LEaRN – LEgo Robot and Netlogo

Goran Zaharija
Faculty of Science, University of Split, Croatia
goran.zaharija@pmfst.hr

Ana Grubač
Faculty of Science, University of Split, Croatia
ana.grubac@pmfst.hr

Andrina Granić
Faculty of Science, University of Split, Croatia
andrina.granic@pmfst.hr

Abstract. This paper presents a framework that can be used in classroom in order to improve current teaching methods and motivate students by adopting a more interactive approach in teaching and learning process. Proposed framework provides a combination of simulated environment and real physical robot, both often used in classroom but rarely at the same time. An overview of the architecture of proposed framework is given. To test the effectiveness of the proposed framework, experimental study was conducted by using the framework in a single college course during one semester. Additionally, obtained observations and conclusions along with potential future work are discussed as well.

Key words: *Computer simulations, robotics, learning, framework*

1. Introduction

There are many different approaches which attempt to improve classroom experience for students. One of such approaches is relying on usage of different simulations in order to (i) promote the use of critical and evaluative thinking and (ii) make lectures more interesting by visualizing different problems, making them easier to comprehend. On the other hand, to improve traditional teaching methods and provide students with more active role in a class, another approach is using real physical robots in a classroom. This paper aims to present a framework that combines these two approaches by establishing a connection between simulated and physical representation of a particular environment (for example a robot in a maze). For the simulated environment we have used NetLogo, a multi-agent programmable modelling environment, while the physical representation was organized using Lego Mindstorms robots.

2. Background to the research

Lego Mindstorms robots have been subject of many researches in the last two decades. Most of those researches discuss advantage and disadvantage of using Lego Mindstorms robots as educational tool in a wide range of subjects such as computer science (Cliburn, 2006), engineering (Khalaf, Balawi, Hitt, & Radaideh, 2010), computer programming (Cliburn, 2006; Ewert, Schilberg & Jeschke, 2013), mechatronic (Tokuyasu, 2007), artificial intelligence (Klassner, 2002) etc. Furthermore, robots can be integrated in course curriculum at all levels of education; from elementary school (Hixon, 2007) to university level (Cliburn, 2006; Khalaf et al. 2010; Tokuyasu, 2007; Klassner, 2002).

Analysis of relevant literature indicates that Lego robots can be used in several educational purposes: (i) teaching basic concepts in computer science, programming, artificial intelligence, engineering and robotics, (ii) developing mathematical and logic thinking, (iii) developing problem-solving skills and (iv) motivating students to be engaged in class activities. Use of Lego robots has been inspired by constructivism learning theory (Alimisis, Moro, Arlegui, Pina, Frangou, & Papanikolaou, 2007) which claims that knowledge is actively constructed by a student, not passively absorbed from textbooks and lectures (Fosnot, 2013).

Cliburn (2006) used Lego Mindstorms to teach students fundamental concepts of computer science, introductory programming and advanced programing. Lego robots, generally, increase student enjoyment of computer science courses but it was concluded that they are not appropriate for every course. In courses of introductory programming and advanced programing teacher must find an appropriate programming interface and carefully choose projects. Most successful implementation of the robots in the classroom has been with fundamental concepts of computer science. Robots can be great examples, but only in the right contexts.

Fagin and Merkle (2003) conducted a year-long experiment that included using robots to teach computer science. Students were divided into “robotics” sections and “non-robotic” sections. Their results showed that “robotics” sections scores were lower than “non-robotic” sections scores.

Klassner case study (2002) examines LEGO Mindstorms suitability as a hardware platform for integrating robotics into an Artificial Intelligence course. Students worked on several team projects. Some of those projects were based on Mindstorms and NQC coding and some on Lisp coding. Students said that robotics problems were positively influencing their learning. Conclusion was that robot projects gave students a strong understanding of the concepts highlighted in the project goals.

In order to successfully implement Lego Mindstorms in the classroom, every student should have Lego Mindstorms kit and enough time to build her/his own robot. In small universities and courses with large number of enrolled students that can be problem. Therefore, many robot simulation environments have been developed like Netologo (Wilensky, 1999), EDURobot (Abiyev, Ibrahim & Erin, 2010), RoboKol (Conkur, 2006)], RoboXAP (Chu, Goldman & Sklar, 2005) and alike.

Abiyev, Erin and Ibrahim (2010) developed EDURobot, an educational software tool aimed to improve understanding of robotics for undergraduate and graduate students. EDURobot was developed to teach students different algorithms for navigation problems of a robot avoiding obstacles. They conducted case study to find out students’ opinions while using the EDURobot simulation program (Abiyev et al., 2010). Results show that the software tool has increased students’ knowledge and understanding of robotics and gave them a better insight into the various robotic path planning and navigation algorithms. All students said that a computer simulation is an effective way for them to learn.

Yadin (2013) designed undergraduate course that employs a visual learning environment that simulates a virtual two dimensional world which contains a robot that performs various tasks while overcoming obstacles that exist in its way. Students were given several assignments. For every assignment they were asked to verbally define the problem and outline the possible solution. After that they had to define the instructions for the robot. The study showed that visual environment is not sufficient for simplifying abstract concepts.

3. Proposed framework

As can be seen from the previous section, there are many different approaches in using simulations or physical robots in educational purposes. All of those approaches have a number of advantages and disadvantages. Main purpose of the framework is to make use of all advantages that using a simulated environment provides us with, while adding the possibility to demonstrate agent behaviour by using a physical robot.

Another advantage of the proposed framework is that it provides a feedback between the simulation and the robot by allowing the two-way communication between them avoiding oversimplification present when using only ideal environment. It is possible not only to transfer data from simulation to the robot, but the framework also allows the robot to send data back to the simulation (for example sensor values, or its current heading or coordinates). The received data can be included in the simulation and have an impact on its execution. This allows us to bridge the gap between the physical and the simulated environment, with all their similarities and differences creating a unique hybrid environment. Described approach should help students to familiarize themselves with some basic concepts associated with programming, artificial intelligence, robotics and multi-agent systems in a simple ways since the common sense reasoning is used when solving navigation problems.

Figure 1 shows an overview of the proposed framework architecture, while each component is described in details in following subsections.

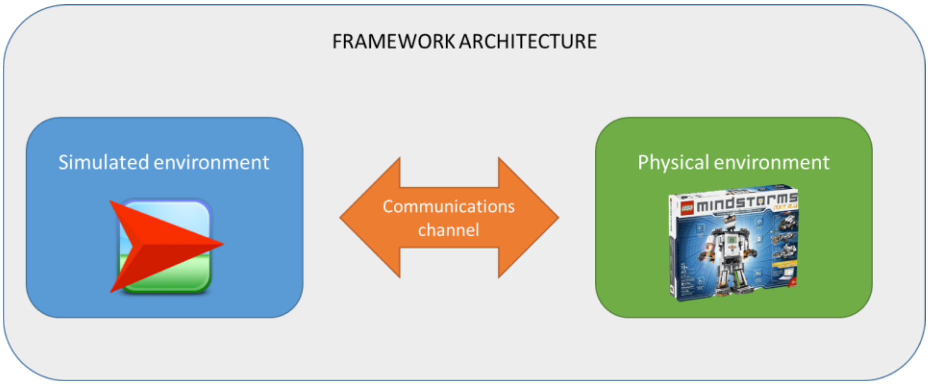

Figure 1 Overview of proposed framework architecture

3.1 Simulated environment

As already mentioned, Netlogo has been used as tool for depicting simulated environment of the robot. This environment was the primary choice due to many reasons, but mainly because its relative simplicity for beginners and also since Logo programming language is part of Croatian public school curriculum (Bakić-Tomić & Dumančić, 2009). For the purposes of this framework we have developed a simulation that displays a single robot in a finite discrete environment (a maze). This simulation allows us to customize various parameters such as dimensions of a maze, number of robots, dimensions of each patch and so on. The idea to introduce the maze is consistent with concrete application of several algorithms already learned throughout the course “Introduction to Artificial Intelligence”.

Simulation is designed using modular approach so it consists of 5 major modules: main code containing parameters specific to that simulation; agent’s behaviour; basic movement commands; maze generation; and data management (writing and loading). Figure 2 shows the overview of the simulation modules. Each of these modules can easily be changed without affecting the behaviour of other modules. Also they are all separated and can be individually included in other simulations. This provides a greater degree of customization thus ensuring that this framework is not limited to a single use case scenario.

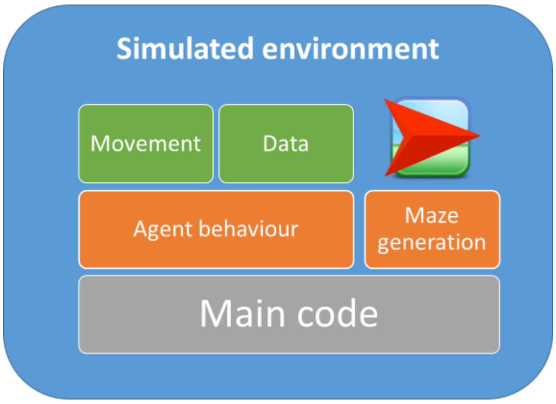

Figure 2 Main parts of simulated environment

In order to avoid reproducing the solutions given trough mentioned course, framework was designed with a functionality of exchanging different environments. Maze design can manually be reconfigured by painting or removing the walls or one of the few predetermined mazes can be loaded. Each robot located in the maze can be navigated manually or by a simple program which can be written using three basic movement commands (go forward, rotate right and rotate left) in conjunction with other Netlogo commands (such as loops and conditional statements). By allowing the manual control of agent's movement, students are given the possibility to solve given problems without possessing any programming knowledge or explicit formalization of algorithms used in maze solving. This approach is used to enable using the framework in the same way as solving problems on paper. Given the possibility to use basic commands, even students with no prior knowledge of programming can slowly be taught how to write basic programs.

Simulation is designed in such a way that every time a single robot performs one of basic movement commands, that command is memorized and added to the sequence of performed commands. This sequence of commands is transferred to the physical robot in order to execute them in real world. After sending current command and after the successful execution of that order by the physical robot, simulation receives feedback from that robot and displays obtained information within itself. If necessary, this last step can be skipped so that the simulation is only sending commands without receiving any feedback from physical robot, thus ensuring the usage of static environment.

As the orders are executed sequentially and with feedback, concept of agent's belief is introduced, as well as static environment that is represented by agent's view of the world within simulation. Following the interpretation and use of received feedback from obtained sensor values, the concept of dynamic environment is also introduced.

Simulated environment, when used as described, actually presents a mental model of the agent that possesses beliefs, desires and intentions (BDI) about its world, only not in a classical sense of BDI architecture (Chong, Tan & Ng, 2007) but as intuitive model of a surrounding world. Figure 3 shows an example of one simulation with different customization options available through the interface.

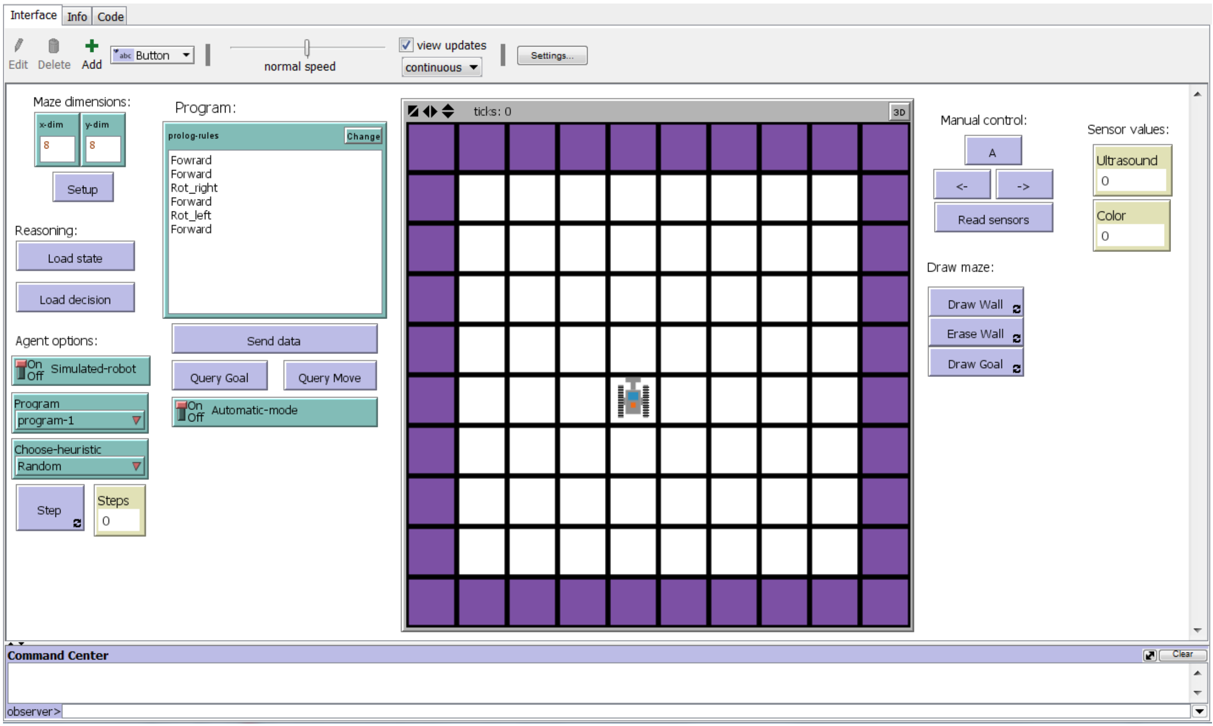

Figure 3 Netlogo simulation representing a single robot in an empty maze

3.2 Physical environment

After defining the mental image of the world, it is necessary to define an embodiment of an agent used in simulation, in order to further enhance teaching method using concrete examples in real world.

Lego Mindstorms robots were used for physical representation of the simulated agents. We have used both Lego Mindstorms NXT 2.0 and Mindstorms EVE 3 robots. One of major advantages of using these robots is that they can be assembled in many different ways which allows their usage in different classroom scenarios. They can be equipped with three different kinds of sensors (ultrasound sensor, colour sensor and touch sensor) and can be constructed with up to three motors. They are also capable of playing different sounds and displaying black and white images on their display. Their central processing unit is capable of memorizing and running different programs loaded in memory and they all come with embedded Bluetooth adapter, making it easier to establish a connection with presented framework. Although programs for Lego Mindstorms robots are usually developed by using Lego NXT-G programming language, robots used in this framework were programmed using NXC programming language. All of these abilities provide a wide range of possibilities for their usage. These robots can be used in differently constructed mazes, according the type of maze presented in the simulated environment. Figure 4 shows both simulated and physical representations of a single robot in a maze.

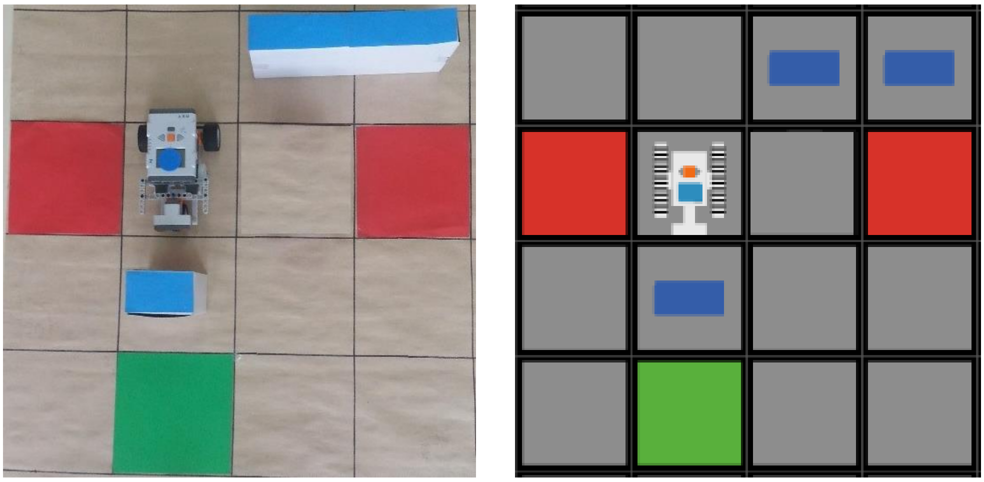

Figure 4 Physical and simulated representations of robot located in a maze.

3.3 Communications channel

In order to enable transfer of information between simulated and physical environments, existence of a communication channel is mandatory. This communication channel allows us to synchronize data between simulated and physical representation of a robot. Every action that physical robot makes in the real world can be presented in a real time the corresponding simulation. However, being a two-way communication it is also possible that physical robot mimics a move which simulated robot just made.

Communication between Netlogo simulation and Lego Mindstorms robot is carried out using Bluetooth adapters, both on robot and computer running the simulation. To properly exchange information between them, a special application was developed using .NET framework and C# programming language. This application is used to connect the computer running the Netlogo simulation with the Mindstorms robot via Bluetooth protocol. Figure 5 offers a snapshot of the application used for communication purpose.

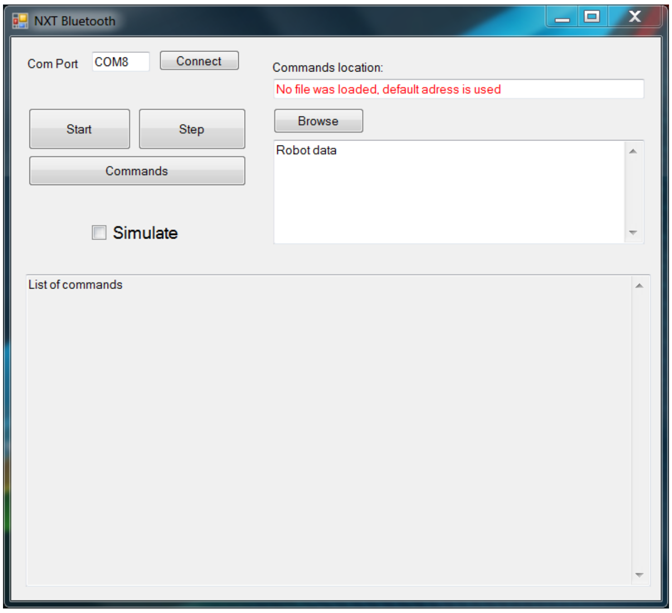

Figure 5 Application used as communications channel

After successfully pairing the simulation and the robot, program then loads the list of commands (or single command) given by the simulation and sends them to the robot.

Commands can be executed step by step or continuously but after every single command robot sends its current sensor values and position to the Netlogo simulation. According to the settings in simulations that information can also be loaded step by step or all at once. When using step by step execution of the command, application allows us to display the list of commands that are currently on queue. It is also possible to load another set of commands while there are some commands waiting to be executed. In that case, new commands are added to the end of queue and will be executed in order. When there are no more commands application will still constantly communicate with both the robot and simulation in order to check if there is any other information that needs to be exchanged. Figure 6 shows an example of communication between the simulation and physical robot.

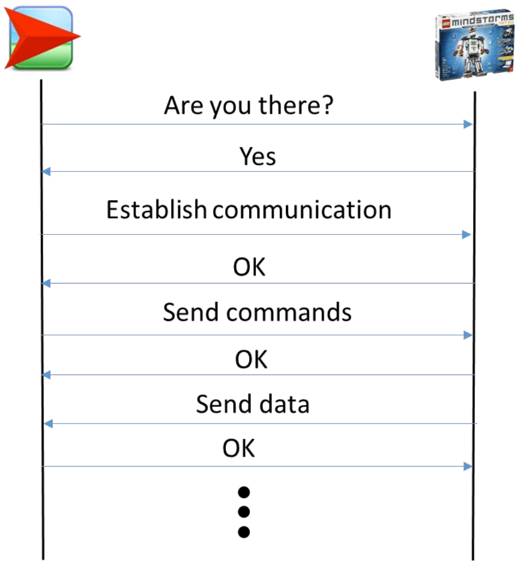

Figure 6 Communication between simulation and physical robot

4. Experimental study

To test the effectiveness of the presented framework, experimental study was conducted during the course "Introduction to Artificial Intelligence" in the year 2014. The course was attended by 39 enrolled students. This group of students was particularly interesting because some of them were enrolled in this class as part of their bachelor's study while for other students this course was part of master's study. In addition, they had different study majors (IT – information technology, mathematics, mathematics and IT, physics, IT and technics) thus forming a heterogonous group with different levels of programming ability as well as familiarity with basic concepts regarding artificial intelligence field. Aforementioned diversity in the group is useful because it can be used as indicator how different types of students react to the proposed framework.

Main goal of the experimental study was to use proposed framework during particular assignments and to see whether it would help students to successfully finish the assignment. Afterwards, students' understanding of given concepts associated with every assignment was evaluated. To successfully evaluate the effect of the framework in a class, it was first necessary to measure students' knowledge at the beginning of a semester, before the classes started, so that the progress during the course can be observed.

As a part of a first assignment, students had to fulfil a questionnaire asking them to evaluate their knowledge and familiarity with certain concepts that would be taught later during the course lectures and assignments. It should be noted that most of the students enrolled in this course previously attended (and successfully passed) the course “Data structures and algorithms” which already introduced them with different search algorithms and similar concepts. Average grade of students that attended the course “Data structures and algorithms” was 2,89 (out of 5) and the students were mostly unfamiliar with basic concepts such as “breadth first search”, “depth first search” and “reactive agent”.

Second part of the experiment was to use proposed framework in a number of selected assignments corresponding to concepts mentioned in given questionnaire and to compare success rate of those assignments with the success rate of assignments performed without using proposed framework. During the course there were 11 different assignments and proposed framework was used in 6 of them. Table 1 shows success rate for each of those assignments and offers an information whether the framework was used or not. An average success rate of assignments without the use of framework was 88,2% while average success rate of assignments with the framework was 96% percent.

Table 1 Success rate for each assignment and usage of framework

Assignment #	1	2	3	4	5	6	7	8	9	10	11
Success rate	74%	90%	97%	100%	90%	95%	95%	92%	92%	95%	97%
Framework used	No	No	Yes	Yes	No	Yes	Yes	No	No	Yes	Yes

Final form of an evaluation comprised of comparison of final grades from students enrolled this year with the final grades of students enrolled in the course past year, specifically 2013. Apart from using the framework during the particular assignments, there were not any significant differences between classes and assignments performed in those two subsequent years. Number of enrolled students was similar (39 this year compared to 42 last year) but there was significant increase in both passing grades (95% compared to 60%) and number of highest grades (23% to 12% percent). Table 2 shows detailed grade statistics and passing grades (from 2 to 5, with 5 being the maximum grade).

Table 2 Percentage of grades in two different groups of students

Year	**# of students**	**Grade percentage**				
		1	**2**	**3**	**4**	**5**
2014	39	10%	13%	26%	28%	23%
2013	42	40%	1%	5%	40%	12%

5. Conclusion and future work

After examining the results obtained from experimental study it can be noted that using the proposed framework improved students’ comprehension of basic concepts from the fields of artificial intelligence and programming. It can be noted that, on the one hand, students were better in performing given assignments with the help of the proposed framework. On the other hand students also obtained significantly better grades as a group when compared to students enrolled last year in the same course with the same curriculum except the use of framework in particular assignments.

In order to further and more thoroughly test the usefulness of the proposed framework, another experimental study which will involve students from elementary school, high school and college classes will be conducted. In such context, students should be divided in two groups, a control group that would not use the framework and a test group that would perform their assignments with the framework. This study would provide us with larger set of results as well as more detailed insight in the effects of this framework in different classes.

REFERENCES

Abiyev, R., Ibrahim, D., & Erin, B. (2010). Navigation of mobile robots in the presence of obstacles. Advances in Engineering Software, 41(10), 1179-1186.

Abiyev, R., Ibrahim, D., & Erin, B. (2010). Teaching robot navigation in the presence of obstacles using a computer simulation program. Procedia-Social and Behavioral Sciences, 2(2), 565-571.

Alimisis, D., Moro, M., Arlegui, J., Pina, A., Frangou, S., & Papanikolaou, K. (2007, August). Robotics & constructivism in education: The TERECoP project. In EuroLogo (Vol. 40, pp. 19-24

Bakić-Tomić, L., & Dumančić, M. (2009). Odabrana poglavlja iz metodike nastave informatike.

Chong, H. Q., Tan, A. H., & Ng, G. W. (2007). Integrated cognitive architectures: a survey. Artificial Intelligence Review, 28(2), 103-130.

Chu, K. H., Goldman, R., & Sklar, E. (2005). Roboxap: an agent-based educational robotics simulator. In Agent-based Systems for Human Learning Workshop at AAMAS-2005.

Cliburn, D. C. (2006, October). Experiences with the LEGO Mindstorms throughout the undergraduate computer science curriculum. In Frontiers in Education Conference, 36th Annual (pp. 1-6). IEEE.

Conkur, E. S. (2006). RoboKol: A computer program for path planning for redundant and mobile robots. Computer Applications in Engineering Education, 14(3), 198-210.

Ewert, D., Schilberg, D., & Jeschke, S. (2013). Problem based learning of object-oriented Programming with LEGO Mindstorms and leJOS. In Automation, Communication and Cybernetics in Science and Engineering 2011/2012 (pp. 315-323). Springer Berlin Heidelberg.

Fagin, B., & Merkle, L. (2003, February). Measuring the effectiveness of robots in teaching computer science. In ACM SIGCSE Bulletin (Vol. 35, No. 1, pp. 307-311). ACM.

Fosnot, C. T. (2013). Constructivism: Theory, perspectives, and practice. Teachers College Press.

Hixon, R. (2007). Teaching software engineering principles using Robolab and Lego Mindstorms. International Journal of Engineering Education, 23(5), 868-873.

Khalaf, K., Balawi, S., Hitt, G., & Radaideh, A. (2010). Innovation in teaching freshman engineering design: an integrated approach. EDULEARN10 Proceedings, 709-720.

Klassner, F. (2002, February). A case study of LEGO Mindstorms'™ suitability for artificial intelligence and robotics courses at the college level. In ACM SIGCSE Bulletin (Vol. 34, No. 1, pp. 8-12). ACM.

Tokuyasu, T. (2007, May). Installation of Mechatronics Education Using the MindStorms for Dept. of Mechanical Engineering, ONCT. In Mechatronics, ICM2007 4th IEEE International Conference on (pp. 1-5). IEEE.

Wilensky, U. (1999). NetLogo. http://ccl.northwestern.edu/netlogo/. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL.

Yadin, A. (2013). A VIRTUAL LEARNING ENVIRONMENT FOR ENHANCING STUDENTS'UNDERSTANDABILITY. In Conference proceedings of" eLearning and Software for Education"(eLSE) (No. 02, pp. 486-493).