TEM Journal. Volume 13, Issue 1, pages 718-725, ISSN 2217-8309, DOI: 10.18421/TEM131-75, February 2024.

Research on Possibilities and Benefits of Programming Educational Robots for Real-Life Scenarios Imitation

Tsvetanka Georgieva-Trifonova [1]

[1] *"St. Cyril and St. Methodius" University of Veliko Tarnovo, Veliko Tarnovo, Bulgaria*

Abstract **– The research interest in programming educational robots is aimed at determining the specific advantages and challenges that educational robots provide in the learning environment. In this paper, the possibilities and benefits of programming educational robots specifically to imitating real scenarios are studied and explored. Typical scenarios with educational robots corresponding to real situations are systematized, the requirements to which the robots must meet and the necessary functionalities for implementation are described. Concrete implementations of projects in the environment of the educational robotics system Engino Robotics Platform are represented, in order to confirm the practical feasibility and acceptable level of difficulty of the scenarios. Programming educational robots by simulating real-world scenarios not only provides valuable experience, but also develops key skills essential to students' future success.**

Keywords **– Educational robots, Engino robotics platform, real-life scenarios imitation.**

1. Introduction

Educational robots can stimulate productive, creative and divergent thinking [1] by creating situations where it is necessary for problem solving, creativity, cooperation and communication, experiential learning, application of theoretical knowledge.

DOI: 10.18421/TEM131-75
https://doi.org/10.18421/TEM131-75

Corresponding author: Tsvetanka Georgieva-Trifonova, *"St. Cyril and St. Methodius" University of Veliko Tarnovo, Veliko Tarnovo, Bulgaria*
Email: cv.georgieva@live.uni-vt.bg

Received: 19 October 2023.
Revised: 27 January 2024.
Accepted: 31 January 2024.
Published: 27 February 2024.

Working with robots connects theoretical knowledge from various fields such as mathematics, physics, informatics, and technology with their practical application. By this way the students are supported to understand the meaning of the school contents in real situations. Teaching educational robot programming by simulating real-world scenarios is extremely useful and relevant for several reasons:

- Real life applicability;

Programming robots to perform specific tasks that resemble real-life situations aids students to insight how technology can be used to solve practical problems. This deepens their understanding of how programming can be applied in real-world situations.

- Development of logical thinking and problem solving;

Programming robots requires formulating sequential instructions for the robot to follow. In this way, the skill of logical thinking and algorithmic problem solving is developed, which is an important skill in every area of life [2].

- Creativity and innovation;

While students have the opportunity to program robots for specific scenarios, they are encouraged to be creative and innovative. They must find the best ways to solve challenges and complete tasks.

- Multidisciplinary training;

Programming educational robots usually involves diverse aspects such as mathematics, physics, engineering, and even art [3]. This promotes multidisciplinary learning and linking knowledge from different fields.

- Preparation for future work;

Technological advances continue to change the job market. Understanding the fundamentals of programming and working with robots can prepare students for future career opportunities in industries such as engineering, programming, robotics, and automation.

- Fun and motivation.

Similar to educational mathematical games designed for children and built using appropriate information technologies [4], [5], learning through practical and interesting robot scenarios can be extremely fun and motivating for students.

They get involved in the active learning process and have the opportunity to see the results of their efforts in real time.

The current research is aimed at programming educational robots to mimic real-life scenarios.

2. Systematization of Scenarios with Educational Robots

In the present section, scenarios for programming educational robots are systematized to resemble real scenarios and the functionalities that are necessary to be able to execute them (Table 1).

Table 1. Scenarios for programming educational robots that resemble real situations, the functionalities necessary for their implementation

Scenario	Functionalities	Requirements for the educational robot
Line following: The robot has to follow a colored line by moving along the line. This scenario can resemble a robotic assembly or packaging line in a manufacturing industry.	Reading data from the infrared sensors under the robot; Line-following algorithm – react to color change; Changing the speed and direction of movement depending on the reading of the sensors.	Wheels and motors for moving forward and backward, turning left and right; Infrared sensor for line following or color recognition sensor.
Avoiding obstacles: The robot must move on the ground while detecting obstacles and circumventing them. This can resemble: the operation of an automatic cleaning robot in the home that avoids furniture and walls; the self-driving robots that are used in industrial or agricultural applications where obstacles must be avoided.	Using an ultrasonic sensor to determine distances to obstacles or an infrared sensor to detect an obstacle; Algorithm for reacting to the proximity of an obstacle – stopping, turning, etc.; Programming maneuvers to go around obstacles.	Wheels and motors for moving forward and backward, turning left and right; Ultrasonic distance sensor and/or infrared sensor for obstacle detection.
Environmental data collection: The robot must measure distances to objects, ambient temperature or light intensity. This scenario corresponds to real situations for environmental monitoring, data collection and recording. It resembles the use of data collection robots in various science and engineering projects.	Usage of various sensors (e.g. temperature sensor, light sensor, ultrasonic sensor); Programming the frequency and method of data measurement; Recording of collected data; Perform calculations with the collected data and display the results.	Ultrasonic distance sensor; Temperature sensor; Light intensity sensor.
Goods delivery: The robot has to move along a predetermined route and deliver small objects or symbolic packages to certain locations. This scenario resembles the use of delivery robots in various situations, such as the delivery of food and goods.	Setting route points; Programming of movement between points, including stopping and turning; Use of colored markers or flags to identify delivery locations.	Wheels and motors for moving forward and backward, turning left and right; Ultrasonic sensor or infrared sensors to avoid obstacles; Color recognition sensor; Sensors for orientation and navigation.
Automated house: It is necessary to control lighting upon entry or dimming; to control air conditioning/fan on entry or based on set parameters such as outside temperature. It resembles automation in a real home.	Turning lights on and off; Monitoring change of light intensity; Using an infrared sensor and/ or a temperature sensor; Switching on and off the air conditioner/fan; Timer programming.	LED lights; Infrared sensor for object detection; Light sensor; Temperature sensor; Timer; Fan drive motor.
Pedestrian crossing: It is necessary to simulate pedestrian crossing signals by matching them with those of vehicles moving (or stopping).	Turning lights on and off; Programming the synchronous operation of traffic lights for vehicles and traffic lights for pedestrians.	LED lights; Buzzer.

The examples described in Table 1, are the most typical use cases for educational robots that can be applied in different situations and environments. They demonstrate the variety of possibilities for programming educational robots that simulate real situations from various areas of life.

3. Projects with Engino Robotics Platform (ERP)

The Engino Robotics Platform is an educational system designed for learning purposes that allows students to learn robotics and programming [6], [7], [8], [9]. It provides a comprehensive set of components with which students can create and program a variety of robots and automated systems [10], [11], [12], [13], [14].

3.1. Features of Engino Robotics Platform

The main features of the Engino Robotics Platform are:

- Modular construction;

The system uses a modular design that allows students to assemble robots and devices using different modules and components. This gives flexibility and the ability to create a variety of constructions.

Together with the controller [15], external sensors are developed that can be directly connected to it – infrared sensors for following a line and for object detection, a touch sensor. They allow robots to interact with their environment. Additionally, there are three engines that robots can use for movement and actions; buzzer; LED lights.

- Block programming;

The platform provides a graphical programming environment that allows students to create programs by dragging and dropping blocks. This type of programming is suitable for beginners and children. Through the blocks, it is possible adding conditional statements, statements for loops; defining and calling a subroutine; reading data from sensors; controlling the motors, buzzer, LED lights. Blocks can be executed either sequentially or in parallel.

In addition, the environment automatically generates a text description of the included blocks, which also allows changing the set parameters.

- Mobile apps;

The mobile applications that support the work with ERP are:

  - Engino kidCAD (3D Viewer) visualizes every step of assembling the models, supporting imagination and spatial thinking.
  - EnginoRobot BT enables ERP management via Bluetooth connection.

- Educational resources;

The Engino Robotics Platform provides educational resources such as study materials, guides, and assignments that help students understand the fundamentals of robotics and programming.

- Suitable for different ages;

The platform offers products and components that can be used by students of various ages and experience levels. This makes the Engino Robotics Platform suitable for both beginners and more advanced users.

- Variety of scenarios;

By using the Engino Robotics Platform, students can create robots for various scenarios such as simulation of real-world situations, games, educational research, and more.

- Extensibility.

The platform provides opportunities for expansion by adding additional components and modules, allowing students to develop robotics skills and knowledge incrementally.

The Engino Robotics Platform is an educational robotics system that combines learning with robot assembly and programming to create an interactive and educational experience.

3.2. Description of the Implemented Projects

In order to confirm the practical feasibility and acceptable level of difficulty of the scenarios discussed in Section 2, some of them are implemented. In this section, the developed examples with the Engino Robotics Platform and the necessary operators to control the execution progress are described.

Project 1. Automated house.

This project provides the following actions:

- After pressing the button, the door opens;
- After identifying the presence of Engino-man, the door closes;
- While the presence of Engino-man is detected, the fan runs, the LED light is on.

Figure 1 shows the blocks included in the Flow Diagram and the setting of the ports with the ERP Simulator – port A is the door open/close motor; port B is for fan drive; port 1 is LED light; port 2 is the infrared object detection sensor; port 4 is a touch sensor.

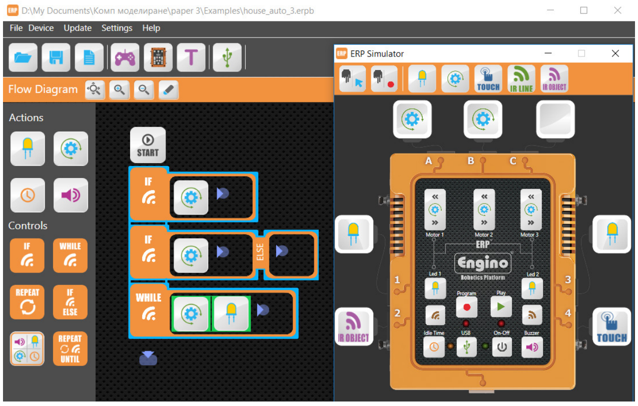
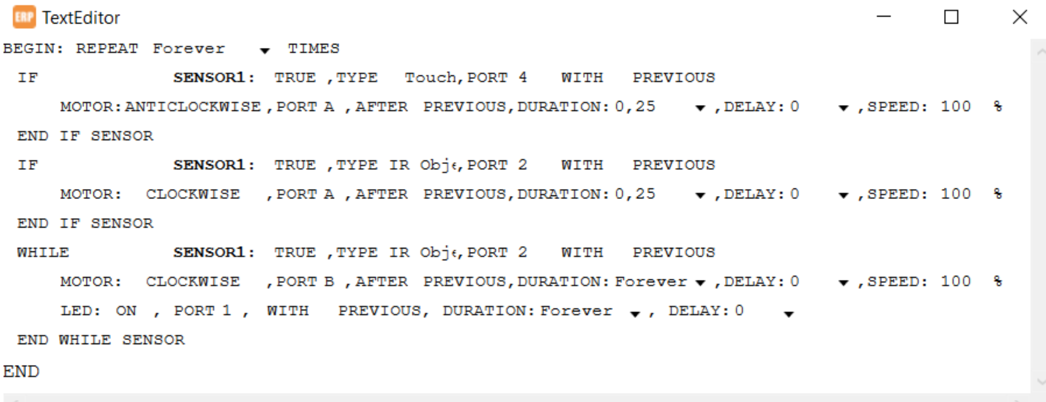


Figure 1. Project „Automated house"

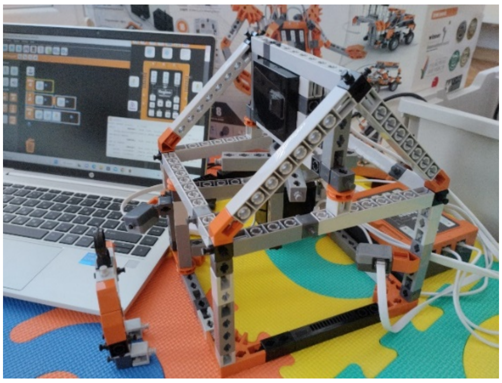
Figure 2 depicts the generated textual description of the blocks with the specified actions and parameters to execute them on. Two conditional `if` statements are used – the first statement checks whether the touch button is pressed in order to open the door; the other one – whether the infrared object sensor detects presence to close the door after Engino-man enters; a loop statement `while` that monitors the result of the infrared object sensor to drive the fan and turn on the LED light or not.



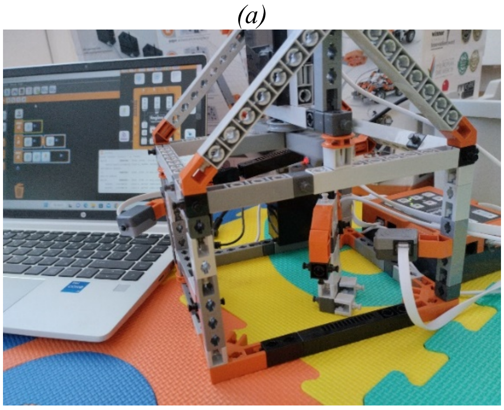
```
TextEditor
BEGIN: REPEAT Forever  TIMES
 IF          SENSOR1: TRUE ,TYPE  Touch,PORT 4   WITH  PREVIOUS
    MOTOR:ANTICLOCKWISE ,PORT A ,AFTER PREVIOUS,DURATION:0,25  ,DELAY:0  ,SPEED: 100 %
 END IF SENSOR
 IF          SENSOR1: TRUE ,TYPE IR Obj,PORT 2   WITH  PREVIOUS
    MOTOR:  CLOCKWISE  ,PORT A ,AFTER PREVIOUS,DURATION:0,25  ,DELAY:0  ,SPEED: 100 %
 END IF SENSOR
 WHILE       SENSOR1: TRUE ,TYPE IR Obj,PORT 2   WITH  PREVIOUS
    MOTOR:  CLOCKWISE  ,PORT B ,AFTER PREVIOUS,DURATION:Forever ,DELAY:0  ,SPEED: 100 %
    LED: ON , PORT 1 , WITH  PREVIOUS, DURATION:Forever , DELAY:0
 END WHILE SENSOR
END
```

Figure 2. Text description of the blocks for the project „Automated house"

Figure 3 illustrates the execution of the implemented project – before (*a*) and after (*b*) Engino-man entering the house.



(a)



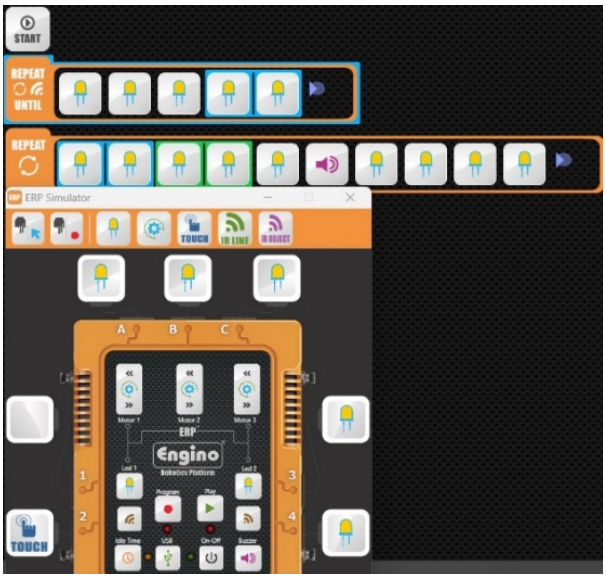
(b)

Figure 3. Project execution „Automated house"

<u>Project 2</u>. Pedestrian crossing.

This construction includes a touch sensor (port 2) and two traffic lights (Fig. 4):

- One for vehicles – the LED lights connected to ports A (green), B (yellow), C (red);
- Another for pedestrians – the LEDs connected to ports 3 (red) and 4 (green).

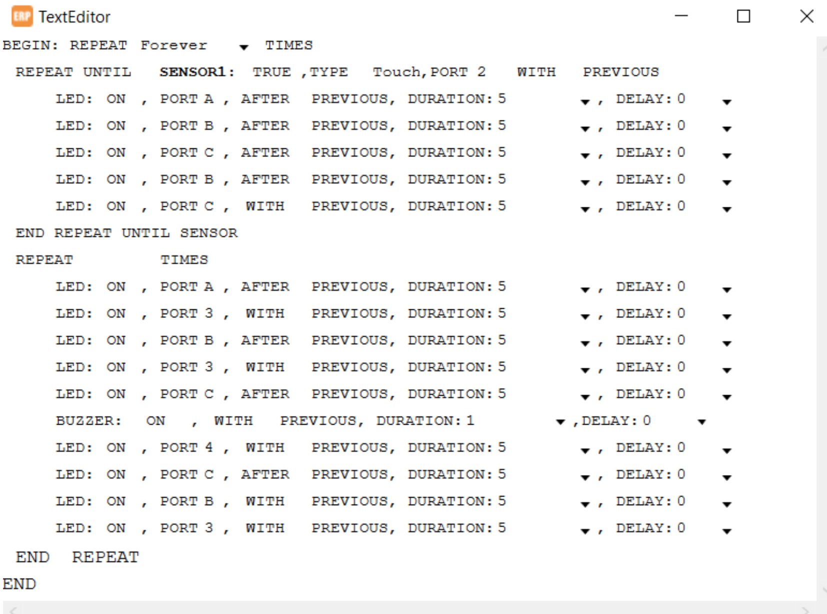


(a)

(b)

Figure 4. Project „Pedestrian crossing"

The program blocks and their parameters must ensure the operation of the traffic lights and their synchronization after pressing the button (Figs. 4 and 5).



```
TextEditor
BEGIN: REPEAT Forever   TIMES
 REPEAT UNTIL   SENSOR1:  TRUE ,TYPE   Touch,PORT 2   WITH   PREVIOUS
    LED:  ON  , PORT A , AFTER   PREVIOUS, DURATION: 5    , DELAY: 0
    LED:  ON  , PORT B , AFTER   PREVIOUS, DURATION: 5    , DELAY: 0
    LED:  ON  , PORT C , AFTER   PREVIOUS, DURATION: 5    , DELAY: 0
    LED:  ON  , PORT B , AFTER   PREVIOUS, DURATION: 5    , DELAY: 0
    LED:  ON  , PORT C ,  WITH   PREVIOUS, DURATION: 5    , DELAY: 0
 END REPEAT UNTIL SENSOR
 REPEAT         TIMES
    LED:  ON  , PORT A , AFTER   PREVIOUS, DURATION: 5    , DELAY: 0
    LED:  ON  , PORT 3 ,  WITH   PREVIOUS, DURATION: 5    , DELAY: 0
    LED:  ON  , PORT B , AFTER   PREVIOUS, DURATION: 5    , DELAY: 0
    LED:  ON  , PORT 3 ,  WITH   PREVIOUS, DURATION: 5    , DELAY: 0
    LED:  ON  , PORT C , AFTER   PREVIOUS, DURATION: 5    , DELAY: 0
    BUZZER:   ON   ,  WITH   PREVIOUS, DURATION: 1    ,DELAY: 0
    LED:  ON  , PORT 4 ,  WITH   PREVIOUS, DURATION: 5    , DELAY: 0
    LED:  ON  , PORT C , AFTER   PREVIOUS, DURATION: 5    , DELAY: 0
    LED:  ON  , PORT B ,  WITH   PREVIOUS, DURATION: 5    , DELAY: 0
    LED:  ON  , PORT 3 ,  WITH   PREVIOUS, DURATION: 5    , DELAY: 0
 END  REPEAT
END
```

Figure 5. Text description of the blocks for the project „Pedestrian crossing"

The implementation of the project consists of two loop statements `repeat` (Fig. 5):

- The body of the first loop operator executes until information is received from the touch sensor for a pressed button and ensures the operation of the traffic light for vehicles by successively flashing green, yellow, red and finally both yellow and red lights;
- The second loop `repeat` statement executes continuously, adding the synchronous operation of the pedestrian traffic light.

For this purpose, the parameter `with previous` indicates the parallel execution of the actions – for example, a green light at the traffic light for vehicles at the same time as a red light at the traffic light for pedestrians, etc. In addition to flashing a green light, an audible signal notifies pedestrians when they are allowed to cross.

<u>Project 3</u>. Humanoid robot: Avoiding obstacles.

The humanoid robot possesses infrared sensors for obstacle detection – port 2 and port 4; the motors providing the movement of the wheels are connected to port A and port B; LED lights – port 1 and port 3 (Fig. 6).

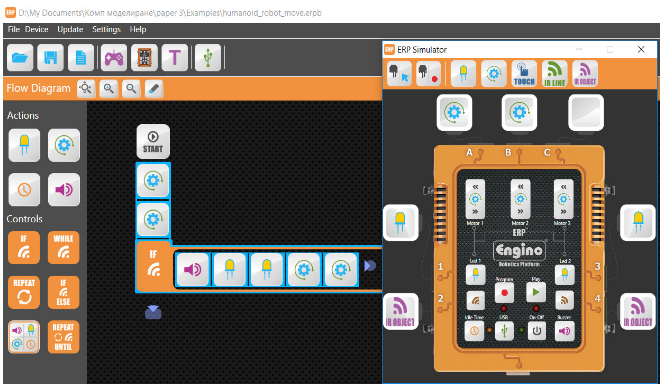


Figure 6. Project „Avoiding obstacles"

The first two blocks move the robot forward (Fig. 7). The conditional statement `if` checks the data received from infrared sensors and if any of them indicate the presence of an obstacle, the robot beeps, lights up and turns to avoid the collision.

```
TextEditor
BEGIN: REPEAT Forever   TIMES
   MOTOR:  CLOCKWISE   ,PORT A , WITH  PREVIOUS,DURATION:Forever ,DELAY:0   ,SPEED: 100  %
   MOTOR:ANTICLOCKWISE ,PORT B , WITH  PREVIOUS,DURATION:Forever ,DELAY:0   ,SPEED: 100  %
 IF           SENSOR1:  TRUE ,TYPE IR Obj,PORT 2   WITH   PREVIOUS
   OR         SENSOR2:  TRUE ,TYPE IR Obj,PORT 4
    BUZZER:  ON  , AFTER  PREVIOUS, DURATION:0,25    ,DELAY:0
    LED:  ON  , PORT 1 ,  WITH   PREVIOUS, DURATION:1     , DELAY:0
    LED:  ON  , PORT 3 ,  WITH   PREVIOUS, DURATION:1     , DELAY:0
    MOTOR:  CLOCKWISE  ,PORT A , WITH  PREVIOUS,DURATION:0,50   ,DELAY:0   ,SPEED: 100  %
    MOTOR:  CLOCKWISE  ,PORT B , WITH  PREVIOUS,DURATION:0,50   ,DELAY:0   ,SPEED: 100  %
 END IF SENSOR
END
```

Figure 7. Text description of the blocks for the project „Avoiding obstacles"

Figure 8 depicts the implementation of the realized project 3 – the humanoid robot identifies an obstacle in front of it and reacts in the indicated way.

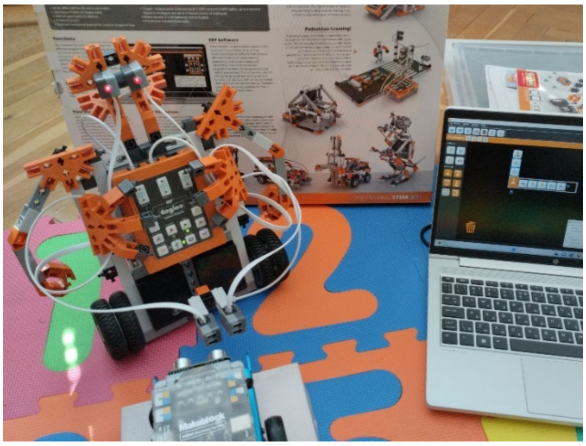


Figure 8. Project execution „Humanoid robot: Avoiding obstacles"

Project 4. Humanoid robot: Line following.

In this variant, the humanoid robot uses the infrared sensors to follow a line (Fig. 9).

Figure 9. Project „Line following"

While the two infrared line-following sensors provide information (Fig. 10) that there is a black line in front of them, the two motors drive the wheels and the robot moves forward; if any of the infrared line following sensors detect the presence of a white line, the corresponding motor will not drive the wheel and the robot makes a turn.

```
TextEditor
BEGIN: REPEAT Forever   TIMES
 WHILE          SENSOR1: FALSE,TYPE IR Line,PORT 4   WITH   PREVIOUS
     MOTOR:  CLOCKWISE   ,PORT A ,AFTER  PREVIOUS,DURATION:Forever ,DELAY:0   ,SPEED: 100  %
 END WHILE SENSOR
 WHILE          SENSOR1: FALSE,TYPE IR Line,PORT 2   WITH   PREVIOUS
     MOTOR:ANTICLOCKWISE,PORT B ,AFTER  PREVIOUS,DURATION:Forever ,DELAY:0   ,SPEED: 100  %
 END WHILE SENSOR
END
```

Figure 10. Text description of the blocks for the project „Line following"

Project 5. Grabber robot: Obstacles removal.

The project consists of programming the grabber robot to move forward until it collides with an object; then grabs the object, sets it aside, and continues its movement in the direction originally followed.

For this purpose, the robot uses the motors providing its movement, connected to port A and port B; a motor that grabs and releases objects (port C); touch sensor (port 3) as shown in Figure 11.

Figure 11. Project „Obstacles removal"

The conditional statement `if` is used for verifing whether the robot has encountered an obstacle based on the data from the touch sensor (Fig. 12). If the condition is satisfied, constructs are executed that specify the following actions: the robot stops its movement; make a sound; shrinks its grips; rotates; relaxes its grips; performs a short backward movement (to avoid colliding with the object again); rotates in the opposite direction so that it continues its original motion.

```
TextEditor
BEGIN: REPEAT Forever    TIMES
   MOTOR:  CLOCKWISE   ,PORT A ,AFTER PREVIOUS,DURATION:Forever ,DELAY:0   ,SPEED: 20  %
   MOTOR:ANTICLOCKWISE,PORT B , WITH  PREVIOUS,DURATION:Forever ,DELAY:0   ,SPEED: 20  %
 IF           SENSOR1:  TRUE ,TYPE   Touch,PORT 3    WITH   PREVIOUS
    MOTOR:     OFF      ,PORT A ,AFTER PREVIOUS,DURATION:1        ,DELAY:0   ,SPEED: 100  %
    MOTOR:     OFF      ,PORT B , WITH PREVIOUS,DURATION:1        ,DELAY:0   ,SPEED: 100  %
    BUZZER:  ON   , AFTER   PREVIOUS, DURATION:1          ,DELAY:0
    MOTOR:ANTICLOCKWISE,PORT C ,AFTER PREVIOUS,DURATION:1,75     ,DELAY:0   ,SPEED: 100  %
    MOTOR:ANTICLOCKWISE,PORT A ,AFTER PREVIOUS,DURATION:3        ,DELAY:0   ,SPEED: 100  %
    MOTOR:  CLOCKWISE   ,PORT C ,AFTER PREVIOUS,DURATION:1,75     ,DELAY:0   ,SPEED: 100  %
    MOTOR:ANTICLOCKWISE,PORT A ,AFTER PREVIOUS,DURATION:2        ,DELAY:0   ,SPEED: 100  %
    MOTOR:  CLOCKWISE   ,PORT B , WITH PREVIOUS,DURATION:2        ,DELAY:0   ,SPEED: 100  %
    MOTOR:  CLOCKWISE   ,PORT A ,AFTER PREVIOUS,DURATION:1,75     ,DELAY:0   ,SPEED: 100  %
 END IF SENSOR
END
```

Figure 12. Text description of the blocks for the project „Obstacles removal"

4. Conclusion

Educational robots provide real opportunities for students to apply theoretical knowledge in practical scenarios, combining learning with entertainment. Specifically, programming educational robots for real-world scenarios is a valuable and important tool for education that encourages students to develop multiple skills necessary for their successful futures. In addition to the already stated benefits, the following can be highlighted:

- Active learning;

Students engage in an active and hands-on learning process by programming robots to perform specific tasks. As a result, the development of problem-solving skills and analytical thinking are supported.

- Collaboration and communication.

In many cases, programming robots requires collaboration between students. They must communicate and cooperate to achieve a common goal. Students can represent their solutions and projects to others, which improves their communication and presentation skills.

Simulating real scenarios and situations through educational robots provides opportunities for students to experiment with realistic circumstances. Program implementations of the projects automated house, pedestrian crossing, robot movement with obstacle avoidance and removal, line following via Engino Robotics Platform are available at https://github.com/tsvetankageorgieva/engino.

References:

[1]. Zecca, L. (2021). The game of thinking. Interactions between children and robots in educational environments. In *Makers at School, Educational Robotics and Innovative Learning Environments: Research and Experiences from FabLearn Italy 2019, in the Italian Schools and Beyond*, 87-94. Cham: Springer International Publishing.

[2]. Xerou, E., & Angeli, C. (2022). How can robotics and programming affect children's computational thinking skills through real life scenarios?, In *Proceedings of the 14th International Conference on Education and New Learning Technologies,* 9153-9159. Doi: 10.21125/edulearn.2022.2197.

[3]. Velikova, E., Mierlus-Mazilu, I., Vasileva-Ivanova, R., & Georgieva, D. (2018). About the stem Education. *Proceedings of University of Ruse*, *57*, 10-14.

[4]. Georgieva, I. M., & Dochkova-Todorova, J. (2010). Semantic descriptions of games in mathematical educational sites for children of preschool and primary school age. In *Proceedings of the Scientific conference "25 years Faculty of Pedagogy",* 435-439.

[5]. Georgieva, I. M., & Dochkova-Todorova, J. (2013). Educational Mathematical Internet Games for Pupils of Preschool and Primary School Age. In *Proceedings of the MASSEE International Congress on Mathematics,* 71-77.

[6]. Saatcioğlu, K. T., & Boru, B. (2015). Using Educational Robotics for Students with Learning Difficulties. In *Proceedings of the 3rd International Symposium on Innovative Technologies in Engineering and Science*, 2152-2160.

[7]. Sisamou, E. (2017). Adaptation of a research tool in Greek language which measures teachers' perspectives about integrating robotics in primary education to enhance stem teaching and learning: evaluation of validity and reliability of the tool. In *Proceedings of the International Conference on Information, Communication Technologies in Education*, 439-448.

[8]. Paspallis, N., Polycarpou, I., Andreou, P., Antoniou, J., Kaimakis, P., Raspopoulos, M., & Terzi, M. (2018). An experience report on the effectiveness of five themed workshops at inspiring high school students to learn coding. In *Proceedings of the 23rd annual ACM conference on innovation and technology in computer science education*, 105-110. Doi: 10.1145/3197091.3197093.

[9]. Evripidou, S., Georgiou, K., Doitsidis, L., Amanatiadis, A. A., Zinonos, Z. & Chatzichristofis, S. A. (2020). Educational Robotics: Platforms, Competitions and Expected Learning Outcomes. *IEEE Access, 8,* 219534-219562. Doi: 10.1109/ACCESS.2020.3042555.

[10]. Hovardas, T., Xenofontos, N., Pavlou, Y., Kouti, G., Vakkou, K., & Zacharia, Z. (2021). Pedagogical design in stem education bridging educational robotics, game-based learning and inquiry-based learning: insights from a bundle of lesson plans building on the inclined plane. In *Proceedings of the 15th International Technology, Education and Development Conference*, 4364-4371. Doi: 10.21125/inted.2021.0890.

[11]. Almpani, S., & Almisis, D. (2021, February). Dance and Robots: Designing a Robotics-Enhanced Project for Dance-Based STEAM Education Using ENGINO. In *Educational Robotics International Conference*, 139-151. Cham: Springer International Publishing.

[12]. Zerega, R., Hamidi, A., Tavajoh, S., & Milrad, M. (2022). A Robotic-based Approach for CT Development: Challenges of Teaching Programming Concepts to Children and the Potential of Informal Learning. In *Proceedings of the International Conference on Computational Thinking Education and STEM Education,* 12-17. Doi: 10.34641/ctestem.2022.456.

[13]. Sisamou, E., & Eteokleous, N. (2023). A pedagogical framework and Evaluation of STEM & Robotics Camp Programs. A case-study. *International Scientific Educational Journal, 11*(1), 45-67.

[14]. Östlund, B., Malvezzi, M., Frennert, S., Funk, M., Gonzalez-Vargas, J., Baur, K., Alimisis, D., Thorsteinsson, F., Alonso-Cepeda, A., Fau, G., Haufe, F., Di Pardo, M., & Moreno, J.C. (2023). Interactive robots for health in Europe: Technology readiness and adoption potential. *Frontiers in Public Health, 11,* 979225. Doi: 10.3389/fpubh.2023.979225.

[15]. Demetriou, G. A., Lambrou, A., Eteokleous, N., & Sisamos, C. (2013). The Engino Robotics Platform (ERP) controller for education. In *Proceedings of the 21st Mediterranean Conference on Control and Automation,* 567-572. Doi: 10.1109/MED.2013.6608778.