Uluslararası İleri Doğa Bilimleri ve Mühendislik Araştırmaları Dergisi
Sayı 7, S. 441-444, 10, 2023

Araştırma Makalesi

International Journal of Advanced Natural Sciences and Engineering Researches
Volume 7, pp. 441-444, 10, 2023

Research Article

https://alls-academy.com/index.php/ijanser
ISSN: 2980-0811

Program Code Simulation by Using Ozobot

Krisztina Czakóová [*1], Ondrej Takáč [1] and Ladislav Végh [1]

[1]*Department of Informatics, J. Selye University, Slovakia*

[*]*(czakoovak@ujs.sk) Email of the corresponding author*

(Received: 21 November 2023, Accepted: 27 November 2023)

(4th International Conference on Engineering and Applied Natural Sciences ICEANS 2023, November 20-21, 2023)

ATIF/REFERENCE: Czakóová, K., Takáč, O. & Végh, L. (2023). Program Code Simulation by Using Ozobot. *International Journal of Advanced Natural Sciences and Engineering Researches*, 7(10), 441-444.

Abstract – Visualization of the problem solving process greatly supports the deep learning and problem solving skills. In this paper, we focus on using of modern technologies in the teaching of informatics, which is highly motivational for programmers. The presented programmable robotic tool Ozobot contributes to experiential teaching of informatics. The robot is easy to control and is capable of making visualization of program code with immediate feedback regarding the current situation in the game. The learners can easily create programs without the need to write commands or to understand programming concepts. The non-textual code representation (using symbolic notation or block-based programming) supports the prevention of syntax errors in high level. The beginners in programming can store the program code by the selection of valid commands and style it by moving blocks or tiles. Young programmers can think creatively while playing and solving a new algorithmical problems. We will introduce the possibilities of Ozobot coding and detail the possibilities of building programming skills and techniques at the lowest as well as at medium level, based on the programming activities. Based on the complexity and variety of the commands OzoBlockly programming environment provides the programming in five categories of difficulty. Our recommended programming activities are based on the interactivity and experimentation.

Keywords – Block-Based Programming, Program Code Simulation, Ozobot, Visualization, Experimentation

I. Introduction

The interactive programming and the visual programming languages support the understanding of program structures that are required to be used later in programming studies. For beginners it is important to easily to create a program without the need to write and remember textual commands. Playing the problem (by playing a program code fed into the robot) greatly supports the deep learning as well as the problem solving skills. It makes programming fun and a good game [1] - [3].

II. Possibilities of Programming the Ozobot in Favour of Active Learning of Programming

Playful programming highly motivates the learners to improve their algorithms and design new solutions. Learners can plan their own games by using programmable robotic devices and also endow the main "live" figure (object, robot). It is a good idea to use such a programming environment, which can help in preventing (syntax) errors while programming (such as: shape matching, selection of valid options etc.) [4].

III. Materials and Methods

programming prerequisites we set in every way. The robot is easy to control and is capable to make visualization of program code with immediate feedback regarding the current situation in the game [4] - [6].

A. Content of the curriculum in programming

Table 1 represents the content of the curriculum of programming in primary informatics, focused to use programmable robotic toy Ozobot for algorithmic problem-solving [4].

Table 1. Content of the programming - primary informatics

Problem analysis	**Program structures to control the robot**	**Possibilities of programming the Ozobot**
And / or / not. Command line and repetition stepping. Faulty command detection.	For loop.	OzoBlockly programming in level 2.
Recognition of relations (repeating parts, recognizing the problem of decision making).	Variables, loops and conditions.	OzoBlockly programming in level 3.

B. Ozobot - Tool of the programming

The miniature programmable robot Ozobot Evo (Figure 1) is controlled by the dedicated mobile (as well as PC) application. A few sensors, motors, buttons and colourful diodes are available to control and move the robot.

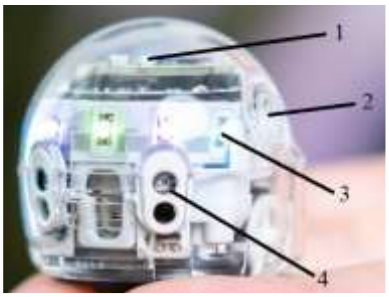


Fig. 1 1 – Status LED, 2 - on / off button, 3 - charging indicators LEDs, 4 – optical sensors.

We will detail the possibilities of building programming skills and techniques in OzoBlockly

We have chosen the Ozobot as a tool for active programming for beginners. Robot meets the programming environment at the lowest as well as at medium level [4].

OzoBlockly is an educational graphics-oriented, visual, block-based programming environment. It allows programming in the well-known and widespread Scratch programming environment. The OzoBlockly app can be easily downloaded to either a personal computer or a mobile device (https://ozobot.com/create/ozoblockly). According to the commands complexity and variety it provides the programming in five difficulty categories [4]. The levels range from beginner to advanced level of programming (Figure 2-3).

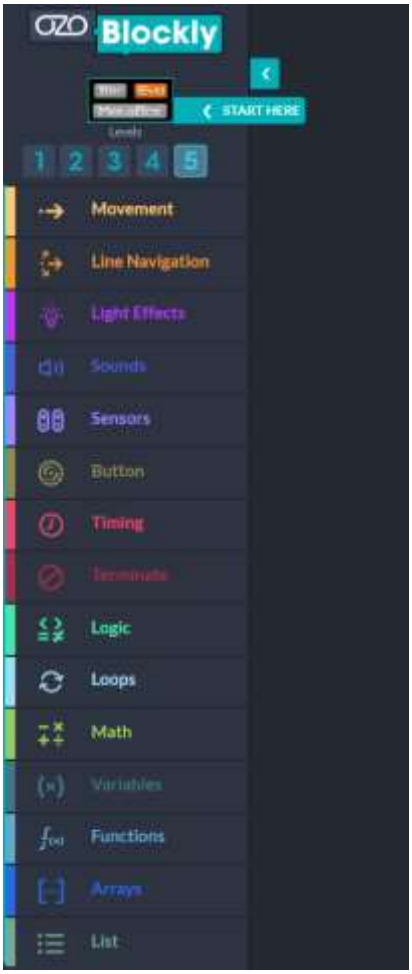


Fig. 3 OzoBlockly – commands list at the 5th difficulty level of programming

We introduce some programming activities, which received positive reception from primary school learners during the pedagogical practice [2].

C. Programming activities in OzoBlockly

Activity: Run to home!

Educational objectives

Students will be able:

- to recognize and interpret command blocks,
- to build command blocks in an appropriate sequence structure,
- to write a program in OzoBlockly at level 1 (beginner level),
- to run the completed program, to recognize errors and correct them.

Task description

On the illustrated track (see Figure 4), do instruct the Ozobot to follow the route to find home (the house)! If the robot arrived home, let it be happy with laughter and flash in the colours of the rainbow!

Fig. 4 The program code of the task solution in OzoBlockly programming environment and the task illustration [2]

Observing Ozobot on the track

We start the Ozobot by placing it on the left side on the track route (Figure 4): black line tracking to the bend (Movement category/ Default speed/ Move Forward [X steps]), turn to the left (Movement category/ Turn left), black line tracking to the house (Movement category/ Default speed/ Move Forward [X steps]), reaching the house - finish (Light effects category/ Rainbow, Wait category/ Wait [X seconds], Sounds category/ Play Sound Happy).

Uploading the program code into the Ozobot

Connect the Ozobot to your device using a USB cable! Select the appropriate Ozobot model (Ozobot EVO version) in the OzoBlockly program interface! Convert the finished code using the "Compile" button so that it can be understood by the Ozobot! When ready, click the "Upload" button to upload our code to Ozobot (see on the Figure 5)! After the upload is successful, disconnect Ozobot from your device! Let's test our program live by starting the robot!

Fig. 5 Process of loading the program code into the robot

Activity: Guide Ozobot to a given target!

Educational objectives

Students will be able:

- to formulate a command and put it in a sequence,
- to put certain commands into a for loop (with a given number of repetitions),
- to understand the relationship between the result and the number of repetitions (the concept of loop),
- to define a simple condition,
- to write the program in OzoBlockly at level 3 and run the program,
- to run the completed program, to recognize errors and correct them.

Task description

Guide the Ozobot with your hands to reach the specified destination (blue finish line)! Once that's done, let the robot sing while flashing colours!

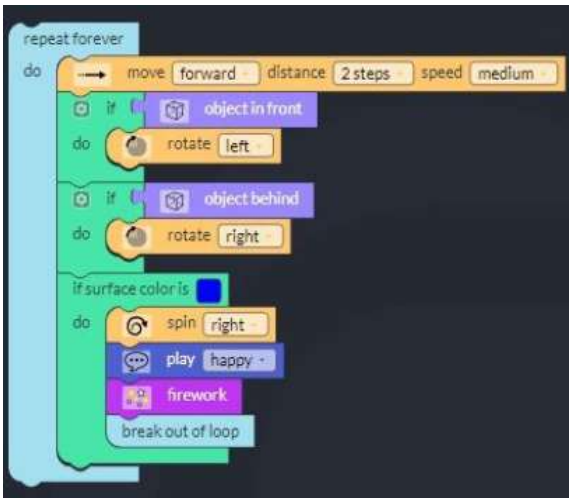
Observing Ozobot on the track

The movement of the robot is defined as moving forward with a distance of 2 steps at medium speed. We have to put this movement in an infinite loop block (repeat forever - do), which is valid until it reaches the finish line (reaching the blue finish line). Using the front sensors of Ozobot, we determine how it should react to an object in front of it (our hand). We need to select “if object in front – do” from Logic category and put inside the condition the “rotate left” command (has added from the Movement category). This specifies to turn left

(avoid it) when it senses an obstacle in front of it. We enable the same setting for the rear sensors so if it detects our hand it direct the Ozobot to the right of the obstacle. In this way, Ozobot will detect the movement of our hands in a given direction. After that, we also have to program to detect colours while the robot is moving. If the robot's sensors detect a blue colour, it reached the finish line. To make sing and flash lights we select the "if surface colour is" command from the Logic category and set the condition to blue (it must execute the following commands one after the other: spin right, play happy, firework, break out of loop). The program code is seen on the Figure 6.

Fig. 6 The program code of the task solution in level 3 of OzoBlockly programming environment [2]

IV. RESULTS



Our experiences from practice are obvious. Learners work with programmable robots intuitively. Their approach to programming and solving a given problem was creative. Interactive programming environment of OzoBlockly allows to follow the steps of the algorithm, which leads learners to a better understanding of individual commands and program structures. They quickly recognized the colour coded classification of commands. The structural unit of the program was easily understood. During the practical testing, the learners successfully guided the robot with their hands. This activity turned out to be a very fun [2].

V. CONCLUSION

Opportunity of visual feedback during the interactive programming highly supports mastering the curriculum as well as deep learning. Our recommended programming activities are based on the interactivity and experimentation [7], using the block-based programming environment. The aim is to introduce learners to learn computational thinking and solve new problems by active learning [8]. We showed possibilities of the mentioned programming languages and its coding levels and we shared the practical experiences. The learners showed positive feedback during the programming activities.

ACKNOWLEDGMENT

The paper was supported by the national project, KEGA 013TTU4/2021 "Interactive animation and simulation models for deep learning."

REFERENCES

[1] N. Annuš, D. Paksi, M. Csóka, "Interactivity and animation-simulation tools in the digital presentation of educational material," i*n INTED*, 2023 DOI: 10.21125/inted.2023.1895, pp. 6975-6980.

[2] B. Csóré, "Experience-based programming with Ozobot robot (Élményalapú programozás Ozobot játékrobottal)," M. thesis, J. Selye University, Komárno, Slovakia, May 2023.

[3] N. Forman, M.S. Avornicului, "Challenges in teaching informatics," *IJANSER*, vol. 7(6), pp. 381–386. DOI: https://doi.org/10.59287/ijanser.1177, 2023.

[4] K. Czakóová, M. Csóka, "Ozobot coding challenges in elementary school informatics," in *INTED*, 2022, DOI: 10.21125/inted.2022.1491, pp. 5834-5840.

[5] V. Stoffová, M. Zboran, "Educational robotics in teaching programming in primary school," in *ICRIC* 2022, vol. 1001, pp. 669-682.

[6] J. Udvaros, N. Forman, M. Avornicului, "Developing computational thinking with microcontrollers in Education 4.0.," *IJANSER*, vol. 7(4), pp. 103-108. DOI: https://doi.org/10.59287/ijanser.563, 2023.

[7] L. Végh, O. Takáč, "Online games to introducing computer programming to children," in *INTED*, 2021, pp. 10007-10016.

[8] K. Czakóová, "Virtual programming environments and simulations in favor of active learning of programming," *IJANSER*, vol. 7 (5), pp. 105-109. DOI: https://doi.org/10.59287/ijanser.910, 2023.