

Article

# Application of Basic Graph Theory in Autonomous Motion of Robots

Petr Coufal <sup>1,\*</sup> , Štěpán Hubálovský <sup>1</sup>  and Marie Hubálovská <sup>2</sup>

<sup>1</sup> Faculty of Science, University of Hradec Kralove, Rokitanskeho 62, 50003 Hradec Kralove, Czech Republic; stepan.hubalovsky@uhk.cz

<sup>2</sup> Faculty of Education, University of Hradec Kralove, Rokitanskeho 62, 50003 Hradec Kralove, Czech Republic; marie.hubalovska@uhk.cz

\* Correspondence: petr.coufal@uhk.cz

**Abstract:** Discrete mathematics covers the field of graph theory, which solves various problems in graphs using algorithms, such as coloring graphs. Part of graph theory is focused on algorithms that solve the passage through mazes and labyrinths. This paper presents a study conducted as part of a university course focused on graph theory. The course addressed the problem of high student failure in the mazes and labyrinths chapter. Students' theoretical knowledge and practical skills in solving algorithms in the maze were low. Therefore, the use of educational robots and their involvement in the teaching of subjects in part focused on mazes and labyrinths. This study shows an easy passage through the individual areas of teaching the science, technology, engineering, and mathematics (STEM) concept. In this article, we describe the research survey and focus on the description and examples of teaching in a university course. Part of the work is the introduction of an easy transition from the theoretical solution of algorithms to their practical implementation on a real autonomous robot. The theoretical part of the course introduced the issues of graph theory and basic algorithms for solving the passage through the labyrinth. The contribution of this study is a change in the approach to teaching graph theory and a greater interconnection of individual areas of STEM to achieve better learning outcomes for science students.

**Keywords:** graph theory; algorithm; maze; robot; education



**Citation:** Coufal, P.; Hubálovský, Š.; Hubálovská, M. Application of Basic Graph Theory in Autonomous Motion of Robots. *Mathematics* **2021**, *9*, 919. <https://doi.org/10.3390/math9090919>

Academic Editor: David Pugalee

Received: 28 February 2021

Accepted: 19 April 2021

Published: 21 April 2021

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

This paper focuses on a new concept for teaching graph theory using educational robots, so it is a connection of the individual parts of the science, technology, engineering, and mathematics (STEM) concept. This article describes research conducted when teaching graph theory at a university and provides a practical solution to a selected part of the course focused on solving a passage through a maze. Students were prepared with teaching in the theoretical field of discrete mathematics with a focus on the field of graph theory. As part of the educational course for theoretical lectures, they also attended practical seminars where students learned to apply the acquired theoretical knowledge to examples. Unfortunately, most of the examples solved with the application of theoretical knowledge were solved using pencil on paper. Students did not have the opportunity to transfer theoretical knowledge to solve real problems in life. In this case, we present a change in a practical seminar that involved educational robots in the teaching of graph theory. Here, we see considerable potential for their use and fulfilment of the STEM concepts.

## 2. Methodology

### 2.1. Aims

The main goal of this study was to determine the benefits of a new concept of teaching graph theory using educational robots. The aim was to connect the individual areas of STEM and use their strengths for the development of students to improve their ability to

practically apply theoretical knowledge. Another goal was to increase the attractiveness of the classical teaching of discrete mathematics and to improve its practicality with an overlap with the real world. Students in the course worked in groups on a joint project with an educational robot. In this study, we also focused on the communication and joint work of students. The teaching of students during graph theory was conducted using old methods; low motivation of students was recorded; and, in some areas, high failure rates of students were recorded. The following were the parameters of the course: course objectives: the aim was to develop and increase the logical and algorithmic thinking of students, to make them familiar with basic graph theory concepts and theorems and combinatorial algorithms; course content: basic concepts, special types of graphs, graph coloring, trees, mazes and labyrinths, searching for Eulerian graphs, graph covering, graph searching, searching trees, and using graph searching in algorithms.

The course aimed to address the problem of high student failure in the mazes and labyrinths chapter. Students' theoretical knowledge and practical skills in solving algorithms in the maze were low. The overall test failure in this chapter was high, with a 35% failure rate on the first test.

## 2.2. Theoretical Background

To address the high failure rate of students, a solution was designed using educational robots. Educational robots enable the easy application of selected algorithms from graph theory. The study focused on algorithms that solve the passage through a maze or a labyrinth. A maze or labyrinth can be easily redrawn into a graph. Each vertex of the graph represents an intersection in the labyrinth and the individual edges of the graph represent paths within the labyrinth. A labyrinth can embody, e.g., a part of a city where a route is sought from a hotel to a university [1,2]. What is the difference between a labyrinth and a maze? Labyrinths contain a path without turns and crossroads, where the path leads from the start to the finish. Therefore, the path through a labyrinth is easy. A maze is composed of more paths and intersections and there are various turns and dead ends. In a maze, several different paths often lead to the end. The journey in a maze can be long. The well-known tips for walking through the maze are to stick to the wall with one hand, or to mark the paths or intersections we have already traveled. An example of the latter is the historically well-known solution using Ariadne's thread [1–3].

Many mathematicians and scientists have created various algorithms within graph theory to navigate through a labyrinth or maze. In some cases, it is necessary to go through the whole labyrinth, i.e., to visit all corridors. Examples of the algorithms that solve this problem are Trémaux [4], Tarry's algorithm [5], or the Edmund-Johnson algorithm [6].

Tarry's algorithm is based on two rules:

Rule T1: Each edge can be passed only once in a single direction (if a labyrinth is without an exit, then exactly once).

Rule T2: The path used to enter the vertex cannot be used for return, except if there is no other option [5,7].

This algorithm finds the path from the entrance to the exit of the labyrinth. If a labyrinth does not have an exit, all paths will be traveled twice, i.e., in one direction when searching for an exit and again when returning to the entrance. The Trémaux algorithm adds a third rule:

Rule T3: If the vertex is reached by the edge used the first time, if we come to an edge that is traversed for the first time, to a known vertex, the edge is used for return in the next step [4,7].

A variety of different algorithms focus on solving the passage through a maze. These algorithms can be divided according to their approach to a maze. The aforementioned algorithms are based on the knowledge of the whole maze in the form of a graph. These algorithms can be used by a human or a computer program. However, the robot in the maze experiences different conditions as it only perceives what its sensors detect, such

as a wall of the maze or a free path. Blind paths and loops gradually appear as it passes through the maze.

Other algorithms do not operate on a complete maze map. They create this map during passing through a maze or they do not use it at all. In the case of the random mouse algorithm, the robot moves completely randomly. This group of algorithms includes pledge algorithm, flood fill algorithm, recursive backtracker, and so on [8].

### 2.3. Research Questions

We formulated the following research questions for this study: (1) What is the contribution of educational robots to the practical teaching of graph theory? (2) How does group work affect students in solving algorithms for the robot to pass through a maze? (3) How is it possible to transfer teaching from educational robots to the online environment?

One of the partial goals of the study was to present the possibilities of the simple implementation of algorithms for solving the passage through a maze using a real robot in a real maze. Another partial goal was to present the direct transition from real robots and their algorithms to a virtual environment, which can then be used in distance learning. Algorithms can be implemented in various programming environments, where, in our case, a visual programming environment is presented. Visual programming is not only suitable for beginners, but is also very widespread. This is the default programming environment for the Lego Mindstorms robotic kit that was used to construct the robot.

### 2.4. Research Tools and Sample

As part of the research survey in this study, a quantitative questionnaire was used to evaluate teaching for students. The questionnaire was created for the purposes of this training course. The questionnaire contained questions focused on this course; the student chose the answer based on the choice of options according to a Likert scale. The Likert scale was used to evaluate the attitudes toward a given statement in the questionnaire. In the next part of the research survey, qualitative methods were used: direct observation and semi-structured interviews with selected students. The group work of students on their joint projects was monitored directly with a focus on their interactivity in the group. Semi-structured interviews were conducted at the end of the course to retrospectively ascertain the attitudes of students to the teaching of graph theory using educational robots. A group of university students of natural sciences was included in the research survey. Specifically, these were students of science teaching in combination with computer science. Students had graduated and knew the following: the basics of university mathematics, the basics of programming, and microcomputer programming. At the same time, students attended a course in programming robotic kits and robots.

### 2.5. Course Implementation

This part of the paper presents, within the framework of our practical research, the implementation of teaching in the practical seminars on the subject of graph theory. In this section, we provide a theoretical introduction to graph theory, which belongs to discrete mathematics, and describe the robot and maze used, an important part in the implementation of graphs and their practical evaluation and comparison. We also present the possibilities of transferring teaching to the online environment.

Students implemented the theoretical knowledge gained in their graph theory lectures on a real maze. The theory contains a large number of algorithms for solving the passage through a maze. However, it is not possible to implement all of them using a robot that passes through a maze and sees only the walls around it. The active method of project-based student learning in groups facilitates the development of their competencies [9,10]. The combination of project-based learning with the construction of a robot model in the field of STEM supports the cooperation and mutual communication of students in a group [11,12]. For example, for inspiration and recognition of science and technology

(FIRST), the program uses project-based learning for the solution of real problems with robots in the field of STEM [13].

In our study, students first designed a suitable robot that met the minimum requirements and then sought a solution to the robot's passage through the maze. In the preparatory and planning phase, students designed various solutions for the construction of the robot, which they further tested in a real maze. The construction and manual skills of individual students were manifested here. During model construction work, robots developed their competencies, learnt new skills, and gained new knowledge [14,15]. The subsequent phase was the selection, design, and implementation of an algorithm for the robot. This part proved to be challenging for the students because they failed to create all the selected algorithms for the robot to travel through the maze in the programming environment. There were also other problems, such as robot sensors, software restrictions on variables and fields, and so on. The most entertaining part for the students was testing the robot in a real maze. Frequent modifications were necessary during the testing of selected algorithms based on the real data obtained in the maze as well as adjusting the setting of sensors and the individual movements of the robot [16].

The final stage was testing and comparing the selected algorithms. The individual areas of the STEM concept are interconnected during the implementation of the algorithm into real movements of an autonomous robot moving in a maze [13,17]. Despite little experience with the construction of robots and the implementation of algorithms in the programming environment, students managed to create suitable robots and compare various selected algorithms for passing through the maze. While cooperating on an autonomous robot project, students learned teamwork and communication, and together they designed and applied solutions for suitable algorithms for the robot to pass through a maze [15,18].

### 2.6. Maze Design

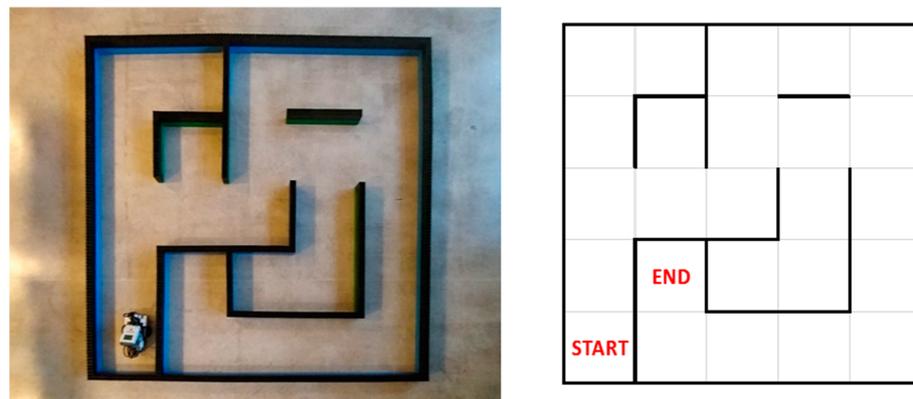
Two extended approaches were used to construct a maze for this study.

The first approach was based on the creation of the individual walls of the maze from an opaque material that separated the individual paths within the maze. Materials that could be used for the construction of the walls of the maze were wood, paper, or plastic. The height of the walls, the type of construction, and the used material were determined. The floor of the maze could be a table top, a uniform classroom floor, or a large enough piece of paper. In some cases, a color designation of the start and finish fields of the maze was used. Furthermore, a maze constructed of square fields of the same size could be used, where all paths were of the same width. In contrast, the version of a maze with variable widths of paths and different locations of the intersections was used less frequently. The construction of robots solving mazes with walls used sensors that detect walls. The basic sensor was a touch sensor, which detected the wall when the wall came into contact with the sensor. More sophisticated ultrasonic sensors were used to detect objects and measure the robot distance from them. Some robots used optical sensors to measure and detect objects, such as an electro-optical proximity detector (EOPD) sensor [19–21].

The second approach was based on a conversion of the maze into individual paths, which were established by black lines placed on a light surface. In this solution, color coding of individual intersections was sometimes adopted. This version is easy to implement with the use of black adhesive tape placed on a light floor. The construction of robots solving these types of mazes was based on optical sensors monitoring the black line. The operation of multiple sensors placed in a row next to each other allowed quick recognition of the shape of the intersection, which expedited the movement of the robot in the maze [22].

Implementing the first approach to maze construction is more difficult, but the output is a real model of a maze. The second approach is significantly simpler and more affordable. The first variant of the maze construction was selected for the presented research. Plastic pieces of children's construction kits were used for the construction of walls. Having a large number of these pieces available allows the construction of almost any maze. The

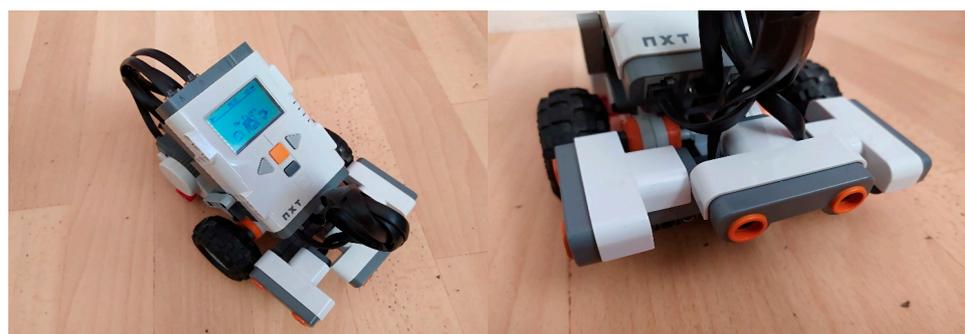
walls were built of four layers of building blocks and their height was approximately 10 cm. The width of the maze path was designed so that it allowed easy movement of the robot, especially the rotation of the robot on the spot. The construction of the maze was based on a square  $30 \times 30$  cm network. The width of the square was equal to the width of the path. The wall of the maze was 3 cm thick. The maze was designed with a size of  $5 \times 5$  squares. The layout of the individual walls is shown in Figure 1. The starting and end squares of the maze are highlighted in the scheme.



**Figure 1.** The maze was constructed using children’s kits; the scheme of the maze.

### 2.7. Construction of the Robot

The Lego Mindstorms kit was used for the construction of the robot for this study. The kit contains a wide range of components that allow various robot designs. The basic construction of the robotic vehicle was extended by a sensor array for our intentions. The construction of the robotic model of the vehicle contains two motors that enable the movement of the robot. The robot’s chassis consists of two wheels connected directly to the motors and a rear uncontrolled wheel that can move freely. Placing the powered wheels in the center of the robot allows the robot to rotate in place. In front of the robot, there are three ultrasonic sensors used for object detection. In our case, we used ultrasonic sensors to detect the walls of the maze. Therefore, one sensor was turned to the right, the second sensor was facing forward, and the third sensor faced the left side. The ultrasonic sensors were connected to the input ports of the control unit under numbers 1, 2, and 3. The location of the ultrasonic sensors was determined by the dimensions of the maze, the height of the walls, and the width of the path. The sensors were located at the front bottom of the robot structure. This location of the ultrasonic sensors allowed the robot to examine intersections. The main part of the robot’s construction was the control unit, which connected two motors and three sensors. Figure 2 shows the construction of the robot, the location of the sensors, the control unit, and the motors providing the movement and rotation of the robot [20,21,23].



**Figure 2.** The robot model used three ultrasonic sensors in the front.

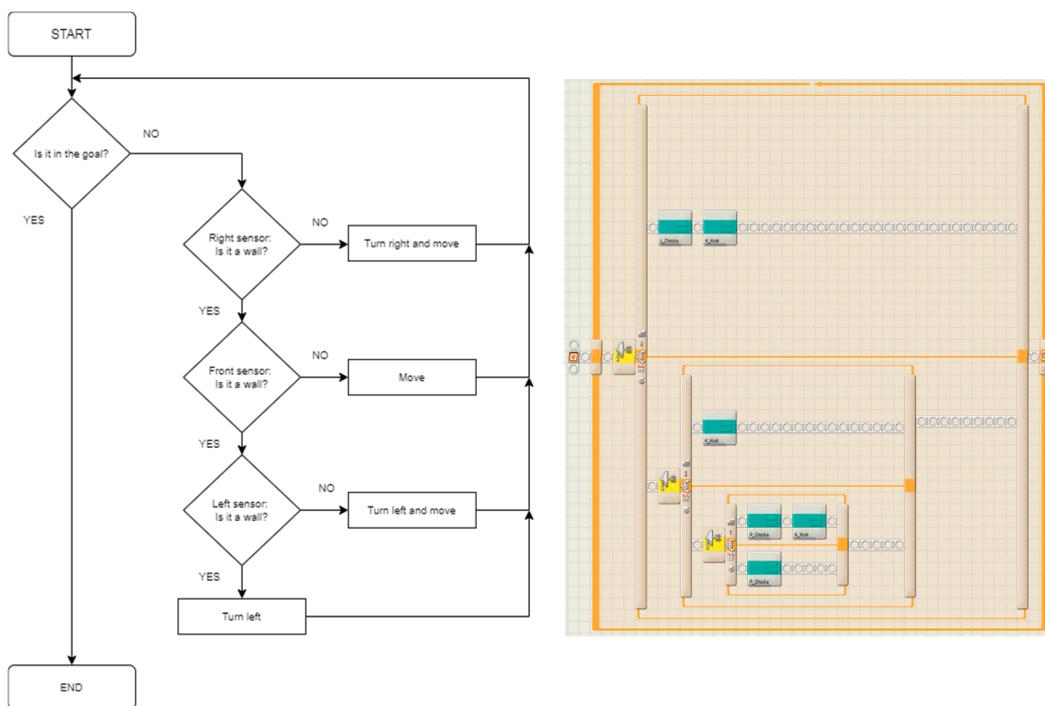
### 2.8. Programming the Robot

Different programs using various programming languages can be used to program and control the assembled robot, including the original Lego Mindstorms programming environment, which was also used in our case. Lego Mindstorms enables simple visual programming. This programming environment is easy to understand and is suitable for beginners. The main program consists of individual blocks arranged one after another using a single control algorithm. The parameters of the robotic or programming elements are set in each icon, e.g., the ports of the connected motors, speed, direction, and rotation time are set as a part of the motor icon [24].

### 2.9. Selection of Algorithms

The algorithms for passing through the maze had to correspond to those given by the selected construction of the robot and the programming environment. Therefore, the algorithms that used a view of the maze or its map could not be used. The robot must gradually explore the maze and look for a suitable path. The robot could not mark already visited intersections and paths in the maze or remember them. Therefore, algorithms that do not use this approach had to be used.

The first such algorithm is wall monitoring, which implements both options, i.e., monitoring the right wall and the left wall of the maze. The algorithm for right wall monitoring is shown in Figure 3. The algorithm for left wall monitoring is similar except the order of the sensors is changed in the diagram as well as the subsequent rotation of the robot on the selected side. Figure 3 shows the implementation of this algorithm in a visual programming environment using subroutines for the rotation and movement of the robot [25,26].



**Figure 3.** Right wall monitoring flowchart and left wall monitoring implementation in a visual programming environment, Lego Mindstorms NXT.

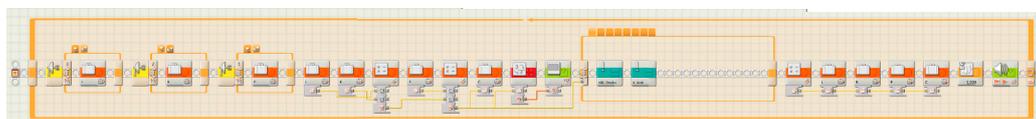
Another possible algorithm is random mouse. The algorithm, as the name implies, uses chance in deciding which path the robot should take when moving in a maze. The random mouse algorithm first determines if individual sensors detect a wall, then the robot completely randomly selects one of the open paths in the maze. When detecting an end of a path, when all sensors detect the wall, the robot rotates 180° [27]. When implementing the algorithm in a visual programming environment, it is necessary to use a decision table

that determines from which open paths the robot randomly selects the path of the next movement. Table 1 shows the shape of the intersection in the first column; the values for individual sensors are in the next three columns. A value of 1 indicates that the path is clear and 0 indicates a maze wall. Each value is then multiplied by the coefficient of each sensor 1, 3, and 5. The next column lists the options in which direction the robot can move. This list is used by the robot to randomly select its new direction. The last column is the sum of the coefficients of Table 1.

**Table 1.** Decision table for the random mouse algorithm. L, left; F, front; R, right.

Crossroads	Sensor			Movement	Sum
	Left (1)	Front (3)	Right (5)		
	0	0	0	Turn 180°	0
	1	1	1	L, F, R	9
	0	0	1	R	5
	0	1	0	F	3
	1	0	0	L	1
	1	1	0	L, F	4
	1	0	1	L, R	6
	0	1	1	F, R	8

The movement of the robot in the maze using the random mouse algorithm is completely random. Figure 4 shows its implementation in the Lego Mindstorms NXT visual programming environment.



**Figure 4.** Random mouse algorithm implementation in the Lego Mindstorms NXT visual programming environment.

The last algorithm used in our study is based on the pledge algorithm. The algorithm uses a selected preferred direction of movement. This algorithm uses the nested right wall monitoring algorithm in the maze. When turning to the right of the preferred direction of movement, the value of the robot’s rotation is increased by one; in the case of turning to the left, the value of the robot’s rotation is reduced by one. If the value of the robot’s rotation is set to zero, the robot continues to move again in the preferred direction. There are several pitfalls in the implementation of this algorithm, such as turning the robot more than 360° to one side. In the Lego Mindstorms visual programming environment, we defined variables that stored the current value of the robot’s rotation in the maze [26].

## 2.10. Comparison of Selected Algorithms for Solving the Robot Passing through a Labyrinth

### 2.10.1. Initial Conditions

For comparison of the selected algorithms, we used a robot assembled from a Lego Mindstorms NXT robotic construction kit with three ultrasonic distance sensors for the detection of maze walls. The maze was located on a flat, unicolored floor, which allowed the robots to move easily without the wheels slipping when turning on the spot. The maze was assembled from a children's plastic kit.

All algorithms used to solve the maze had to be implemented in the Lego Mindstorms programming environment to be usable in the virtual environment as well as for easier programming by beginner students.

### 2.10.2. Realization

The real-life times needed for passing the robot through the maze using the four aforementioned algorithms were compared. The robot began the journey on the starting field of the maze and ended the journey on the end field of the maze. The time was measured from the start of the program in the control unit to the robot's arrival at the end field of the maze. The time was measured manually on a stopwatch, which was recorded on tables.

The measurements were performed in the first version of the maze shown in Figure 1. The second measurement was performed in the same maze in the opposite direction (labelled as version 2), i.e., the start and end fields of the maze were swapped.

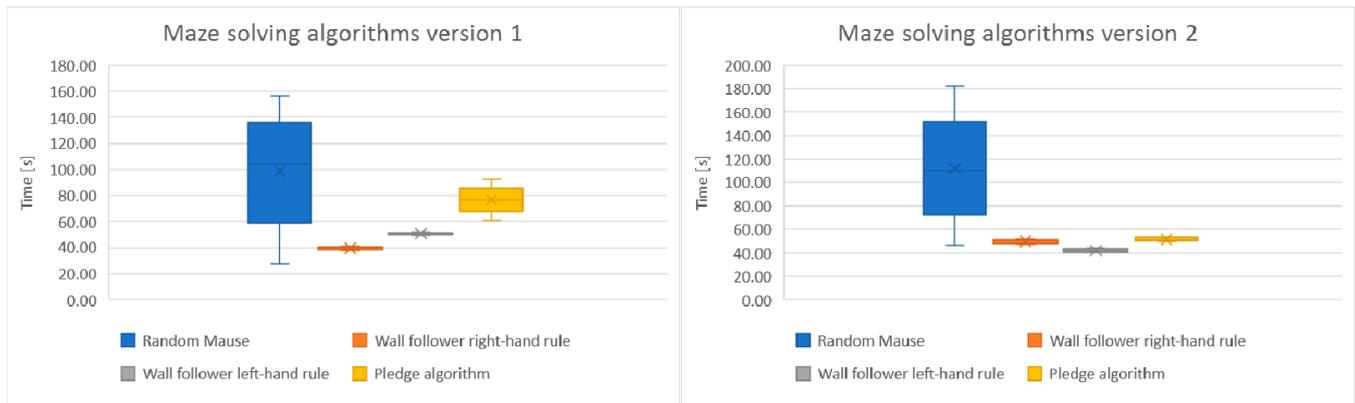
### 2.10.3. Results

The times needed for passing of the robot through the maze were statistically analyzed. The average time for the robot to pass through the maze for each algorithm is shown in Table 2. The table shows the average times for both versions of the measurement. The results revealed that the slowest algorithm was the random mouse, which uses the completely random movement of the robot in the maze. The fastest algorithm in the first version of the maze was tracking the right wall. The fastest algorithm in the opposite direction was tracking the left wall. This is because of the selected layout of the maze walls.

**Table 2.** The average time (seconds) of passing of the robot through the maze for each algorithm.

Maze	Algorithm			
	Random Mouse	Wall Following Right-Hand Rule	Wall Following Left-Hand Rule	Pledge Algorithm
version 1	101.27	39.56	50.63	76.40
version 2	109.65	49.84	42.08	51.41

Further analysis of the measured data and their division into quartiles allowed us to graphically display the results in box plots in Figure 5. The most significant area in the graph is the blue box area for the random mouse algorithm, where a large scatter of measured values is evident for different passes through the maze. Even more significant is the minimum value of this solution, which is the shortest measured time for the passing of the robot through the maze. In maze version 1, there was still significant variance in the measured values regarding the pledge algorithm (yellow box), where its shortest time was the longest overall. Other wall monitoring algorithms showed a low variance in the measured values, proving that they provided a quick solution for passing the robot through the maze.



**Figure 5.** Boxplots comparing individual algorithms in two versions of the maze.

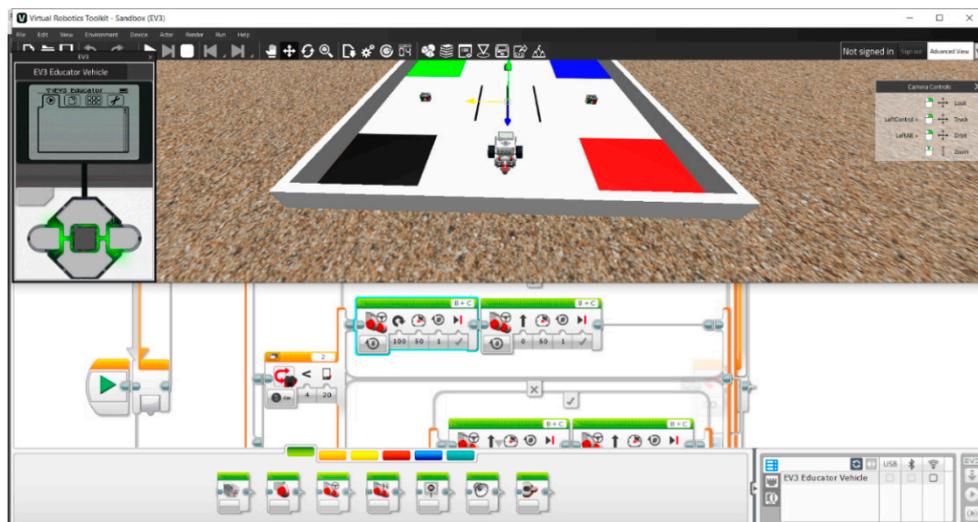
In the maze version 2, the values for the random mouse algorithm were similar. Wall monitoring algorithms provided the fastest solutions, only in reversed order. There was an interesting reduction in the scatter of values for the pledge algorithm and its overall shortening when the robot passed through maze version 2.

### 2.11. Virtual Environment

Lego Mindstorms kits can be replaced by the Virtual Robotic Toolkit program, which enables a simulation of the robot's journey according to the program created in a visual programming environment. It is possible to use either EV3 or NXT versions of the control unit; it is only necessary to use the appropriate version of the program. As the first step, the environment and then the robotic model have to be selected from the menu; consequently, the virtual robot can be controlled and programmed. All robots from Lego construction kits can be designed and assembled in a virtual environment. It is also possible to freely design and modify your own environment, e.g., a maze. The Virtual Robotic Toolkit environment uses the original Lego Mindstorms programming environment by uploading the created control program for the robot to the virtual control unit, which is automatically connected after starting the program. In the virtual environment of the program, it is possible to work completely like with a normal robot: you can freely manipulate the robot, turn it, run the control program, and so on [28]. For virtual work with robots from the Lego kit, other virtual tools can be used, such as [29,30], or robots can be connected in online robotic laboratories [31].

The virtual environment has several advantages that can be applied in teaching, e.g., each student can work with their own robot in their own maze. It is not necessary to deal with charging batteries of robots. Students can control and program robots outside the classroom, and thus they have enough time and virtual space to work with the robots, consequently solving even more demanding tasks.

The robot in the virtual environment of the Virtual Robotic Toolkit program moving in the upper right window is shown in Figure 6. The bottom of the window shows the visual programming environment in the Lego Mindstorms EV3 environment. In the left upper part of the window, there is the virtual control unit of the robot, which allows controlling and selection of programs to run. Another option is to use an online robotic laboratory, where it is possible to work with a remotely controlled real robot [31].



**Figure 6.** The virtual environment of the Virtual Robotic Toolkit.

During the global COVID-19 pandemic, this possibility of using a virtual environment for working with robots is an appropriate solution to teaching in schools. In full-time teaching, the virtual environment prevents unwanted contact of students when manipulating robots and especially when letting robots use a common maze. It would be necessary to disinfect the maze after each robot rides through it in the case of a single maze for all students. This measure would be costly and time consuming. The option for all students to have their own maze was unrealistic in our case. Another advantage is maintaining a safe distance between students in the class, because each student works only in their own location on a single computer. The student does not have to move anywhere and no one unnecessarily waits for the maze to be vacant. The disadvantage is that each student works individually and cannot follow the progress or solutions of other students. However, this can be prevented by sharing problem solving with other students in a virtual environment, which is why we support the use of the virtual environment in regular teaching in schools.

It is also possible to use the virtual environment as a part of the distance form of teaching while meeting the basic requirements of the program. It is necessary to secure licenses for all programs for school students and arrange their use on student computers. Student computers must also meet the minimum configuration requirements of the Lego Mindstorms and Virtual Robotic Toolkit programs. Another option is to use the virtual environment in a hybrid form of teaching. Students' reactions were mixed. Some preferred real robots that can be easily and quickly modified, while some appreciated the opportunity to work with their own virtual robots outside the classroom. Virtualization of work with robots is currently a necessity owing to the pandemic rather than just an alternative form of project-based learning.

### 3. Research

The aim of the research was to answer the research questions in Section 2.3.

#### 3.1. Study Results

At the end of the course, students were presented with a questionnaire containing twelve statements. Each statement contained five possible answers on a Likert scale according to the student's attitude toward the question. The questionnaire was presented and completed by a group of 15 students in paper form. Data from the paper questionnaire were digitized and further statistically processed.

First, the reliability of the questionnaire was verified by calculating the Cronbach's alpha, which is used to verify the reliability of the questionnaires, and was calculated according to the formula

$$\alpha = \frac{k}{k-1} \left( 1 - \frac{\sum_{j=1}^k \text{var}(Y_j)}{\text{var}(Y)} \right) \quad (1)$$

where  $k$  is the number of test items,  $\text{var}(Y_j)$  is the variance of the evaluation of the  $j$ th item, and  $\text{var}(Y)$  is the variance of the total scores in the test [32].

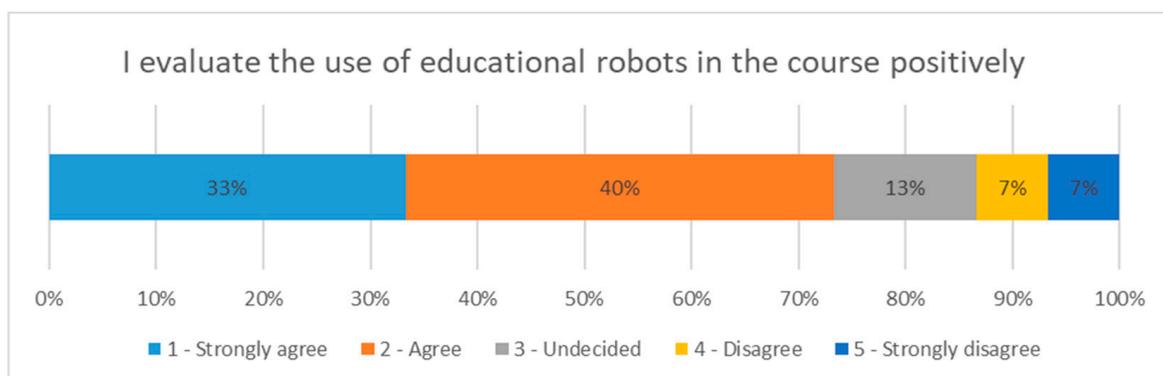
The value of Cronbach's alpha in our research for the questionnaire used was 0.84. This value is sufficient to verify the reliability of the questionnaire used. To verify the validity of the questionnaire used, content validity was used, which followed the questions in the questionnaires in relation to the research carried out during the course.

At the end of the course, a semi-structured interview with selected students was conducted. The aim of the interview was to clarify the individual attitudes of the students and to better understand and describe their interactions. The semi-structured interview also reflected students' behavior during the research observation. This qualitative research survey complemented the individual results of the questionnaire.

### 3.2. Research Question

The first research question was as follows: (1) What is the contribution of educational robots to the practical teaching of graph theory?

In the questionnaire, the first research question was addressed using five statements. The type of graph used in Figure 7 visualizes the displayed data from the Likert scale used in the questionnaire. Students evaluated their attitudes toward the first statement, with the use of educational robots in the course being answered positively in most (73%) answers. Why the use of educational robots in the course was evaluated positively was clarified in the answers from the semi-structured interview. From the answers of students, we found the following: a modern approach to teaching, the application of theory in practice, the interconnection of several disciplines, and more entertaining teaching of graph theory. Students positively perceived the connection between graph theory and the application of algorithms used in programming in real robots. We classified and generalized the answers more and found that students perceived the use of educational robots as a modern approach to teaching and positively perceived the direct application of theory in practice to solve specific problems, which was a great benefit of educational robots in this course.



**Figure 7.** Expression of students' attitudes to the statement: "I evaluate the use of educational robots in the course positively".

The use of simple visual programming of educational robots in the course facilitated a quick beginning with the direct application of algorithms to solve specific tasks. The programming environment used made it possible for complete beginners to create robot control programs. Therefore, students' attitudes to the statement "Programming and

controlling educational robots was easy” were very positive (87%; 1 = strongly agree and 2 = agree).

Students negatively evaluated the time needed at the beginning of the course to get acquainted with the educational robots, as well as to construct their maze and learning in the programming environment. They perceived that this occurred at the expense of their theoretical and practical knowledge and skills acquired in the course. Students can design their own robot; for this reason, they worked on the robot design for a long time. Students can attend further educational courses at the university focused on the construction and programming of robots.

The second research question was, (2) How does group work affect students in solving algorithms for the robot to pass through the maze?

The answer to the second research question helped us, to a large extent, to clarify the observations of students working together on a project with educational robots. During the observation, it was important to monitor and capture their mutual interactions in the group, their joint communication, division, and application of group roles. When focusing on the formation of the group, which was entirely the students’ choice, it was clear that the students’ previous acquaintances influenced their further cooperation. This was reflected in a greater degree of communication in the group, faster decision-making in solving, and greater commitment in solving tasks. For the newly formed group of students, who did not have previous mutual acquaintances, there was a long phase of getting to know each other and defining each other. This led to a lower observed performance of the group in solving the task. Another influence was the mood of the students in the group, which was variable and often closely related to the success of solving tasks. The division of roles and responsibilities in the group had a significant effect. Students often mentioned competitiveness as wanting to find better solutions and be more successful than students from another group. The social support of students in the group motivated them to become more involved even in times of failure.

Students largely agreed with the statements from the questionnaire in Figure 8: working on a joint project motivated me to be more committed (74%; 1 = strongly agree and 2 = agree). With a deeper focus on students’ answers, it was clear that the mood in the working group was also decisive. Group communication, mutual interaction, and the pursuit of a common goal influenced students in responding to another statement: “To achieve total success, I am willing to do work for other members of the group”; in this case, 33% of the answers were 3 (undecided).

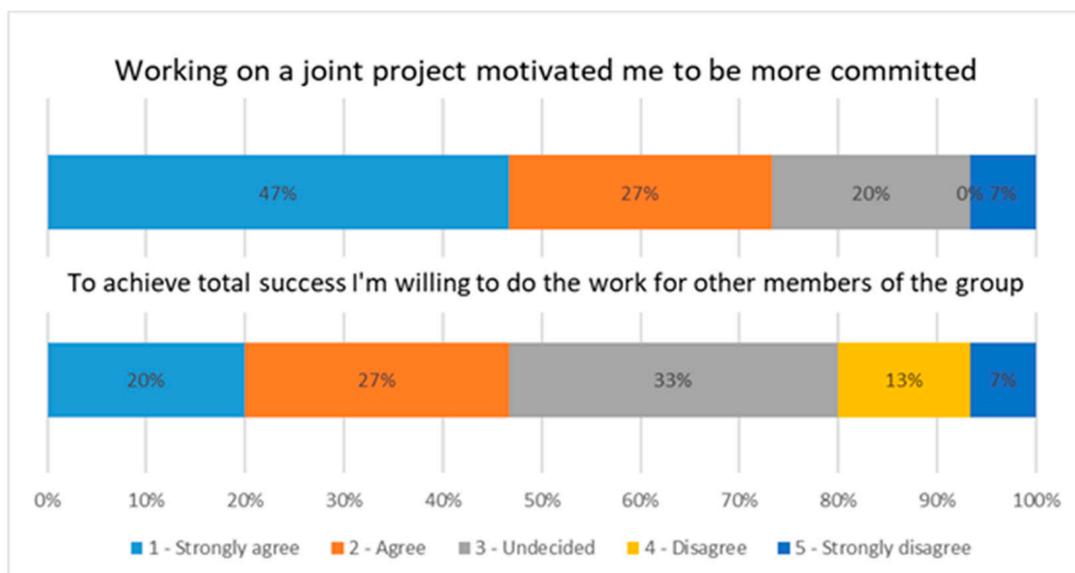


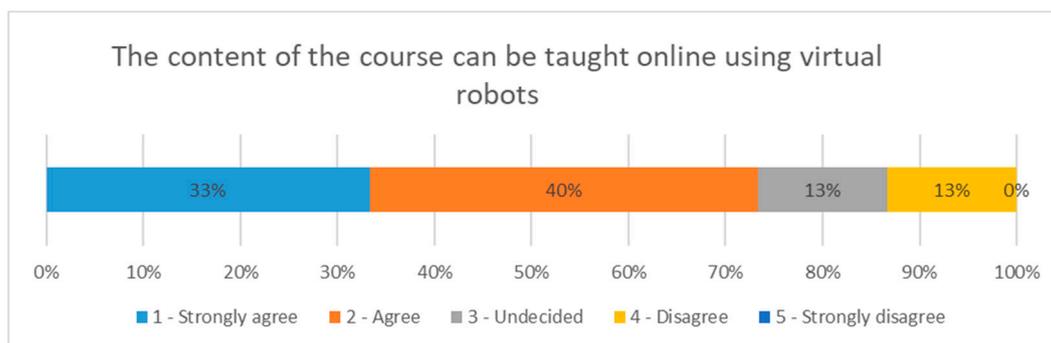
Figure 8. Expression of students’ attitudes to the statements “Working on a joint project motivated me to be more committed” and “To achieve total success, I am willing to do the work for other members of the group”.

In a summary of our findings on the second research question on joint work on the project motivating students to be more involved, we found that students were both positively and negatively affected by mood in the group and that students consolidated the curriculum of graph theory more when applying theoretical knowledge.

The third research question was as follows: (3) How is it possible to transfer teaching with educational robots to the online environment?

The last two statements in the questionnaire focused on the possibility of changing the full-time form of teaching to online teaching in the course. When changing the form of teaching the course, it is also necessary to change the approach to programming educational robots and change the form of students' work on the project. Changes in the form of teaching were discussed together with the students. When conducting online teaching on the application of theoretical knowledge in the field of graph theory to the practical solution of problems with virtual robots, it is appropriate to allow each student to work independently. The biggest threat for students was working independently in an online environment with a virtual robot. Some students feared that it would be difficult for them at the beginning of the course. Working in a group provides students with some support and confidence.

In the case of the answers to the questionnaire, the following were evident from the answers in the graph in Figure 9: teaching the course was considered possible after minor modifications of the course curriculum; we could transfer to an online environment and use virtual robots instead of real robots. A total of 73% of the selected answers (1 = strongly agree and 2 = agree) of students supported this statement.



**Figure 9.** Expression of students' attitudes to the statement: "The content of the course can be taught online using virtual robots".

An important finding was to reduce the failure rate of students in the chapter of the course focusing on mazes and labyrinths. Students' theoretical knowledge and practical skills in solving algorithms in the maze were low. The overall test failure in this chapter was high, reaching 35% on the first test. After the implementation and use of educational robots for teaching this chapter in the course, the failure rate of students on the first test improved to 20%. This value was determined only on the basis of successful completion of the first test of the knowledge and skill regarding algorithms for the maze. Therefore, we cannot consider this improvement to be authoritative.

### 3.3. Next Steps

We want to continue the study of the application of educational robots in the field of discrete mathematics, remaining within the STEM concept. We are currently focusing on online programming tools for virtual robots on various platforms.

## 4. Conclusions

In this study, we focused on a new concept of teaching graph theory using educational robots. The field of graph theory falls into discrete mathematics. The normal teaching of this course is more theoretical than practical. This paper presented a new concept of

teaching within the interconnection of individual areas of STEM. It is an application of algorithms to solve a robot's passage through a maze. Students compiled and programmed autonomous robots within the course. The robot was assembled from the Lego Mindstorms robotic kit. Students worked in groups on a project to solve the construction of the robot and to implement selected algorithms for passing through the maze. During the group work on the joint project, there was mutual learning and the interconnection of individual areas of STEM. An important part of the work was the implementation of selected algorithms for passing through the maze in the visual programming environment of the Lego Mindstorms program so that it was possible to use them in virtual environments to simulate working with real robots. The environment enables a simple and fast implementation of algorithms even for beginners. An option that would enable implementation in distance learning is offered by the use of simulated robots in a virtual environment.

An important part of the study was a research survey that used a questionnaire focused on students' attitudes toward the completed teaching in the course. During the course, students' work and their mutual interaction in the group were observed. The observations were focused on their mutual communication, work on a common task, division of roles in a group, and successful learning. At the end of the course, a research survey was conducted through semi-structured interviews with selected students. Here, this tool was used for a deeper understanding of students' behavior in the course and their answers to the questionnaire. The implementation and use of educational robots for teaching in the course provided many of the mentioned benefits. An important finding for our course was determining the level of failure of students in the chapter on maze and labyrinths. There was an improvement, with a reduction in student failure by 15%.

This study may inspire and motivate other lecturers and teachers to use educational robots in other mathematical areas.

**Author Contributions:** All authors contributed equally to all aspects of the research presented in this paper. Š.H. and M.H. conceived and designed the experiments; M.H. and P.C. performed the experiments; P.C. and M.H. analyzed the data; P.C., Š.H., and M.H. wrote the paper. All authors have read and agreed to the published version of the manuscript.

**Funding:** The authors was supported by the Project of Specific Research PrF UHK no. 2101/2021, University of Hradec Králové, Czech Republic.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Informed consent was obtained from all subjects involved in the study.

**Data Availability Statement:** Not applicable.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Biggs, N.L.; Lloyd, E.K.; Wilson, R.J. *Graph Theory 1736–1936*; Oxford University Press: Oxford, UK, 2006.
2. Kern, H. *Through the Labyrinth: Designs and Meanings over 5000 Years*; Prestel: Munich, Germany, 2000.
3. Okhotin, A. Graph-Walking Automata: From Whence They Come, and Whither They are Bound. In *Implementation and Application of Automata*; Lecture Notes in Computer Science; Hospodár, M., Jirásková, G., Eds.; Springer: Cham, Switzerland, 2019; Volume 11601.
4. Lucas, E. *Récréations Mathématiques*; Gauthier-Villars et fils: Paris, France, 1882; pp. 41–55.
5. Tarry, G. Le problème des labyrinthes. *Nouv. Ann. Math.* **1895**, *14*, 187–190.
6. Edmonds, J.; Johnson, E.L. Matching, Euler tours and the Chinese postman. *Math. Program.* **1973**, *5*, 88–124. [[CrossRef](#)]
7. Fleischner, H. *Algorithms in Graph Theory*; TU Wien, Database and Artificial Intelligence Group: Wien, Austria, 2016; pp. 37–42.
8. Pullen, W. Think Labyrinth: Maze Algorithms. Available online: <http://www.astrolog.org/labyrinth/algrithm.htm> (accessed on 20 September 2020).
9. Kilpatrick, W.H. The project method. *Teach. Coll. Rec.* **1918**, *19*, 319–335.
10. Dewey, J. *Experience and Education*; Macmillan: New York, NY, USA, 1938.
11. Dominguez, C.; Jaime, A. Database design learning: A project-based approach organized through a course management system. *Comput. Educ.* **2010**, *55*, 1312–1320. [[CrossRef](#)]

12. Nguyen, T.P.L.; Nguyen, T.H.; Tran, T.K. STEM Education in Secondary Schools: Teachers' Perspective towards Sustainable Development. *Sustainability* **2020**, *12*, 8865. [CrossRef]
13. Rocker, Y.S.; Shwartz, A.D.; Schohet, M.; Dori, Y.J. The Effect of the FIRST Robotics Program on Its Graduates. *Robotics* **2020**, *9*, 84. [CrossRef]
14. Jumaat, N.F.; Tasir, Z.; Halim, N.D.A.; Ashari, Z.M. Project-Based Learning from Constructivism Point of View. *Adv. Sci. Lett.* **2017**, *23*, 7904–7906. [CrossRef]
15. Zheng, J.; Xing, W.; Zhu, G.; Chen, G.; Zhao, H.; Xie, C. Profiling self-regulation behaviors in STEM learning of engineering design. *Comput. Educ.* **2020**, *143*, 103669. [CrossRef]
16. Wiek, A.; Withycombe, L.; Redman, C.L. Key competencies in sustainability: A reference framework for academic program development. *Sustain. Sci.* **2011**, *6*, 203–218. [CrossRef]
17. Tan, N.; Mohan, R.E.; Foong, S.; Yamakita, M.; Iwase, M.; Hatakeyama, S.; Kamamichi, N.; Song, L.; Wang, Y.; Zhu, Q. IDC Robocon: A Transnational Teaming Competition for Project-Based Design Education in Undergraduate Robotics. *Robotics* **2016**, *5*, 12. [CrossRef]
18. Jung, S. Experiences in Developing an Experimental Robotics Course Program for Undergraduate Education. *IEEE Trans. Educ.* **2013**, *56*, 129–136. [CrossRef]
19. Tjiharjadi, S. Design and Implementation of Flood Fill and Pledge Algorithm for Maze Robot. *Int. J. Mech. Eng. Robot. Res.* **2019**, *8*, 632–638. [CrossRef]
20. Ecsedi, E.; Silaghi, H.; Mihok, E.; Spoială, V. The Development of an Autonomous Maze Robot. In Proceedings of the 15th International Conference on Engineering of Modern Electric Systems (EMES), Oradea, Romania, 13–14 June 2019; pp. 169–172. [CrossRef]
21. Kumar, R.; Jitoko, P.; Kumar, S.; Pillay, K.; Prakash, P.; Sagar, A.; Singh, R.; Mehta, U. Maze Solving Robot with Automated Obstacle Avoidance. *Procedia Comput. Sci.* **2017**, *105*, 57–61. [CrossRef]
22. van Putten, B.J.S. *Design of a Maze Solving Robot Using Lego Mindstorms*; Technische Universiteit Eindhoven: Eindhoven, The Netherlands, 2006.
23. Iwano, T.M.; Vieira, J.S.D.S.; de Oliveira, D.M.; Scherer, D. A Teacher Experience Report with LEGO Robotics. In Proceedings of the IEEE 19th International Conference on Advanced Learning Technologies (ICALT), Maceio, Brazil, 15–18 July 2019; pp. 277–278. [CrossRef]
24. Seung, K.; Jeon, J. Programming LEGO Mindstorms NXT with visual programming. In Proceedings of the 2007 International Conference on Control, Automation and Systems, Seoul, Korea, 17–20 October 2007; pp. 2468–2472.
25. Del Rosario, J.R.; Sanidad, J.; Lim, A.; Uy, P.; Bacar, A.; Cai, M.; Dubouzet, A. Modelling and Characterization of a Maze-Solving Mobile Robot Using Wall Follower Algorithm. *Appl. Mech. Mater.* **2013**, *446–447*, 1245–1249. [CrossRef]
26. Saman, A.B.S.; Abdramane, I. Solving a Reconfigurable Maze using Hybrid Wall Follower Algorithm. *Int. J. Comput. Appl.* **2013**, *82*, 22–26.
27. Research Kitchen. RoboMind Challenges: Maze Solving. Available online: <https://www.robomind.net/downloads/Maze%20solving.pdf> (accessed on 20 September 2020).
28. Králík, V.; Majherová, J. Virtual Robotic Laboratories Robot C in the Informatic Teacher Training. In Proceedings of the XXIX. DIDMATTECH 2016, Budapest, Hungary, 25–26 August 2016.
29. Virtual Robotic Toolkit. Available online: <https://www.virtualroboticstoolkit.com/> (accessed on 20 September 2020).
30. Robot Virtual Worlds—LEGO 4.x. Available online: <http://www.robotc.net/purchase/rvw/rvw-lego-4.php> (accessed on 20 September 2020).
31. Solak, S.; Yakut, Ö.; Dogru Bolat, E. Design and Implementation of Web-Based Virtual Mobile Robot Laboratory for Engineering Education. *Symmetry* **2020**, *12*, 906. [CrossRef]
32. Cronbach, L.J. Coefficient alpha and the internal structure of tests. *Psychometrika* **1951**, *16*, 297–334. Available online: [http://psych.colorado.edu/~jcarey/Courses/PSYC5112/Readings/alpha\\_Cronbach.pdf](http://psych.colorado.edu/~jcarey/Courses/PSYC5112/Readings/alpha_Cronbach.pdf) (accessed on 20 September 2020).