The Programmable Drone for STEM Education

Patrik Voštinár, Dana Horváthová, Nika Klimová

To cite this version:

Patrik Voštinár, Dana Horváthová, Nika Klimová. The Programmable Drone for STEM Education. 17th International Conference on Entertainment Computing (ICEC), Sep 2018, Poznan, Poland. pp.205-210, 10.1007/978-3-319-99426-0_18 . hal-02128591

The Programmable Drone for STEM Education

Patrik Voštinár[1], Dana Horváthová[2] and Nika Klimová[3]

[1,2,3] Matej Bel University, Tajovského 40, 974 01 Banská Bystrica, Slovakia
patrik.vostinar@umb.sk

Abstract. Technologies in the current world are constantly changing. These changes are also reflected in teaching process. Computer Science subject uses mostly programming language Scratch in Slovak schools. We wanted to try out a new tool for pupils as a way for programming and STEM education – modular and programmable drone Airblock. In this contribution we wanted to present this drone which we think is suitable for teaching programming. Drones are very popular among people and that was the reason why we think drones can be used as a motivational tool in order to teach programming at schools. This was done by the extracurricular activity at the Matej Bel University and the pupils were from lower secondary schools in Banská Bystrica, Slovakia. The pupils developed many buttons with different behaviours. The Airblock drone could fly without pre-programmed buttons. The subject matter of this article is a presentation of robotic toy – programming of the flying drone Airblock suitable for teaching programming.

Keywords: programming, drone, Airblock, STEM education.

1 Tangible technological devices

Nowadays technological devices are very popular. They are devices which provide interactivity, response and communication [1]. This includes programmable tools as well as programmable drones.

Technological devices should be included into early experiences because everyday technologies are controllable; engaging in control activities obliges pupils to deal with and to construct simple programs; control technology activities may help them to develop more general abilities to think and learn [2].

Programmable toys appeared to be a good choice because they are tangible technological devices and pupils can directly manipulate with them; they can stimulate problem-solving in real conditions of pupils' environment [3].

There are 3 ways of receiving information [4]: visual – sights, diagrams, symbols, pictures; auditory – words, sounds; kinaesthetic – touch.

We wanted to use these modalities into our learning process. It was important to include all of them and make our lessons tangible. Because of this, programmable drone Airblock meets our requirements.

It is generally known how much important a motivation is – especially internal motivation. While outside motivation can be added to the education process by the teachers and family, the internal motivation needs to be awakened for the pupils regardless of the possible reward [5].

In Slovak schools Computer Science subject uses mostly programming language Scratch. This programming becomes more self-evident. Because of this reason we should be looking for a new motivation for teaching programming. New motivation could be Airblock: the modular and programmable drone.

2 Micro-project: Computer Science Extracurricular Activity

Micro-project: Computer Science extracurricular activity was realized with support of the Comenius Institute and the Department of Computer Science of the Faculty of Natural Sciences at the Matej Bel University in Banská Bystrica. There have been involved around 75% of future Computer Science teachers of all grades from our faculty. Our students had the opportunity to work actively with elementary and secondary children from the beginning of their bachelor study at the Department of Computer Science. Indicators of the progress of the participating students were reflection, self-reflection and assessment of individual pedagogical competences through assessment scales.

The Comenius Institute offers to Slovak teachers (including future teachers) one-year costless practice course with many workshops. There are chosen yearly only 20 participants. Nika Klimová (one of the authors) was chosen as the only future teacher of these 20 participants (the rest of them were teachers or principals). She received money to develop her micro-project – something special that could be used in the teaching process. She decided to buy a learning tool – drone Airblock: the modular and programmable drone. We supposed using this drone could increase interest of teaching programming. Because of our experimental lessons took place for the first year, we did not come to a more formal form of study on the curve of learning, efficiency and motivation, and therefore we cannot analyse our results with other teaching processes.

2.1 Airblock: The Modular and Programmable drone

Airblock is the modular and programmable drone. It can be turned into a flying drone, hovercraft, water hovercraft, spider etc. It is made from engineering foam (it is indoor-friendly) and includes 1 main module (CPU) and 6 dynamic magnetic hexagons that can be assembled or disassembled without the need of tools (see Fig.1). Drone can be programmed by “drag-and-drop” programming. Airblock was the winner of Reddot Award: Product Design 2017, Good Design Award 2017, K-Design Grand Prize 2017 and Idea Finalist 2017.

It is based on STEM education joining science, technology, engineering, and mathematics together.

Hovercraft mode allows Airblock to do a number of stunts, and by selecting two propellers from either side, it can generate specific movements. For instance, the two propellers located on the back of the hovercraft drive the hovercraft forward.

The two propellers located in the front drive the hovercraft backward. Getting two propellers to push against each other enables the hovercraft to go left or right.

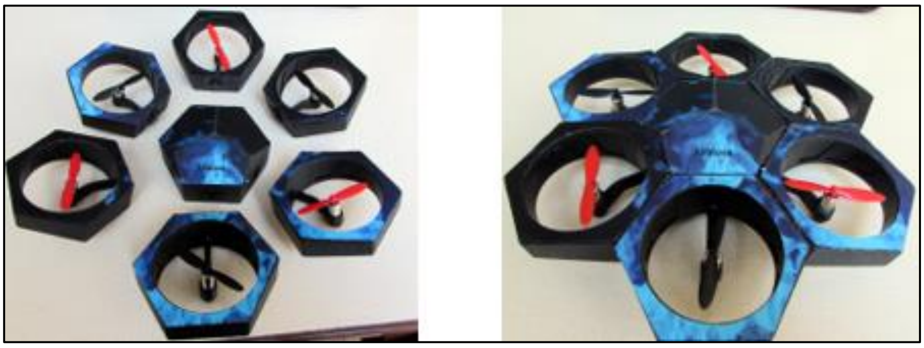

Fig. 1. Flying drone

For the programming drone we need a mobile device such as tablet or mobile phone (smartphone). We can download a mobile application Makeblock which could be downloaded for the operating system Android (free app at Google Play) and iOS (free app at App Store).

2.2 Programming Airblock

Programming in this environment is comparable to programming in Scratch. Simply drag-and-drop different blocks of commands – like forward, pause, turn, etc. and connect them together to create a seamless action.

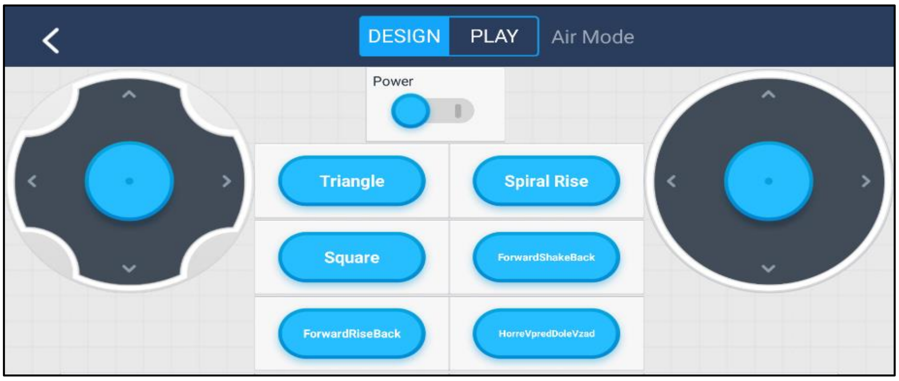

Fig. 2. Makeblock application – code environment.

In the application Makeblock it is possible to design your own environment of the controller. Into environment application can be added joystick control or predefined buttons for flying drone into circle, triangle, etc. Predefined buttons can be modified. It is possible to create your own buttons with different behaviours (see Fig. 2).

The application allows two modes – design and play. In the *design mode* you can add some components or develop some behaviour. In the *play mode* you can test your application. Behaviour can be modified after some buttons if *design mode* is pressed.

Drag-and-drop blocks are divided into following categories (see Fig. 3):

- **Begin** – what happens when button is pressed/released, power on/off.
- **Move** – motion of the drone – move, turn, rising, roll, shake, hover and landing drone.
- **Display** – offer possibility to set LED on the drone.
- **Event** – shake tablet, switch on/off tablet, tilt tablet.
- **Detect** – slider value, read gyro value, temperature, battery level
- **Math** – mathematics operation (addition, subtraction, Boolean value, change item, round, random integer, etc.)
- **Controls** – if condition, wait, repeat, break the loop, etc.

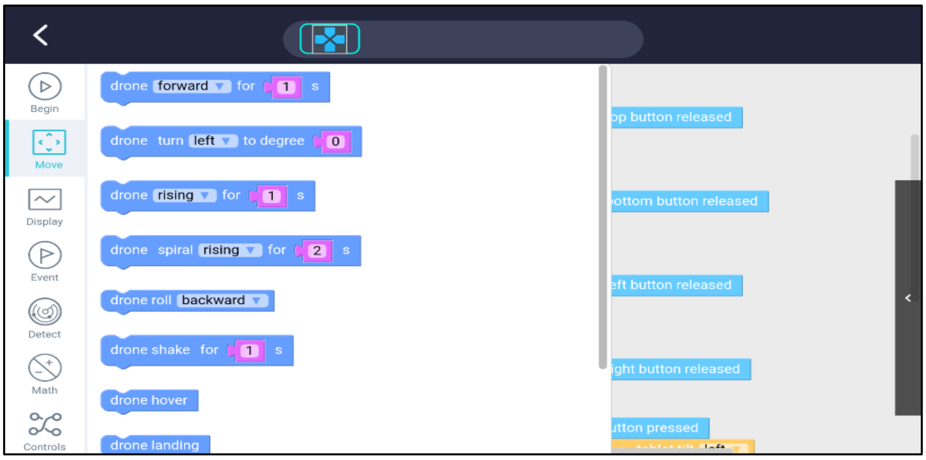

Fig. 3. App Makeblock – example of “drag-and-drop” blocks.

2.3 Drone Airblock – experience

We were teaching lower secondary school children *Program your drone!* 5 lessons at Computer Science Extracurricular Activity. During this activity there were children from 8-12 years old, but it wasn't limited by the age. Our Extracurricular Activity was voluntary for children and for future teachers and it was without any fee. For these lessons we used following methods: brainstorming – to come up what students know about drones and what they think drones can perform. Based on STEM education, we were discussing how the drone can fly and students were trying out to build the drone into many flyable shapes. Then we were developing programs through [6]:

- algorithm and data structure planning,

- coding,
- testing,
- error detection,
- correction,
- efficiency control, quality control,
- documentation.

Brainstorming turned out that all of pupils know what a drone is and could describe drones e.g. can fly or move some lightweight objects.

We taught pupils in a group and algorithm planning was made similarly as brainstorming with having a task of what the drone should perform, e.g. a triangle in the air. Students came up how to program it. After that they coded it together with the support of the group (through discussion). Testing showed if the program is correct. In the case of performing triangle it was wrong, so we were finding an error which occurred in angles. We tested it again. Then we were discussing if the program can be more efficient. Finally, we documented it through a print screen. This method was used for all tasks.

Feedback showed that programming of the Airblock drone was very exciting for the pupils (see Fig. 4). They said they created a lot of different environments and programmed many drone behaviours. They also had "much fun" and the best part of the lesson was testing.

Fig. 4. Students were developing app for the drone.

For example, they created buttons with following behaviour:

- **drawing 2D geometrical shapes** such as triangle, square, rectangle, circle, etc.,
- some of the students were able to develop **3D geometrical shapes such as** cube and rectangular prism,
- **traffic light:** drone turn on, shows the red colour for 1 second, wait for 1 second, change colour to orange for 1 second, wait for 1 second, change colour to green for 1 second and hover until 1 meter, do spiral, land,
- **action –** repeat 10 times: rising for 1 second, wait for 1 second, set all LED colours to red, downward for 1 second.

3 Conclusion

In this article we described educational tool Airblock: the modular and programmable drone. This drone could be used for increasing motivation of pupils for teaching programming. Pupils can programme behaviour of the drone in blocked programming language similar to Scratch. They created many drone control environments with their own buttons. They “had much fun” and the best part of the lesson was testing.

We want to improve the teaching of drone at Computer Science Extracurricular Activity by improving pedagogical backgrounds and monitoring related research in the future.

Acknowledgements

This contribution has been processed as part of the grant project *Implementation of Blended Learning into Preparation of Future Mathematics Teachers*, project no. 003UMB-4 / 2017th and project *Interactive Applications for Teaching Mathematics at Primary Schools*, project no. 003TTU-4/2018.

This contribution has been supported also by the Comenius Institute and its donors:

Nadácia orange

NADAČNÝ FOND TELEKOM

VARKEY FOUNDATION
Changing lives through education

Nadácia ESET

References

1. Learning and Teaching Scotland: Early Learning, forward Thinking. Learning and Teaching Scotland, Dundee, Scotland (2003).
2. Siraj-Blatchford, J., Whitebread D.: Supporting Information and Communications Technology in the Early Years. 4th Edition. McGraw-Hill Education, London (2003).
3. Pekarova, J.: Using a programmable toy at preschool age: why and how. In Proceedings of SIMPAR-2008, pp. 112-121. Simpar, Venice (2008).
4. Felder, R. M., et al.: Learning and teaching styles in engineering education. Engineering education, 78(7), 674-681 (1988).
5. Picka, K., Pešková, K.: Perception of digital games as educational media by lower secondary school pupils. Journal of Technology and Information Education 10 (1), 17-33 (2018).
6. Szlávi, P., Zsakó, L.: Methods of teaching programming. Teaching Mathematics and Computer Science 2(7), 247-257 (2003).