Olympiads in Informatics, 2023, Vol. 17, 43–51

DOI: 10.15388/ioi.2023.04

Online Robotics Activities During the Pandemic Period – Challenges and Experiences

Bence GAÁL
ELTE Faculty of Informatics – Department of Media and Educational Informatics, Budapest
e-mail: gaalbence@inf.elte.hu

Abstract. The situation caused by the Covid-19 made it impossible to keep our traditional in-person summer camps and activities, so our university decided to hold the sessions online. Thus, enthusiastic students aged 13–16 were able to participate in robotics sessions that were fully implemented in the online space. In the article, we want to present the experiences and challenges of these sessions, as well as the way of implementation and the range of tools needed for it.

Keywords: robotics, Covid-19, distance learning, IT education, learning activities.

1. Introduction

The robotics workshops that are the subject of our article were organized by ELTE T@T Kuckó in the summer of 2020 in three different weeks. A different group took part each week. The content of the curriculum was the same each week. The three-week split was necessary to make the groups easier to manage and to allow more time for individual assistance. Since the target group was 13–16 years old, we did not want to implement coding in a block-based programming environment. The workshops were primarily designed for students who had not programmed in python and/or had not yet encountered the micro:bit. The aim of the workshop was therefore to learn the basics of programming in addition to getting to know the device. During the three weeks, each workshop started with the maximum number of students (14 students) and there was no dropout.

When choosing the programming language, it was important to use a language that is ideal for beginners. The advantages of python include its simplicity, security, and support for object orientation. (John, 1999) Research also shows that students found python more fun and easier to learn than the languages they had used before. However, it is important to note that there are also disadvantages to using the language. Since it is a scripting language, performance loss occurs when running longer programs. Another disadvantage is the lack of information encryption and dynamic type assignments (assigning multiple types to a variable) (Grandell *et al.*, 2006). From an educational point of view, however, it is advantageous that language requires the user to create code in a structured way because in python, indentations will indicate the beginning and end of code snippets. In other languages, readability is left to the programmer, while here the language requires us to code in a readable and “good looking” way (Donaldson, 2003). This helps students internalize this type of coding, which can be perfectly capitalized on in other languages as well.

When choosing micro:bit, we considered two main criteria. One was its excellent usability in education and the other was the price of the devices. Participants had to purchase their own devices to participate in the workshops. In addition to purchasing the device, no other costs were incurred by participants as the workshops were free of charge. Thanks to its versatility, it can be suitable for children with different interests and can be integrated into several areas of disciplines and programs related to the topic can be created (Abonyi-Tóth, 2018; Gaál, 2019). In addition, we talk about a compact device that can be easily extended even for beginner users.

2. The Curriculum of the Workshop

The main goal was to have enough knowledge and understanding of the device to implement a game on micro:bit by the end of the week. The main elements of the workshop’s curriculum were mainly based on the following workshop resources:

- Programming micro:bits in python – Norbert Szűcs[1].
- Programming micro:bits – Andor Abonyi-Tóth[2].

The curricula mentioned above must be adapted to the duration of the workshop. and the number of sessions, which took place in 5x2 hour intervals. Due to distance learning, we completely omitted the presentation of the device’s radio functions during the sessions, as this would have required more devices for the students.

Below we briefly explain the content of each session. Learning materials supported by full lesson plans can be found in the resources above. We tried to combine the classes with continuous independent assignments, which were submitted online. Do not forget to use these materials as a kind of outline and always try to customize the course of the lesson to the participants of the given course.

Session 1:

The essential part of the session was preceded by a so-called installation part. This is because the interface, which we will detail later, will only recognize devices with the latest firmware version, and only then will functions be available that greatly speed up workflows. This can be easily achieved by visiting the following website: `https://microbit.org/get-started/user-guide/firmware/`.

This was followed by the preparation of our first program, which was nothing more than the announcement of “Hello World!”. The rest of the lesson was spent drawing

[1] `http://microbit.inf.elte.hu/wp-content/uploads/2019/10/microbit_python_szakkor_szucsnorbert.zip`

[2] `http://microbit.inf.elte.hu/wp-content/uploads/2018/05/Programozzunk-microbiteket-2018.pdf`

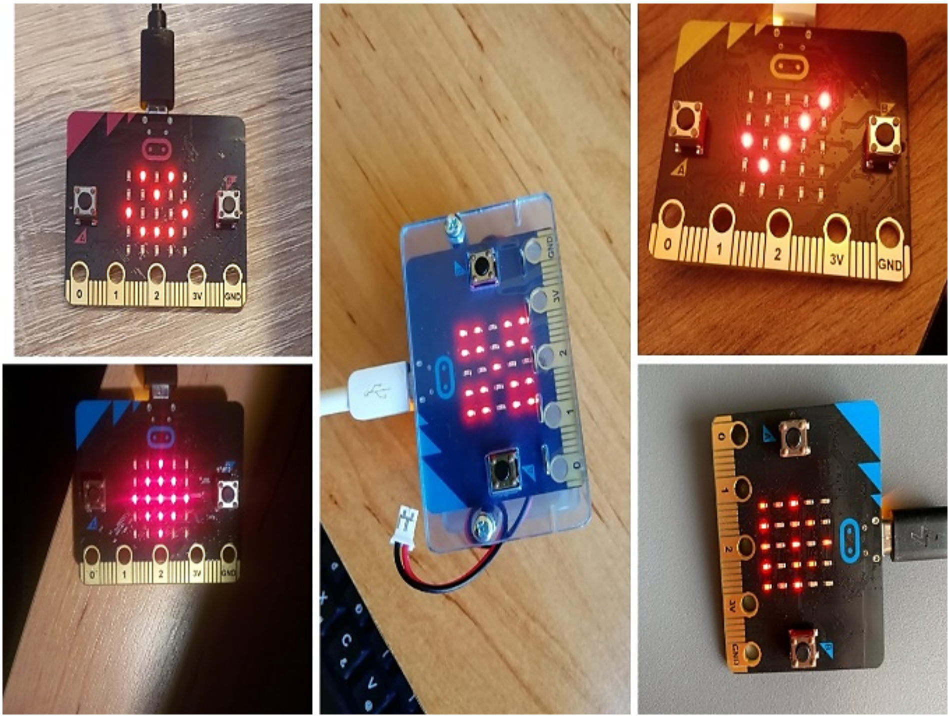

Fig. 1. Some results of students work on the first day.

shapes and animations on the micro:bit LED display. As an independent task, it was left to the children to take their own pictures (Fig. 1).

Session 2:

We also started the second session focusing on the display of the microbit. However, here we no longer displayed coherent images, but addressed the points of the LED matrix separately. Here random numbers and their role in IT came up. The attention of children can be captured very well by using their favorite computer games as examples and introducing new knowledge through them, so the presentation of random numbers is presented through the loot mechanism known from the games.

In the second half of the lesson, we implemented the animation of the bunk house available from the block curriculum in python language and the students also had time to implement a manipulated dice roll It can be concluded that a playful and interesting approach to random numbers greatly facilitates the acquisition of knowledge by children and their ability to apply it.

This is where the basics of event-driven programming through the use of buttons were presented.

Session 3:

In this session, sensors already played the main role. After creating the compass, we created a jump counter application in python.

With the students, we reviewed other ways to give instructions to the micro:bit besides button presses, and then we created our own magic 8 ball[3] program, where the data structure and operations of the list were already needed. The essential part of the implementation was the independent task of the students, after they became familiar with the commands for the new operations.

Session 4:

In the fourth session, we solved "do it yourself"-type tasks. This occasion was very popular among children during all three weeks. Here the following accessories were also needed:

- 4 crocodile clips with cable/lots of tinfoil.
- 2 pieces of nails.
- Pot and earth, which was preferably dry.
- 1 banana.
- 1 orange.

In the first half of the lesson, we made a fruit piano (Fig. 2), where we extended the micro:bit circuit with the help of ourselves and the fruits, thus playing sounds with the device. After making the piano, the students were given the task of making a siren.

In the second part of the session, we created a moisture meter (Fig. 3) that can be used in a smart home project, which monitors soil moisture in real time by monitoring soil resistance. Here an elaborate, micro:bit controlled irrigation system was presented, and the construction of smart homes was also discussed.

It was this task that really left its mark on the children, and taking sustainability and energy saving into account, many people thought about the task further and started to implement their own project.

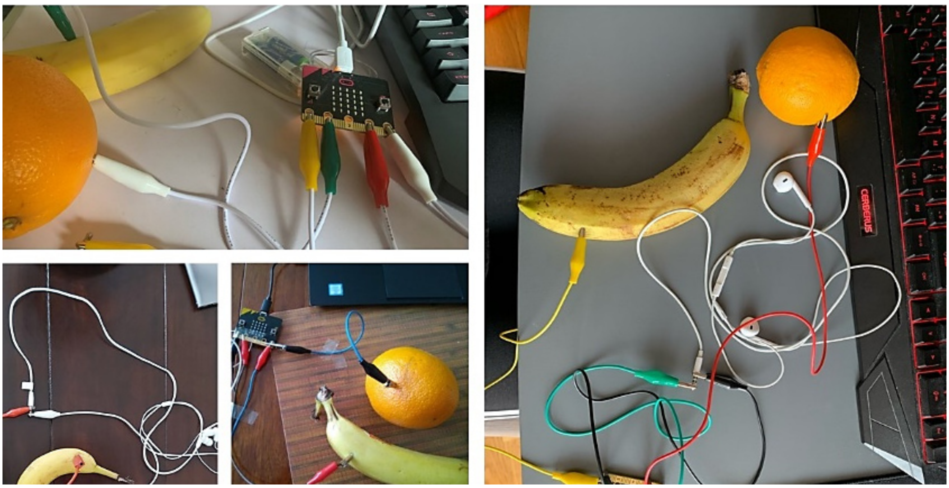

Fig. 2. Fruit pianos made by students.

[3] In fairy tales and films, the magic 8 ball has repeatedly starred, which randomly answers our questions.

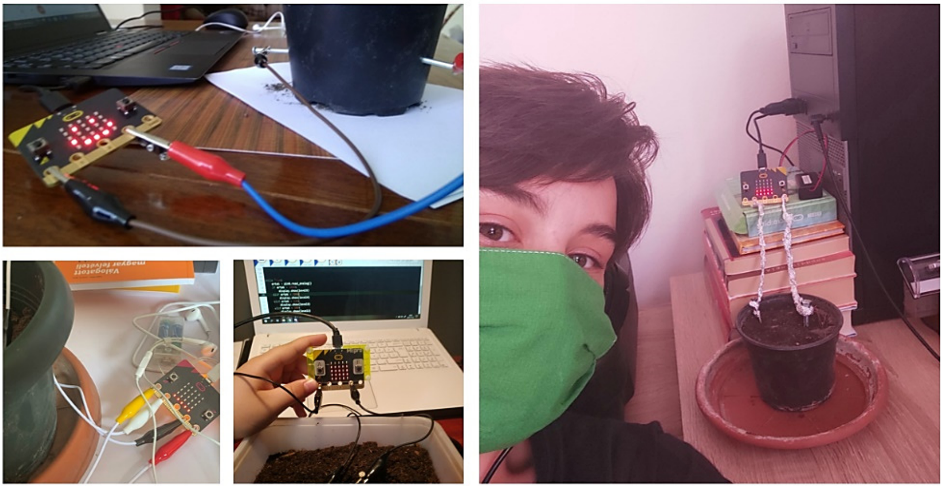

Fig. 3. The implemented moisture meters.

Session 5:

On the last day, the most awaited activity for the students took place. Here we implemented Flappy Bird on micro:bits. During the process of creating the game, we built heavily on what we have learned so far, and almost every single element we learned has been put into use. The students can now use what they had learned to create a complex program and receive feedback on how well they had mastered this knowledge.

During the creation of the game, object-oriented programming appeared as a new knowledge, and the participants gained insight into the different development phases. The programing process was divided into parts, and some sub-problems had to be solved by the students in an independent form of work.

3. Software Requirements of the Workshop, Presentation of the Virtual Environment

During the workshop, we used several software to create the most ideal environment for online education for students. We tried to choose programs that are suitable for organization and task assignment, holding online classes, visualization and of course coding and its supervision. Below we will review these platforms.

The virtual classroom and sessions

For this part of the implementation, we used Google services. On the Google Classroom interface it is possible to organize groups, assign assignments, and share information with students. It is perfectly usable for traditional school activities as well, as Google offers us a complete LMS.

In the "Classroom", we can enter the schedule into the group calendar, which can be associated with a conference call using the Google Meet application. "Meet" also has all the features needed to conduct online meetings. Children can easily start screen sharing, so the teacher can provide personalized help to them.

Several factors influenced the selection criteria. We needed a software package that not only allowed for classroom organization but also for holding online classes. It was also important to be able to easily integrate different file types into the site, as tasks had to be assigned and students had to upload pictures and videos. We needed a system that most students were familiar with and could easily handle and access. The Google software package fully met these criteria and only required an email address to access it.

If students under the age of 16 want to use Google accounts, logging in with their parent's email address can be a solution in non-institutional cases. However, institutions have the option of applying for the G-Suite system, which allows for a uniform email address to be given to students and these accounts can also be supervised by those with system administrator privileges. It is also worth noting that Google's software package is perfectly usable on multiple platforms, so students do not necessarily need a computer.

The virtual board

We would like to highlight a possibility for replacing board drawings in the case of online classes. We believe that board drawing is an indispensable pedagogical visualization tool in teaching programming, which helps students better understand processes and representations at an abstract level.

In our case, we used one of Google's applications for this, namely "Jamboard". In addition, we had a tablet with a pen. This can be replaced with a simpler drawing tablet or, if there is no tablet in our toolkit, we can draw with a mouse. Jamboard is a good choice because our virtual notebook can be shared with others and thus students can receive notes and create products together.

During the workshop, we opened the Jamboard application on the teacher's computer and participants could see what we were drawing on the tablet on the shared screen. This solution, where they do not open another application but only see the result projected, is a better solution so as not to overload their devices too much, as participants are not equipped with uniform strength hardware.

The online programming interface

We haven't talked about the programming interface yet. Online programming and collaborative document editing is not a new thing. Instead of well-established sites like repl.it, we chose a site that is still in beta but was specifically designed for the micro:bit. This interface is the micro:bit classroom.

On the site, we can program not only in python but also in a block-based environment. After selecting the language, the website generates an entry password consisting of icons and a pin code. This ensures that unauthorized persons do not enter the group.

In this article, we will focus on presenting the coding part of the website (Fig. 4) and explore its possibilities. The interface is perfectly usable in the field of online education. Its outstanding usability lies in providing solutions that make it easier for students to connect between the device and the computer and provide good opportunities for teachers to bridge distances. We will detail these properties below.

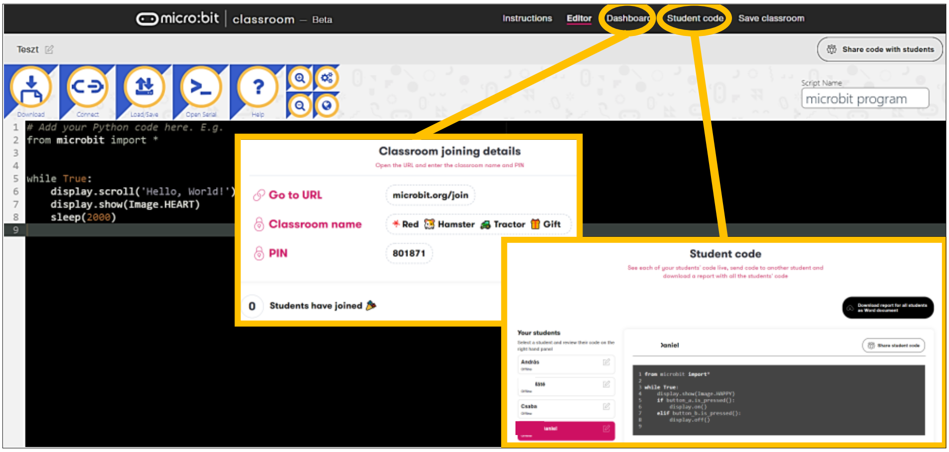

Fig. 4. Old interface of micro:bit classroom.

At the time of the workshops, the code editor was nothing more than a classic python IDLE. This was perhaps the only drawback of the environment that it does not contain any kind of verification system, as in Visual Studio, for example, or on the repl.it page mentioned earlier. Since then, however, an interface has been created that mixes block and text programming, and students can insert ready-made python code snippets in drag and drop principle. In addition, the editor provides tips related to various commands and debugging has been implemented. (Fig. 5)

We can also download our finished code, but if we have connected a micro:bit, we can skip all file operations. With a simple click, we can load our code directly onto the connected device. The connection of the device should be implemented as described earlier. As teachers, we can view the code of all students in the classroom and with one

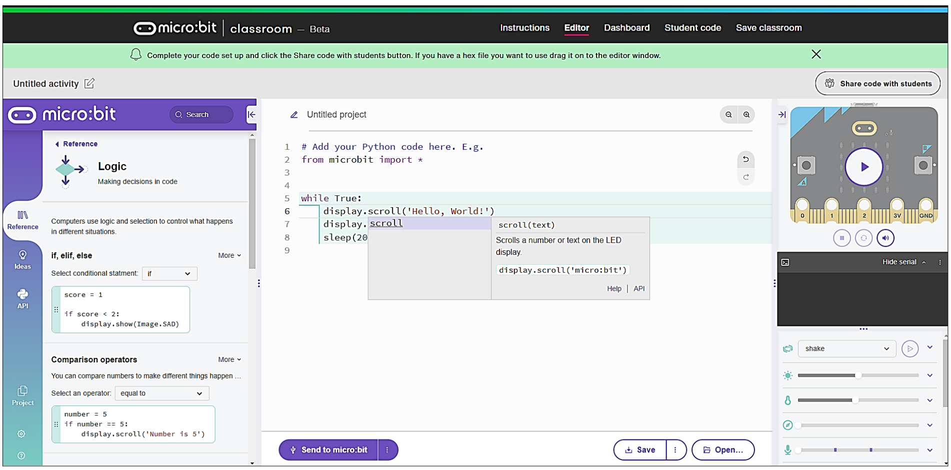

Fig. 5. The current version of micro:bit classroom interface.

click we can choose which student's code we want to see and easily help them. There is no need for screen sharing or other settings, and it is also a great advantage from a teacher's point of view that we can send the code written by us to all or selected students with one click. Students will be notified of this and can accept our code or continue working with their own. It is extremely useful for issuing framework programmes. In addition, we have the possibility to create a document of all codes and save it. This document will contain the codes of each student and structure them for us.

The interface was created specifically for micro:bit, and considering all the positive features mentioned above, it can be concluded that we can see the realization of an almost perfect concept on the micro:bit classroom site. Compared to the beta version stage, there were very few bugs and it has now taken the place of the text editor, a semi-block interface that is much more friendly to students.

4. Final Thoughts

Even within the framework of digital education, robotics can motivate children and stimulate their interest in programming. The group can be considered completely successful, as anyone had the opportunity not to come to the next event, but fortunately no one took advantage of this, and all students participate in every event. The participants had a good time during the workshop and perhaps despite the situation at that time, they managed to bond a little. There were several examples of students reaching out to each other during the afternoon as well, about learning or playing together.

Approaching it from the side of the device, it is important to note that we could make better use of the functions of the device if we are attending a classic classroom class. Nevertheless, we should not discourage holding online workshops either, as this way students who would almost certainly not have been able to attend robotics classes could not have been introduced to robotics by overcoming the distance, as they live hundreds of kilometers away from Budapest. We believe that with the help of online workshops we can reach more children, thus transferring the knowledge from which they can discover the world of programming and the functions of the given device on their own.

References

Abonyi-Tóth, A. (2018). A micro:Bitek felhasználási lehetőségei az oktatásban. InfoDidact 2018. Retrieved May 30, 2023, from `https://people.inf.elte.hu/szlavi/InfoDidact18/Manuscripts/ATA.pdf`

Gaál, B. (2019). A robotika témakör integrálásának lehetőségei a természettudományos tantárgyak ok-tatásában. InfoDidact 2019. Retrieved May 30, 2023, from `https://people.inf.elte.hu/szlavi/InfoDidact19/Manuscripts/GB.pdf`

Grandell, L., Peltomäki, M., Back, R.J., & Salakoski, T. (2006). Why complicate things? Introducing programming in High School using Python. In: Tolhurst, D. and Mann, S., Eds. *Proc. Eighth Australasian Computing Education Conference (ACE2006)*. Hobart, Australia. CRPIT, 52. ACS, pp. 71–80.

Donaldson T. (2003). *Python as a First Programming Language for Everyone.*

Zelle, J.M. (1999). *Python as a First Language.*

B. Gaál – PhD student and assistant lecturer at ELTE Institute of Computer Science- Department of Media & Educational Technology. He is a member of the Hungarian Bebras Organization Team. Graduated with an MSc degree in teaching Geography and teaching Informatics. His main research areas are the integrability of robotics in the natural sciences and the usage of micro:bit in education.