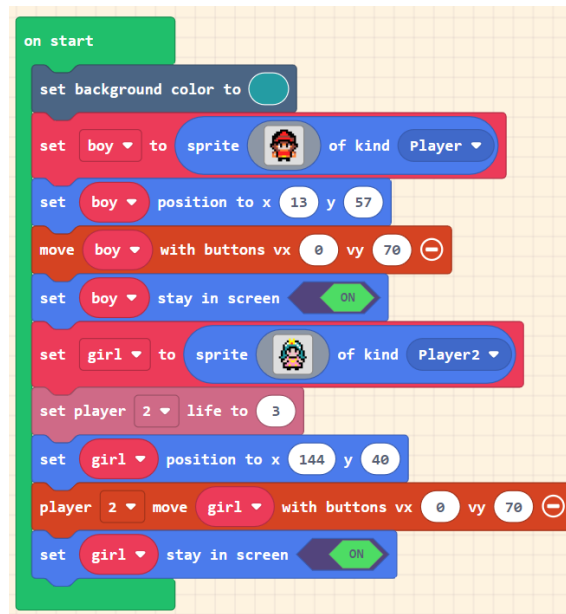


Fig. 6. Blocks code - creating two characters

Figure 7 shows a block of code that, after pressing the A button creates a projectile, which will appear on player 1 (boy) and has the velocity x (vx set to 100) and velocity y (vy set to 0), projectile will fly horizontally with speed 100 pixels per second.

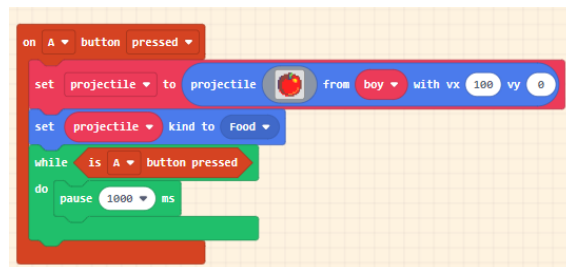


Fig. 7. Blocks code - events

The last step is to create behavior, what will happen when Player 1 (boy) overlaps the projectile Food2 (which was thrown by Player 2-girl) and the same when Player 2 (girl) overlaps the projectile Food (which was thrown by Player 1-boy). After a player overlaps one of the projectiles, we will destroy the projectile and increase the score of the player who touched it. Figure 8 shows objects overlapping behaviour.

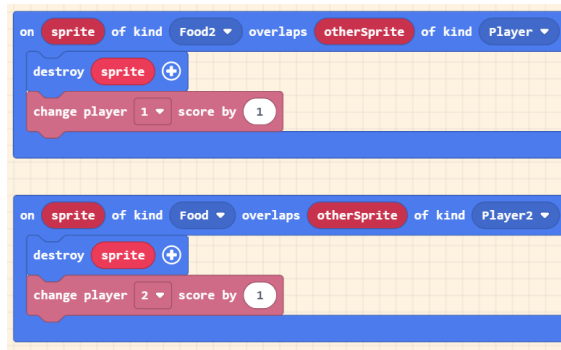


Fig. 8. Blocks code - overlapping behaviour

In this step, it is very important to remove the projectile from the game. If we do not remove it, it will still exist in the memory of the game device, and after a certain time the game console (or computer) will start to freeze.

4. Teaching through the MakeCode Arcade environment

We started using the MakeCode Arcade environment for teaching in an extracurricular activity for primary schools, which we organized in our department at the university. After testing the environment in the extracurricular activity, we started teaching with this environment in primary school in Banská Bystrica, secondary school in Žiar nad Hronom and Zvolen and later then on special online activity propagating computer science called IT tea room. After that we also showed this environment in an activity called The club of computer science teachers in Banská Bystrica (training for teachers).

4.1. Computer science extracurricular activity

In our Department of Computer Science FNS UMB, we are organizing free extracurricular activities for primary school students since the academic year 2017/2018. The activity is organized twice a week for 1,5 hours. The students of the Teaching trainee of the Computer Science program and Applied Computer Science program are involved as lecturers or tutors in this activity. Each of our students teaches 1-2 children from primary school. The main focus of this activity is to obtain additional teaching experience for our students and increase motivation to study computer science (programming). Each year we have approximately 10-13 for each group (one group is on Wednesday and another group on Thursday), so together 20-26 students.

In our extracurricular activity, we are using many types of educational aids such as micro:bit, Ozobot, mBot, lego EV3, programming drones, games, webpages, etc. We also use many types of tasks. Since February 2020, schools have been closed due to the Covid-19 pandemic. We had to switch to online teaching through Microsoft Teams, and we decided to use the online environment MakeCode Arcade for teaching game programming, mainly because students do not need to install any hardware on their computers.

First we started to program according to tutorials from official webpage:

- Chase the Pizza
- Happy flower
- Lemon Leak
- Galga
- Barrel Dodger
- Maze

The tutorials were quite easy for the students who had previous experience programming many educational robots. After each tutorial was finished, we started improvising our lectures after each game we asked students to brainstorm on how we could improve. We also added to our tutorials some additional enemies, some items that cannot be touched and some items (for example life) that appear only a few seconds in random positions, after some score we added boss (alien) enemy, which have to be shot more times, etc.

After that we started to create our own story for games, a multiplayer game, and many other games.

students were happy with the classes, visual block programming was nothing new for them. The observation shows that this environment is suitable for teaching programming and also for developing your creativity. We also sent them an anonymous questionnaire after we finished coding in this environment - we used this environment 6 months. Fig. 9 shows teaching the extracurricular activity.

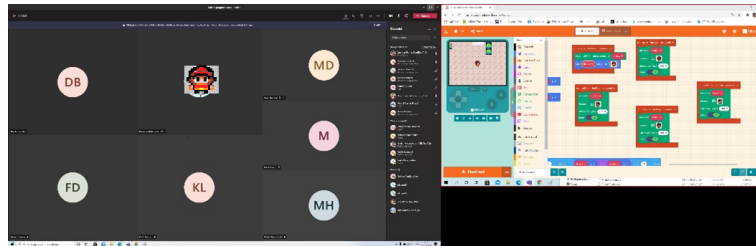


Fig. 9. The extracurricular activity – online teaching

4.2. Primary school

After our experience with programming in the environment MakeCode Arcade in our extracurricular activities, we decided to use it in the normal teaching hours of the primary school Ďumbierska 17 in Banská Bystrica. We started teaching with this environment from April 2020 in 6 classes (13-15 years old children, 45 minutes for each class weekly). During three months of online teaching in this environment we can say, based on our observations, that the students liked this environment because it was something different for them than programming in Scratch. For them, the previous experience with subject Computer Science was mainly programming in Scratch. Since the academic year 2019/2020, they started using educational board micro:bit for teaching of programming, so they were familiar with the MakeCode environment.

For these lessons, we had selected the most interesting games from the extracurricular activity (Chase the Pizza, Lemon Leak, Galga, Maze, Multiplayer game, Shark attacking,

Simon says, etc.). After they familiarized with MakeCode Arcade, they had to work in 4 person groups to create their own game. They worked in MS Teams breakout rooms, so during the lessons we could check their progress. Most of the groups also worked in their spare time and created interesting games.

In the end, the teams had to present their games to their classmates. After that they were evaluated.

4.3. Secondary schools

As part of the special project called Harpuna (the project focuses on increasing children's soft skills in the Computer Science subject), we also had the opportunity to teach in two secondary schools, where we taught computer science during the academic year 2021/2022 for 6 months. We had two classes available at the Milan Rúfus Secondary School in Žiar nad Hronom and one class at the Ľudovít Štúr Secondary School in Zvolen. Both schools are eight-year secondary schools that prepare students from 11 to 18 years of age. In both schools, we taught programming using micro-bit educational boards, game programming in MakeCode Arcade, and mobile application programming in App Inventor (each topic for about 2 months). At the beginning of the lesson, we showed the environment and some examples of the application. Then the students worked in groups of 4 on their own projects. During the lessons, we continuously helped them if necessary.

4.4. “IT čajovňa” and “Veda v meste” (“IT teahouse” and “Science in the city”)

In January 2021 the Gymnázium Sama Chalupku in Brezno requested that our department teaches a motivational online workshop for computer science. At this time, we were also requested by the primary school Trieda SNP in Banská Bystrica to teach the activity called “Veda v meste” (“Science in the city”).

In both schools, we conducted two-hour motivational workshops for children 14-15 years of age with topic programming games in the MakeCode Arcade. In Brezno we had 30 children and in Trieda SNP we had 29 children. We showed them simple games similar to Space Invaders. Based on comments from children during the workshops and after the workshops from their teachers, the students were excited. In February 2021, we were approached by the national IT Academy project (activity called “IT čajovňa” – “IT teahouse”) to do IT workshops/lectures for primary and secondary school students. As part of these workshops, we conducted a series of four workshops for primary school students and a series of two workshops for secondary school students, focusing on game programming in the MakeCode Arcade environment. The workshops lasted 2 to 2.5 hours and they were recorded so that the students who could not attend, could watch them afterwards. The difference between the workshop content for primary and secondary school was in the level of difficulty: in the workshops for primary school we programmed simpler games than in the workshop for secondary school where we programmed more advanced games with animations. A total of 47 secondary school students and 126 primary school students participated in the workshops. There were more different topics available for secondary schools, so fewer students participated than in workshops from primary schools. From the organizers, we received continuous feedback in comments and after the workshops. We also asked the students to fill out a questionnaire on how they liked the workshop. The

students were also interested in the workshops because we held six different workshops where they programmed six different games.

Fig. 10 shows the online workshops in the MakeCode environment.

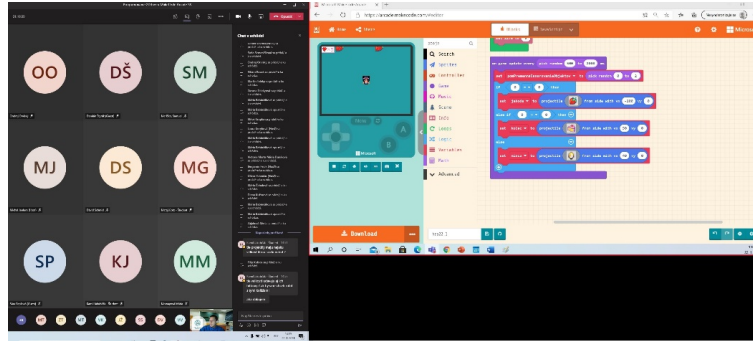


Fig. 10. The online workshops teaching

4.5. Club of computer science teachers

Our Department of Computer Science, FNS, UMB started organizing in January 2020 a training with new technologies for teachers. In this activity, we focus on new popular technologies, ICT devices and innovative software, to teach computer science. Based on previous positive feedback from the MakeCode Arcade, we decided to show teachers this environment, which is suitable for presence and distance learning.

In November 2020 we organized the first four hours of the online training for primary school teachers in the Banská Bystrica area. We showed them the environment and together we programmed three simple games. After the training, we asked them how did they like this environment and if they want to have another training on the same topic, but with new games.

Based on the positive feedback from the questionnaire, we organized in March 2021 another training with MakeCode Arcade with new games. Both workshops were attended by 10 teachers. The teachers were very enthusiastic about the MakeCode Arcade platform. They liked that they could use it to explain the principles of programming, graphics, and also that they could very easily create their own educational games that they could use in teaching various subjects.

5. Methodology and Research

We started our research in the academic year 2020/21. In our research, we used the MakeCode Arcade environment for learning programming in our extracurricular activity at our university during the Covid-19 pandemic, in classic primary school lectures and as a topic in motivational workshops. We also verified this environment by discussing the advantages and disadvantages of this environment with primary school teachers.

We used a questionnaire to measure the data of our research. 315 children - 178 boys and 137 girls, filled out the anonymous questionnaire using Google Forms. The questionnaire consisted of 12 questions. The first three questions focused on gender, the name of the school, and the grade of the pupil visiting the school at the time when filling out the questionnaire. The questionnaire was filled out by 176 students from secondary schools and 139 students from primary schools. All students first completed the same introduction to the MakeCode Arcade environment. Secondary school students programmed more challenging games than primary school students in the same environment (we even tried Python with some groups). A questionnaire of our design was used to obtain empirical data, which were processed in Microsoft Excel using descriptive statistics methods. In evaluating the responses to the chosen items, we utilized the Chi-Square test.

In our research, we set several hypotheses, which we investigated, e.g., the differences between the answers of boys and girls and the differences between the answers of students from primary and secondary schools.

The first question that interested us was whether they had ever programmed games (e.g. in Scratch). 202 students (64.1%) answered that they did and 113 students (35.9%) answered that they had never programmed any games. In the next question, we asked if they had ever programmed games in the MakeCode Arcade environment - 297 students (94.3%) answered that they had never programmed in this environment. Only 18 students answered that they had already tried this environment.

Using the questionnaire item, *Were these activities meaningful to you?* we investigated the difference between the answers of boys and girls, students from primary and secondary schools. Table 1 shows the hypotheses for this questionnaire item.

Table 1. Hypotheses for the questionnaire item *Were these activities meaningful to you?*

H_0	H_1	Probability of error	The null hypothesis
There are no significant differences in boys' and girls' responses about whether they thought the activities with MakeCode Arcade were meaningful.	There are significant differences in boys' and girls' responses about whether they thought the activities with MakeCode Arcade were meaningful.	0,62%	reject
There are no significant differences in the responses of primary and secondary school students as to whether they thought the activities with MakeCode Arcade were meaningful.	There are significant differences in the responses of primary and secondary school students as to whether they thought the MakeCode Arcade activities were meaningful.	24,62%	accept

Based on the arithmetic mean, boys think that the activities in the MakeCode Arcade environment were more meaningful to them than girls. There are no differences in the answers of students from primary and secondary schools to the question whether the activities were meaningful to them. The answers were positive, both groups think that these activities were meaningful for them.

Using the questionnaire item, *Did you like programming games in MakeCode Arcade?* we investigated the difference between the answers of boys and girls, students from primary and secondary schools. Table 2 shows the hypotheses for this questionnaire item.

Table 2. Hypotheses for the questionnaire item *Did you like programming games in MakeCode Arcade?*

H_0	H_1	Probability of error	The null hypothesis
There are no significant differences in the answers of boys and girls about whether they liked programming in MakeCode Arcade.	There are significant differences in the answers of boys and girls about whether they liked programming in MakeCode Arcade.	0,07%	reject
There are no significant differences in the answers of students from primary and secondary schools, whether they liked programming in MakeCode Arcade	There are significant differences in the answers of students from primary and secondary schools, whether they liked programming in MakeCode Arcade	38,52%	accept

Based on the arithmetic mean, boys liked game programming more than girls. There are no differences in the answers of students from primary and secondary schools to this question. The answers were positive, both groups like programming in this environment.

Using the questionnaire item, *Were the activities difficult for you?* we investigated the difference between the answers of boys and girls, students from primary and secondary schools. Table 3 shows the hypotheses for this questionnaire item.

Table 3. Hypotheses for the questionnaire item *Were the activities difficult for you?*

H_0	H_1	Probability of error	The null hypothesis
There are no significant differences in the answers of boys and girls, whether programming in MakeCode Arcade was difficult.	There are significant differences in the answers of boys and girls about whether programming in MakeCode Arcade was difficult.	42,09%	accept
There are no significant differences in the responses of students from primary and secondary schools as to whether programming in MakeCode Arcade was difficult.	There are significant differences in the answers of students from primary and secondary schools, whether programming in MakeCode Arcade was difficult.	63,89%	accept

There were no differences in the responses of primary and secondary school students, boys and girls, as to whether the activities were difficult for them. The answers showed that the activities were easy for them.

Using the questionnaire item, *Would you like to continue with MakeCode Arcade?* we investigated the difference between the answers of boys and girls, students from primary and secondary schools. Table 4 shows the hypotheses for this questionnaire item.

Table 4. Hypotheses for the questionnaire item *Would you like to continue with MakeCode Arcade?*

H_0	H_1	Probability of error	The null hypothesis
There are no significant differences in the answers of boys and girls as to whether they would like to continue teaching programming in MakeCode Arcade.	There are significant differences in the answers of boys and girls as to whether they would like to continue teaching programming in MakeCode Arcade.	00,00%	reject
There are no significant differences in the answers of students from primary and secondary schools, whether they would like to continue teaching programming in MakeCode Arcade.	There are significant differences in the answers of students from primary and secondary schools, whether they would like to continue teaching programming in MakeCode Arcade.	9,37%	accept

Based on the arithmetic mean, boys would like to continue programming games in the MakeCode Arcade environment more than girls. There are no differences in the answers of students from primary and secondary schools to this question. The answers were positive, both groups would like to continue in this environment.

In another open question, we asked what games they would like to program. Most of the answers were I don't know or I don't care, in addition to these answers, students would like to program games in the style of Super Mario Bros and Pac Man, single and multiplayer games, RPG games. To the last question, whether they wanted to say something to the authors, the students again mostly answered no, or that they were thankful for the opportunity to try game programming.

6. Discussion

Programming environment MakeCode is a very popular programming environment for teaching with educational board micro:bit, Lego EV3, and also for programming games in MakeCode Arcade. The main advantage of this environment is that it is free and that MakeCode developers keep adding new features. We used this environment for teaching programming and our small research confirms that this environment is suitable for teaching programming in Slovak schools. Unfortunately, there is not so much research on using this environment, but our small research confirmed similar results to Siever, Rogers, and Brown, the students liked this environment (more than Scratch), it is suitable for promotional activities for computer science, it is suitable for explaining the creation of scenes, game design principles. Comparing the answers of our students with the answers of the students from the research Siever and Rogers and Brown the students liked this environment and would like to continue in it.

We also verified the suitability of using the MakeCode Arcade environment in the teaching process at a free training for computer science teachers. 10 teachers from different schools participated in this training. After the 8-hour lecture, we asked the teachers for their opinion and had them fill out a short questionnaire. When asked whether the training met expectations, 9 teachers answered yes and one teacher rather yes. In the next question, we asked if they would use the tasks created by us for this environment in the teaching process - 5 teachers answered yes and 5 teachers rather yes. When asked if they think this environment is suitable for teaching, they all answered yes.

7. Conclusion

In this contribution, we describe our experience with the online environment MakeCode Arcade, which, in our opinion, is a suitable environment for teaching programming and for motivational activities to increase the interest in programming. This environment contains many created games that can be used for game-based learning. We tested this environment in the education process on selected students who are interested in programming (extracurricular activity and motivational online workshops), but also in normal lessons for primary school children. The suitability of this environment was confirmed by a questionnaire from participants (students) and by teachers from primary schools who completed a training at our department for this environment. Based on our experiences we will prepare more MakeCode Arcade tasks and we will continue with this environment in our activities to motivate children to study computer science. Teachers like this environment, as they can explain different game principles, the basics of graphics, animations. They can use one of the created educational games, or create their own game as needed.

Acknowledgments. This contribution has been processed as part of grant project no. 001UMB-4/2023 Implementation of blended learning in the training of professional bachelors and teachers of mathematics and computer science.

References

1. Omonayajo, B., Al-Turjman, F., Cavus, N.: Interactive and Innovative Technologies for Smart Education. *Computer Science and Information Systems*, Vol. 19, No. 3, 1549-1564. (2022), <https://doi.org/10.2298/CSIS210817027O>
2. Juškevičienė, A., Passey, D.: Outcomes of Computing Education. In: Tatnall A. (eds) *Encyclopedia of Education and Information Technologies*. Springer, Cham (2019).
3. The Committee on European Computing Education (CECE): *Informatics Education in Europe: Are We All in the Same Boat? 2017 Technical Report*. ACM, New York, NY, United States (2017). DOI: <https://doi.org/10.1145/3106077>
4. Dagienė, V., Jevsikova, T., Stupurienė, G., Juškevičienė, A.: Teaching Computational Thinking in Primary Schools: Worldwide Trends and Teachers' Attitudes. *Computer Science and Information Systems*, Vol. 19, No. 1, 1-24. (2022), <https://doi.org/10.2298/CSIS201215033D>
5. Sigman, M., Peña, M., Goldin, A., Ribeiro, S.: Neuroscience and education: prime time to build the bridge. *Nat. Neurosci.* 17, 497–502 (2014)
6. Sălceanua, C.: The Influence of Computer Games on Children's Development. Exploratory Study on the Attitudes of Parents. In *Procedia – Social and Behavioral Sciences*, 149, 837-841. (2014)

7. Ekaputra, G., Lim, Ch., Eng, K., I.: Minecraft: A Game as an Education and Scientific Learning Tool. In *Information Systems International Conference (ISICO)*. (2013)
8. Han, J., Liu, G., Gao, Y.: Learners in the Metaverse: A systematic review on the use of roblox in learning.”. In *Education Sciences*. Vol. 13, No. 3. (2023)
9. The Children of the Cost, Quality, and Outcomes Study Go To School, Available : <http://fpg.unc.edu/sites/fpg.unc.edu/files/resources/reports-and-policy-briefs/NCEDLCQOtechnicalreport.pdf>. (current May 2019)
10. Huotari, K., Hamari, J.: A definition for gamification: Anchoring gamification in the service marketing literature. In: *Electronic Markets*, 27(1), 21–31, (2017).
11. Warmelink, H., Koivisto, J., Mayer, I., Vesa, M., Hamari, J.: Gamification of production and logistics operations: Status quo and future directions. In: *Journal of Business Research*. (2018).
12. Caponetto, I., Earp, J., Ott, M.: Gamification and education: A literature review. *European Conference on Games Based Learning*. In *Academic Conferences International Limited*. Vol. 1. (2014)
13. Garcia-Iruela, M., Hijón-Neira, R.: What Perception Do Students Have About the Gamification Elements?. In *IEEE Access*. Vol. 8, 134386-134392, 2020)
14. Lai, C. H. , Lin, Y. C., Jong, B. S., Hsia, Y. T.: Adding Social Elements to Game-Based Learning. In *International Journal of Emerging Technologies in Learning (IJET)*. Vol. 9, 12–15. (2014)
15. Schaffer, D., W., Squire, K., D., Halverson, R., Gee, J. P.: Video games and the future of learning. In *Phi Delta appan*. Vol. 87, pp. 104-111. (2005)
16. Lamrani, R., Abdelwahed, E. H.: Game-based learning and Gamification to improve skills in early years education. *Computer Science and Information Systems*, Vol. 17, No. 1, 339-356. (2020), <https://doi.org/10.2298/CSIS12345678994833333>
17. Hussein, M. H., Ow, S. H., Cheong, L. S., Thong, M. -K.: A Digital Game-Based Learning Method to Improve Students’ Critical Thinking Skills in Elementary Science. In *IEEE Access*. Vol. 7, 96309-96318. (2019)
18. Boyle,, E. A., Hainey, T., Connolly, T. M., Gray, G., Earp, J., Ott, M., Lim, T., Ninaus,M., Ribeiro, C., Pereira, J.: An update to the systematic literature review of empirical evidence of the impacts and outcomes of computer games and serious games. *Computers and Education*, Vol. 94, 178–192. (2016)
19. Hafeez, M.: Effects of game-based learning in comparison to traditional learning to provide an effective learning environment: A comparative review. *Contemporary Educational Researches Journal*, Vol. 12, No. 2, 89–105. (2022)
20. Michael, D. Chen, S.: *Serious games Games that educate, train, and inform*. Boston, MA.: Thomson Course Technology. (2005)
21. Susi, T., Johannesson, M., Backlund, P.: *Serious Games : An Overview*. In Retrieved from Institutionen for kommunikation och information. (2007)
22. Christinaki, E., Vidakis, N., Triantafyllidis, G.: A Novel Educational Game for teaching Emotion Identification Skills to Preschoolers with Autism Diagnosis. *Computer Science and Information Systems*, Vol. 11, No. 2, 723–743. (2014), <https://doi.org/10.2298/CSIS140215039C>
23. Alamri, A., Hossain, M. A., Hassan, M. M., Hossain, M. S., Alnuem, M., Ahmed, D. T., Saddik, A. E.: A Cloud-Based Pervasive Serious Game Framework to Support Obesity Treatment. *Computer Science and Information Systems*, Vol. 10, No. 3, 1229-1246. (2013), <https://doi.org/10.2298/CSIS120717046A>
24. Malliarakis, C., Satratzemi, M., Xinogalos, S.: Optimization of server performance in the CMX educational MMORPG for Computer Programming. *Computer Science and Information Systems*, Vol. 11, No. 4, 1537–1553. (2014), <https://doi.org/10.2298/CSIS131220078M>
25. de Jesus, Â., M., Silveira, I., F.: A Collaborative Learning Framework for Computational Thinking Development through Game Programming. In *Informatics in Education*. Vol. 21, No. 2, 253-281. (2022)

26. Dele-Ajayi, O., Strachan, R., Pickard, A. J., Sanderson, J. J.: Games for Teaching Mathematics in Nigeria: What Happens to Pupils' Engagement and Traditional Classroom Dynamics?. In IEEE Access. Vol. 7, 53248-53261. (2019)
27. Kian, T. W., Sunar, M. S., Su, G. E.: The Analysis of Intrinsic Game Elements for Undergraduates Gamified Platform Based on Learner Type. In IEEE Access. Vol. 10, 120659-120679. (2022)
28. Kazimoglu, C.: Enhancing Confidence in Using Computational Thinking Skills via Playing a Serious Game : A Case Study to Increase Motivation in Learning Computer Programming. In IEEE Access. Vol. 8, 221831-221851. (2020)
29. Montes, H., Hijón-Neira, R., Pérez-Marin, D., Montes, S.: Using an Online Serious Game to Teach Basic Programming Concepts and Facilitate Gameful Experiences for High School Students. In IEEE Access. Vol. 9, 12567-12578. (2021)
30. Siever, B., Rogers, M., P.: Game On! Inspired CS Education with MakeCode Arcade. Proceedings of the 52nd ACM Technical Symposium on Computer Science Education (SIGCSE '21). In Association for Computing Machinery, New York, NY, USA. (2021)
31. Brown, S. A., Chu, S. L., Yin, P.: A Survey of Interface Representations in Visual Programming Language Environments for Children's Physical Computing Kits. Interaction Design and Children (IDC '21). In Association for Computing Machinery, New York, NY, USA, 268–275 . (2021)
32. Begel, A., Dominic, J., Phillis, C., Beeson, T., Rodeghero, P.: How a Remote Video Game Coding Camp Improved Autistic College Students' SelfEfficacy in Communication. In Proceedings of the 52nd ACM Technical Symposium on Computer Science Education (SIGCSE '21). Association for Computing Machinery, New York, NY, USA, 142–148. (2021)
33. Voštinár, P.: MakeCode Arcade: Interesting environment for programming 2D games. In 2021 IEEE World Conference on Engineering Education (EDUNINE). 1-6. (2021).

Patrik Vostinar is a young researcher who received his Ph.D. with a dissertation thesis “Multimedia as a motivational instrument in preparation for a Math teacher”. His research mainly focuses on training pre-service computer science teachers with innovative robotics aids, e-learning, mobile learning, microcontrollers, 3D modelling, 3D printing and developing virtual reality environments. Since September 2017, he has been an Assistant Professor in the Department of Computer Science, where he gives lectures and seminars in courses on programming, virtual reality, and many subjects related to teacher trainee students. He is the author and co-author of more than 78 publications in scientific journals and proceedings of scientific conferences and the author of two textbooks. He organises free extracurricular activities for students in primary schools, which aim to teach children programming for various types of robotics, programming, 3D printing, and creating virtual reality games. Moreover, he leads a training program for in-service teachers focusing on new popular technologies, ICT devices, e-learning and innovative software applicable to computer science education. <http://www.fpv.umb.sk/en/pvostinar/>

Received: November 11, 2023; Accepted: March 19, 2024.