

Review

Simulators in Educational Robotics: A Review

Sokratis Tselegkaridis and Theodosios Sapounidis *

Department of Information and Electronic Engineering, International Hellenic University (IHU),
57400 Thessaloniki, Greece; stsleggaridis@ihu.gr

* Correspondence: sapounidis@ihu.gr

Abstract: Educational robotics (ER) seems to have a positive effect on students and, in many cases, might help them to successfully assimilate knowledge and skills. Thus, this paper focuses on ER and carries out a literature review on educational robotics simulators with Graphical User Interfaces (GUIs). The review searches for relevant papers which were published in the period 2013–2020 and extracted the characteristics of the simulators used. The simulators that we describe in this article cover various robotic technologies, offering students an easy way to engage with virtual robots and robotics mechanisms, such as wheeled robots or drones. Using these simulators, students might cover their educational needs or prepare themselves for educational robotic competitions by working in as realistic as possible conditions without hardware restrictions. In many cases, simulators might reduce the required cost to obtain a robotic system and increase availability. Focusing on educational robotics simulators, this paper presents seventeen simulators emphasizing key features such as: user's age, robot's type and programming language, development platform, capabilities, and scope of the simulator.

Keywords: educational simulators; STEM education; educational robotics



Citation: Tselegkaridis, S.; Sapounidis, T. Simulators in Educational Robotics: A Review. *Educ. Sci.* **2021**, *11*, 11. <https://doi.org/10.3390/educsci11010011>

Received: 7 December 2020

Accepted: 30 December 2020

Published: 1 January 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

One of the purposes of educational robotics is to enhance even preschooler students' ability to acquire knowledge and skills through robot programming and development [1,2]. Students seem to develop cognitive and social skills by utilizing appropriate educational robots [3]. Such is the bonding of children with robots that children often treat them as pets and give them nicknames [4]. Usually, educational robotics cover a wide range of ages and cognitive requirements [5]. It is interesting that recent research studies have shown educational robotics having a positive impact even on children with cognitive and physical disabilities (e.g., [6–10]).

Simulators have already been used for educational robotics, from courses forwarding simple kinematics for children with the Lego Mindstorms [11], up to advanced courses for college students like RobotCraft [12,13]. Consequently, different simulators are almost incomparable because they cover different needs and have completely diverse characteristics. Particularly, simulators for research and STEM have been created for many years as an alternative way of learning with robots, promoting knowledge through gamification (e.g., [14–17]). While physical educational robots are largely made up of hardware which is not always easy to transport, simulators consist entirely of software that can easily be transported from place to place. This increases the portability and allows students to engage with robotics in any place and time as long as they have a computer or tablet. Unfortunately, simulators are based on GUIs and recent studies have shown that the use of a GUI creates difficulties and limitations on the interaction and cooperation of students, and shows a more positive attitudes towards other types of interfaces like the Tangible User Interface [18–20]. Although, it is quite difficult to accurately represent the real conditions of complex process with a simulator, ER simulators for virtual robots are considered a particularly appropriate method when real robots are not available. Similar to virtual labs,

simulators might reduce costs and have proven their value in the educational process by increasing availability, engagement, and simultaneously seem to be appreciated by students [21–23].

Some examples of desktop application for educational robotics simulators are Khepera Simulator [24], USARSim [25] and TRIK Studio [26]. Khepera, is a simulator which either drives a virtual Khepera wheeled robot or programs a real one, with C/C++. Using the simulator, users may program a virtual or real robot to follow lines and avoid obstacles, through IR sensors. USARSim has many more capabilities and uses plenty of virtual sensors, like sonar, RFID, and touch. This simulator enables the user to develop various simulation scenarios, with wheeled vehicles, submarines, or even humanoids. On the other hand, TRIK Studio has the ability to program robots with either visual diagrams or high-level programming languages. For the development of simulators, programmers frequently use general-purpose platforms such as MATLAB [27] and Processing [28], or open-source robotics platforms like Gazebo, V-REP, and others [29–31]. Both Gazebo and V-REP have a 3D dynamic multi-robot environment [32]. In addition, the V-REP simulator seems to be more user friendly than Gazebo, whereas Gazebo is more hardware-demanding than V-REP [33].

Along with desktop simulators the educational community also uses online simulators, such as: (a) Open Roberta Lab [34], which is a programming environment introducing kids to robotic systems, (b) Tinkercad [35], which includes simulation for electronic components, (c) MakeCode for the Lego Mindstorms [36], which programs Lego Mindstorms with blocks or JavaScript, (d) Snap4Arduino [37], which is block-based and supports plenty of Arduino boards, and (e) OzoBlockly [38], which is block-based and simulates Ozobot robots.

All things considered, educational robotics simulators cover a wide range of educational scenarios. Previous reviews have focused on: (a) open source simulator platforms like Gazebo or V-REP [31], (b) simulators in teaching industrial robotics [39], and (c) simulator platforms for teaching kinematics, dynamics and control, specifically in the industrial sector [40]. Thus, this paper, focusing on ER, provides an overview of the educational robotics simulators and aims, this way, to inform both teachers and scientists about the simulators which have been scientifically used by the community. In more detail, simulators' key features have been recorded, such as: a) users' age, (b) robot's programming language, (c) robot type, (d) development platform, (e) scope (education, competition, research), (f) year of development, (g) the needed user experience to "run" the simulators, and (h) operating system (Windows, Linux, MacOS), etc.

2. Methods

For the implementation of this review, Google Scholar was used. We extracted articles published between 2013 and 2020. As search terms, we used the keywords: (a) "Simulator for educational robotics", (b) "Educational robotics" and "simulator developed", (c) "Technology for education" and "robotic simulator", (d) "Robot challenges" and "simulator", and (e) "Autonomous driving", "educational robotics" and "simulator". In total, 84 results were found and, based on the criteria above (see Table 1), seventeen educational robotics simulators emerged, which are briefly presented below.

It should be noted that this review was based on the literature and does not describe simulators which were not found on the extracted papers, such as the online simulators mentioned in the Introduction.

Table 1. Inclusion—Exclusion criteria.

Keywords	Inclusion Criteria	Exclusion Criteria
“Simulator for educational robotics”	Papers published between 2013 and 2020	Papers published before 2013
“Educational robotics” and “simulator developed”	Papers published between 2013 and 2020	Papers published before 2013
“Technology for education” and “robotic simulator”	All items were retrieved	No exclusion criteria
“Robot challenges” and “simulator”	Papers published between 2013 and 2020	Papers published before 2013
“Autonomous driving”, “educational robotics” and “simulator”	Papers published between 2013 and 2020	Papers published before 2013

3. Results

The following is a brief description of the simulators that emerged from the review.

3.1. Robot Soccer Simulator—RoSoS

The RoSoS [41] was developed for the RoboCup competition, and created to offer students an easy and fun way to engage in educational robotics. This simulator is open-source (can be downloaded from [42]) and was developed aiming at children who cannot participate in educational competitions. In particular, the simulator focused on first- and second-grade students and simulates a football match. The system provides many settings, such as: field size, ball size, number of players for each team, and player speed.

RoSoS uses Processing Integrated Development Environment (IDE) for its programming [28]. The virtual robots were developed to use similar programming code to the real ones. Each robot may be an attacker or a goalkeeper and can have its own code and set of sensors. The simulator includes the sensors that are usually needed in real robots playing football such as: (a) infrared sensor for the recognition of the ball, (b) compass sensor for the correct orientation of the robot on the field, and (c) distance sensor for the recognition of the field borders.

RoSoS was tested by university students at the Federal University of ABC. RoSoS was also used in the Robotics course of the Federal Institute—Espírito Santo (IFES).

3.2. RoboSim

The RoboSim [23] is programmed in the C/C++ language and aimed at secondary school students. This simulator was created to provide a graphical user interface for the RoboPlay Challenge competition [43]. The competition uses the Linkbots modular robots and peripherals [44]. Every Linkbot has two motors, and therefore two degrees of freedom, thus, many Linkbots can be combined to create multiple activities/tasks.

The simulator requires the same actions to program the virtual robot, as the real one. This way, each student can be prepared and be acquainted with the competition. Specifically, the user can see three areas in the GUI. On the left area, the user might see the available modular robots/accessories. The upper right area shows the current state of the simulation, while the lower right area has a dialog box for setting up each object. Furthermore, grid lines can be added to the GUI so the user is able to see the exact position of the robot as well as check whether the requested exercise is correctly performed. Finally, by using one button, the user can load the code directly from the simulator to the real robot.

3.3. Robot One

The Robot One [45] is an open-source simulator (can be downloaded from [46]) for various robots and provides sensors such as GPS, light detection, and an RGB camera. The simulator can be programmed in many languages and provides a fully configurable graphical environment for the robot. Thus, it was designed as two separate parts. The first is the R-One User Interface (UI) which is based on the Unity3D platform [47], and the second is the R-One Application Programming Interface (API). With the R-One UI, users

can represent a robot and its environment, while with the R-One API can program the robot. This simulator was aimed at college students and was used during a semester in the course Mobile Robotics at the Federal University of Juiz de Fora. According to the authors, the simulator worked as expected [45].

3.4. Tactode Simulator

The Tactode [48] simulator was created for elementary school students to learn robot programming through play and fun. The programming language is a tangible puzzle with block-shaped pieces. When the user completes the puzzle, they must take a photo of the sequence and upload it to the simulator. An uploaded puzzle is processed as an abstract syntax tree [49]. The abstract syntax tree includes letters, numbers, operators, movements, control, variables, events, visuals, and sensors. Various sensors are used, such as proximity for obstacle recognition. Finally, there is one more programming element, the “Question” piece [50]. When the system recognizes a “Question” piece, an input window opens for the user to enter an answer, much like entering variables in a program. This simulator can also work on smartphones and tablets.

3.5. Pololu 3pi Simulator

The Pololu 3pi [51] is an open-source simulator (can be downloaded from [52]) and was developed for the Pololu 3pi robot. The robot has got LEDs, a piezo buzzer, an LCD screen, pushbuttons, and other components. It is wheeled with basic functions, such as to follow a line or solve maze problems. The simulator’s programming language is C/C++ and accepts the .ino files, which may also program the real robot. The graphical interface of the simulator consists of three areas. On the left area there is a window of 600x600 pixels, in which the simulation takes place. On the top right area, the virtual robot’s components are displayed, such as: LCD screen, pushbuttons, and LEDs. On the bottom right area there are two other buttons to control the simulation: (a) Start/pause, (b) Reset. The simulator was tested by two groups of five undergraduate students, and the results showed that the use of the simulator increased the students’ score.

3.6. EdUcational Ros rObot PLAtform—EUROPA

EUROPA platform [53] was developed for secondary school students. It has a wheeled robot that is programmed via Python, and it is based on the Raspberry Pi3 B+ board [54]. The virtual robot, among other components, includes: a raspberry pi camera, ultrasonic, and a Light Detection and Ranging (LIDAR) sensors. This simulator can be used in STEM courses and has been used to teach trigonometry and physics. EUROPA can also be used to make students aware of basic kinematics and remote-control difficulties. The simulator was developed with the Robot Operating System (ROS) [55] and the Gazebo platform [29]. Finally, the platform was tested by secondary school students with previous robotics knowledge.

3.7. Autonomous Driving Simulator—ADS

Autonomous Driving Simulator (ADS) [56] was developed to provide a virtual training environment for the Autonomous Driving Competition of the Portuguese Robotics Open (ADC PRO) [57]. This open-source simulator (can be downloaded from [58]) visualizes the track and the driving challenges of the competition, such as off-road travel, traffic light recognition, parking, and much more. The autonomous driving simulator is aimed at college students and it is based on the Gazebo platform. The simulator is divided into two parts: (a) the environment, which represents the natural environment that the contestants face, (b) the design of the robot.

Two robots are available for simulations, Conde and Major Alvega. The first one took part in the competitions between 2016 and 2018, while the second took part in 2019. The simulator was used in a Master’s course in Informatics and Computing Engineering at the University of Porto, as well. The creators propose the creation of a virtual automatic referee as a future work, so that autonomous driving competitions can be created in the simulator.

3.8. Simulator with MLPNN

This simulator [59] is aimed at college students and has a wheeled robot that is controlled by Multilayer Perceptron Neural Network (MLPNN). In particular, it was designed with MATLAB, and the robot's code and settings are defined in .m files. The simulator offers several scenarios, while the robot can be modified in size and shape. The simulator's window shows a 2×2 m "room", which contains the virtual robot and obstacles. When the user makes a manual simulation, the data can be recorded, so the natural network is trained to be more efficient.

3.9. Drone Simulator

This simulator [60] is aimed at secondary school students and was developed to simulate a quadcopter-drone's flight tasks for the Drone Challenge [61]. In fact, each team in the Drone Challenge has to complete three steps: (a) take the drone off its base, (b) pass the drone through colored frames in a specific order, and (c) return the drone to its base. Provided that more than one team succeeds in all steps, the team that completes the missions in less time gets a higher score. Multiple scenarios can be created by changing the position of the take-off/landing base, as well as the position of the frames. The simulator was developed with MATLAB/Simulink [62] and Gazebo platform. The Gazebo is responsible for the virtual representation of the drone's flight, while the Simulink calculates the drone's route, according to the user's parameters. The two applications constantly communicate through a vector with the values of the three axes x, y, z . Thus, the position and orientation of the drone can be visualized. Finally, the simulator was tested by 36 people, 19 students, and 17 teachers, and showed a positive attitude towards the platform.

3.10. Simulator for PiBot

PiBot [63] is a wheeled robot platform for secondary school students which can be used to teach various exercises in robotics and is compatible with the most common Arduino sensors. This open-source simulator (could be downloaded from [64]) for PiBot is based on the Gazebo platform and provides a 3D model of the real robot. The simulation includes several sensors, such as sonar, IR, and a camera. Moreover, the simulation covers scenarios like follow-line and avoid obstacles. Currently, the robot is programmed by Python, but the creators' intention is to expand simulator's capability and also support Scratch [65]. Finally, the simulator was used in a Robotics workshop for teachers at the Campus of Fuenlabrada of the Rey Juan Carlos University.

3.11. AlphaBot2 Simulator

AlphaBot2 [66] is an open-source simulator (could be downloaded from [67]) and was created for AlphaBot2—Pi, which is a wheeled robot for children over 12 years old. The simulator is based on the Gazebo platform and features a virtual camera. For simplicity, the shape of the virtual robot was implemented with simple geometric shapes, such as squares and cylinders. The robot in the simulation has two operations: (a) identify obstacles and (b) follow a line. To detect/avoid obstacles, the robot uses two IR sensors while, for line recognition and tracking, the robot uses five IR sensors.

3.12. MSRP Simulator

The Modified Stingray Robot Platform (MSRP) simulator [68] is based on the Stingray wheeled robot [69]. The simulator is aimed at university students and has three parts. The first part is a visualization engine, which is built based on Unity3D [47], and creates the virtual environment from one XML file. The second part is the chassis simulation, which is the system that calculates the position and direction of the robot. The last part is the sensor simulation, which manages all the sensors required for simulation.

3.13. Khepera IV Simulator

The Khepera IV simulator [70] was created for the Khepera IV wheeled robot. The virtual robot is programmed in C language and can perform various functions such as: (a) recognizing and avoiding obstacles in a dynamically changing environment, (b) following a line, and (c) following a path. The parts of the virtual robot were designed in Autodesk Inventor [71] and the simulator was based on the V-REP [30], which is a platform for developing simulators. Finally, the robot's model includes all the sensors of the real robot, like ultrasonic and IR sensors, without the gyroscope and the accelerometer.

3.14. OBR Simulator

This simulator [72] is used in the challenges of the Brazilian Robotics Olympiad—OBR (Olimpíada Brasileira de Robótica) [73]. The system simulates tasks for wheeled robots, such as: line following, obstacle avoidance, and crossroad selection. The OBR competition is well known in Brazil and more than 300,000 students aged from 6 to 18 have taken part [74].

The simulator is based on the Robot Operating System (ROS) [55] and the V-REP platform. The simulation includes two maps: (a) the first map consists of a 10×10 grid where the robot moves, while (b) the second map is used for routes with obstacles. The virtual robots are programmed through flowcharts and the simulations support sensors, such as vision, proximity, and ultrasonic. To create the programming flowchart, the software application FluxProg was developed. Finally, the simulator was tested by technical high school students in electronics and mechanics in the city of Curitiba, Brazil.

3.15. MiniSim

Minibloq [75] is an open-source software application (could be downloaded from [76]), which includes the MiniSim 2D simulator designed for kids. The provided virtual robot is wheeled and equipped with a pencil drawing lines. It is two-way programmable, with Python or block-based graphical programming language. Specifically, some of the blocks are: (a) control for robot's movement, (b) pencil control, (c) variable creator, (d) variable value assignment, (e) waiting specified time interval, and (f) logic condition-based decisions. The simulator also provides a sensor for the robot to recognize the color of the drawing lines.

3.16. SRM

The Simulator for Robotic Manipulator (SRM) [77] represents a virtual robotic arm and was created for college students. The SRM is based on the MATLAB and V-REP platform. The MATLAB performs the manipulator control, while the V-REP is responsible for the virtual representation of the manipulator's movement. The virtual arm can move in two axes and consists of two parts: (a) the body, with two degrees of freedom, and (b) the end-effector, with three degrees of freedom. The SRM was used and tested by university students at the Federal University of Juiz de Fora.

3.17. LaRoCS + Unesp Simulator

LaRoCS + Unesp simulator [78] was created by undergraduate and graduate students from Unicamp and Unesp universities [79,80]. The simulator provides a virtual quadcopter-drone based on the Parrot AR Drone 2.0 [81]. The simulator was also developed to participate in RoboCup's Flying Robot challenges. In this competition, every team faces challenges like take-off and obstacle avoidance, following a trajectory and landing. The simulator was based on the V-REP platform, which is responsible for the virtual representation of the drone's flight.

3.18. Summary Table

The following Table 2 is a summary that shows the simulators and their features. More specifically, the table indicates the name of the simulator, the year of development, the type of the robot, the users' age, and the robot's programming language. Furthermore, the table

indicates the scope of the simulator. In particular, the simulators which developed through a research program are marked as “Research”. The simulators developed for a competition are marked as “Competition”, while the simulators developed for the educational process are marked as “Education”. Moreover, we have recorded the usage, the needed user experience to “run” the simulators and whether simulators are based on a real robot or not. Finally, the last two columns indicate the simulator’s development platform and operating system (Windows, Linux, MacOS).

Table 2. Features of simulators.

Name	Develop Year	Type of Robot	Users' Age	Programming Language	Scope	Used	User's Level	Based on a Real Robot	Development Platform	Operating System
RoSoS *	2016	Wheeled	6~7	Processing	Competition	1, 2	Beginner	No	Processing	W, L, M
RoboSim	2017	Modular	10~18	C/C++	Competition	1	Intermediate	Yes	-	W, M
Robot One *	2019	Multiple	>18	Many	Education	2	Expert	No	Unity3D	W, L
Tactode	2019	Multiple	6~11	Puzzle/Tangible	Education	-	Beginner	No	-	n/s
Pololu *	2018	Wheeled	n/s	C/C++	Education	2	Beginner	Yes	-	W
EUROPA	2019	Wheeled	>12	Python	Education	3	Intermediate	Yes	Gazebo	n/s
ADS *	2019	Wheeled	>18	n/s	Competition	1, 2	Intermediate	Yes	Gazebo	L
MLPNN	2020	Wheeled	>18	MATLAB	Research	-	Expert	No	MATLAB	W, L
Drone Simulator	2019	Drone	>12	MATLAB	Research	1, 3	Expert	No	MATLAB/Gazebo	W, L
PiBot *	2018	Wheeled	>12	Python	Education	4	Intermediate	Yes	Gazebo	W, L, M
AlphaBot2 *	2019	Wheeled	>12	n/s	Research	-	Intermediate	Yes	Gazebo	W, L
MSRP	2017	Wheeled	>18	n/s	Education	-	Expert	Yes	Unity3D	n/s
KheperaIV	2016	Wheeled	n/s	C/C++	Education	-	Intermediate	Yes	V-REP	n/s
OBR simulator	2018	Wheeled	6~18	Flowchart	Competition	1, 5	Intermediate	Yes	V-REP	W, L
MiniSim *	2014	Wheeled	6~10	Python/Block	Education	-	Beginner	No	-	W, L
SRM	2018	Arm	>18	MATLAB	Education	2	Expert	No	MATLAB/V-REP	W, L
LaRoCS + Unesp	2018	Drone	n/s	n/s	Competition	-	Intermediate	Yes	V-REP	W, L, M

Note: 1: Competition, 2: University, 3: Secondary School, 4: Robotics workshops for teachers, 5: Technical high school; W: Windows, L: Linux, M: MacOS; *: Open access; n/s: not specified.

4. Discussion

In this article, seventeen educational robotics simulators were described and six key features are highlighted: (a) the robot's type, (b) the users' age, (c) the robot's programming language, (d) the purpose of development, (e) the users' level and (f) the development platform. More specifically, almost half of the simulators were developed after 2019. Users' ages vary from elementary students (6+) to college students (18+). Seven of the simulators are programmed by Python or C/C++, three are programmed by MATLAB, one by a tangible user interface, and one by flowcharts. Six of them were developed for educational competition letting contestants be better prepared and familiarized with the competitions. Lastly, most simulators were based on development platforms.

Simulators are part of educational robotics, which easily and quickly enable the user to engage virtually with the development and programming of robots through GUIs. Using a simulator means that it is not necessary to deal exclusively with real robots that might have a significant cost. Hence, simulators are a useful tool that might save resources and assist the educational process. However, the previously described simulators, which cover a wide range of needs, are programmed differently and aim at various ages of students. Therefore, the comparison among them is almost impossible.

5. Limitations

The simulators described in this review, which emerged from research based on specific criteria (see Table 1). Educational robotics simulators which did not appear on the extracted papers (such as Webots) are not described in this review.

Author Contributions: S.T. and T.S. conceived, designed, and wrote together this paper. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Ethical review and approval were waived for this study, due to literature review.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

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

References

1. Sapounidis, T.; Demetriadis, S. Educational robots driven by tangible programming languages: A review on the field. *Adv. Intell. Syst. Comput.* **2017**, *560*, 205–214.
2. Todorovska, K.; Bogdanova, A.M. Educational robots in preschool education. In Proceedings of the CIIT 2020 17th International Conference on Informatics and Information Technologies, Krushevo, North Macedonia, 8–10 May 2020; pp. 17–19.
3. Damaševičius, R.; Narbutaitė, L.; Plauska, I.; Blažauskas, T. Advances in the use of educational robots in project-based teaching. *TEM J.* **2017**, *6*, 342–348. [\[CrossRef\]](#)
4. Bungert, K.; Bruckschen, L.; Müller, K.; Bennewitz, M. *Robots in Education: Influence on Learning Experience and Design Considerations*; The International Academic Forum (IAFOR): London, UK, 2020.
5. Sapounidis, T.; Alimisis, D. Educational Robotics for STEM: A Review of Technologies and Some Educational Considerations. In *Science and Mathematics Education for 21st Century Citizens: Challenges and Ways Forward*; Nova Science Publishers: Hauppauge, NY, USA, 2020; pp. 167–190. ISBN 978-1-53618-334-4.
6. Robins, B.; Dautenhahn, K.; Ferrari, E.; Kronreif, G.; Prazak-Aram, B.; Marti, P.; Iacono, I.; Gelderblom, G.J.; Bernd, T.; Caprino, F.; et al. Scenarios of robot-assisted play for children with cognitive and physical disabilities. *Interact. Stud. Stud. Soc. Behav. Commun. Biol. Artif. Syst.* **2012**, *13*, 189–234. [\[CrossRef\]](#)
7. Conchinha, C.; Osório, P.; De Freitas, J.C. Playful learning: Educational robotics applied to students with learning disabilities. In Proceedings of the 2015 International Symposium on Computers in Education (SIIE), Setúbal, Spain, 25–27 November 2016; pp. 167–171. [\[CrossRef\]](#)
8. Valadao, C.; Bastos, T.F.; Bortole, M.; Perim, V.; Celino, D.; Rodor, F.; Goncalves, A.; Ferasoli, H. Educational robotics as a learning aid for disabled children. In Proceedings of the 2011 ISSNIP Biosignals and Biorobotics Conference: Biosignals and Robotics for Better and Safer Living, BRC 2011, Vitoria, Brazil, 6–8 January 2011; pp. 80–85. [\[CrossRef\]](#)

9. Bargagna, S.; Castro, E.; Cecchi, F.; Cioni, G.; Dario, P.; Dell’Omo, M.; Di Lieto, M.C.; Inguaggiato, E.; Martinelli, A.; Pecini, C.; et al. Educational Robotics in Down Syndrome: A Feasibility Study. *Technol. Knowl. Learn.* **2019**, *24*, 315–323. [CrossRef]
10. Chiara, M.; Lieto, D.; Pecini, C.; Castro, E.; Inguaggiato, E.; Cecchi, F.; Dario, P.; Sgandurra, G.; Cioni, G. Smart Learning with Educational Robotics. *Smart Learn. Educ. Robot.* **2019**, 229–250. [CrossRef]
11. González-García, S.; Rodríguez-Arce, J.; Loreto-Gómez, G.; Montaña-Serrano, V.M. Teaching forward kinematics in a robotics course using simulations: Transfer to a real-world context using LEGO mindstorms™. *Int. J. Interact. Des. Manuf.* **2020**, *14*, 773–787. [CrossRef]
12. Fonseca Ferreira, N.M.; Araujo, A.; Couceiro, M.S.; Portugal, D. Intensive summer course in robotics—Robotcraft. *Appl. Comput. Informatics* **2020**. [CrossRef]
13. RobotCraft. Available online: <https://robotcraft.ingeniarius.pt/about/venue> (accessed on 19 October 2020).
14. Hugues, L.; Bredeche, N. Simbad: An autonomous robot simulation package for education and research. In *Proceedings of the Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*; Springer: Berlin/Heidelberg, Germany, 2006; Volume 4095, pp. 831–842.
15. Caccavelli, J.; Pedre, S.; De Cristóforis, P.; Katz, A.; Bendersky, D. A new programming interface for educational robotics. In *Proceedings of the Communications in Computer and Information Science*; Springer: Berlin/Heidelberg, Germany, 2011; Volume 161, pp. 68–77.
16. Kortemeyer, G. Game Development for Teaching Physics. In *Proceedings of the Journal of Physics: Conference Series*; Institute of Physics Publishing: Johannesburg, South Africa, 2019; Volume 1286, p. 012048.
17. Fernandez, B.G.; Del Toro, X.; Santofimia, M.J.; Dorado, J.; Villanueva, F.J.; Villa, D.; Lopez, J.C. Robotics vs. game-console-based platforms to learn computer architecture. *IEEE Access* **2020**, *8*, 95153–95169. [CrossRef]
18. Schneider, B.; Sharma, K.; Cuendet, S.; Zufferey, G.; Dillenbourg, P.; Pea, R. Using Mobile Eye-Trackers to Unpack the Perceptual Benefits. *ACM Trans. Comput. Interact.* **2016**, *23*, 39.
19. Sapounidis, T.; Demetriadis, S.; Papadopoulos, P.M.; Stamovlasis, D. Tangible and graphical programming with experienced children: A mixed methods analysis. *Int. J. Child Comput. Interact.* **2019**, *19*, 67–78. [CrossRef]
20. Kurniawan, O.; Lee, N.T.S.; Datta, S.; Sockalingam, N.; Leong, P.K. Effectiveness of Physical Robot Versus Robot Simulator in Teaching Introductory Programming. In *Proceedings of the 2018 IEEE International Conference on Teaching, Assessment, and Learning for Engineering (TALE)*, Wollongong, Australia, 4–7 December 2019; pp. 486–493. [CrossRef]
21. Lamb, R.; Lin, J.; Firestone, J.B. Virtual reality laboratories: A way forward for schools? *Eurasia J. Math. Sci. Technol. Educ.* **2020**, *16*. [CrossRef]
22. Estriegana, R.; Medina-Merodio, J.A.; Barchino, R. Student acceptance of virtual laboratory and practical work: An extension of the technology acceptance model. *Comput. Educ.* **2019**, *135*, 1–14. [CrossRef]
23. Gucwa, K.J.; Cheng, H.H. Making robot challenges with virtual robots. In *Proceedings of the Conference on Innovation and Technology in Computer Science Education*, Bologna, Italy, 3–5 July 2017; pp. 273–278. [CrossRef]
24. Michel, O. *Khepera Simulator version 2.0 User Manual*; French National Centre for Scientific Research: Paris, France, 1996.
25. Lewis, M.; Wang, J.; Hughes, S. USARSim: Simulation for the Study of Human-Robot Interaction. *J. Cogn. Eng. Decis. Mak.* **2007**, *1*, 98–120. [CrossRef]
26. TRIK Studio. Available online: <https://trikset.com/en/products/trik-studio> (accessed on 17 December 2020).
27. MATLAB—MathWorks—MATLAB & Simulink. Available online: <https://www.mathworks.com/products/matlab.html> (accessed on 28 November 2020).
28. Processing.org. Available online: <https://processing.org/> (accessed on 8 November 2020).
29. Hughes, J.; Shimizu, M.; Visser, A. A Review of Robot Rescue Simulation Platforms for Robotics Education. In *Proceedings of the Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*; Springer: Sydney, Australia, 2019; Volume 11531, pp. 86–98.
30. Robot Simulator CoppeliaSim: Create, Compose, Simulate, Any Robot—Coppelia Robotics. Available online: <https://www.coppeliarobotics.com/> (accessed on 18 October 2020).
31. Vrochidou, E.; Manios, M.; Papakostas, G.A.; Aitsidis, C.N.; Panagiotopoulos, F. Open-Source Robotics: Investigation on Existing Platforms and Their Application in Education. In *Proceedings of the 2018 26th International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*, Split, Croatia, 13–15 September 2018; pp. 1–6. [CrossRef]
32. Koenig, N.; Howard, A. Design and use paradigms for Gazebo, an open-source multi-robot simulator. In *Proceedings of the 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sendai, Japan, 28 September–2 October 2004; Volume 3, pp. 2149–2154. [CrossRef]
33. Nogueira, L. Comparative Analysis between Gazebo and V-REP Robotic Simulators Lucas. In *Proceedings of the 2011 International Conference on Materials for Renewable Energy & Environment*, Shanghai, China, 20–22 May 2011; Volume 2, pp. 1678–1683. [CrossRef]
34. Open Roberta Lab. Available online: <https://lab.open-roberta.org/> (accessed on 20 November 2020).
35. Learn How to Use Tinkercad | Tinkercad. Available online: <https://www.tinkercad.com/learn/circuits> (accessed on 20 November 2020).
36. LEGO®MINDSTORMS®Education EV3—Blocks/Javascript Editor. Available online: <https://makecode.mindstorms.com/> (accessed on 20 November 2020).
37. Snap4Arduino 6.2. Available online: <http://snap4arduino.rocks/run/> (accessed on 20 November 2020).

38. OzoBlockly | Ozobot. Available online: <https://ozobot.com/create/ozoblockly> (accessed on 21 November 2020).
39. Teixeira, J.V.; Da, M.; Hounsell, S. Educational Robotic Simulators: A Systematic Literature Review. *Nuevas Ideas en Inf. Educ. TIIE* **2015**, *20*, 340–350.
40. Technology, I.; Technology, I. Role and review of educational robotic platforms in preparing engineers for industry. *Maejo Int. J. Sci. Technol.* **2017**, *11*, 17–34.
41. Martins, F.N.; Gomes, I.S.; Santos, C.R.F. RoSoS—A free and open-source robot soccer simulator for educational robotics. In *Proceedings of the Communications in Computer and Information Science*; Springer: Uberlândia, Brazil, 2016; Volume 619, pp. 87–102.
42. GitHub—Ivanseidel/Robot-Soccer-Simulator: An Open-Source Soccer Simulator for Virtual Robots Tournaments. Available online: <https://github.com/ivanseidel/Robot-Soccer-Simulator> (accessed on 19 December 2020).
43. UC Davis Center for Integrated Computing and STEM Education»RoboPlay Challenge Competition. Available online: <https://c-stem.ucdavis.edu/roboplay/challenge/> (accessed on 21 November 2020).
44. Cheng, H.H. *Learning Robot Programming with Linkbot for the Absolute Beginner*, 7th ed.; UC Davis Center for Integrated Computing and STEM Education: Davis, CA, USA, 2018.
45. Silva Barbosa, A.; Barros Costa, E.; Rocha Olivi, L.; da Fonseca Braga, A. Robot One: A cross-platform and multi-language simulator developed for educational robotics. *Galoa Events Proc.* **2019**, *1*, 108240.
46. GitHub—AlexanderSilvaB/Robot-One: “Robot One” is a Robotics Simulator Created by Alexander Silva Barbosa as a Graduation Work at the Federal University of Juiz de Fora—UFJF. Available online: <https://github.com/AlexanderSilvaB/Robot-One> (accessed on 19 December 2020).
47. Unity Real-Time Development Platform | 3D, 2D VR & AR Engine. Available online: <https://unity.com/> (accessed on 18 October 2020).
48. Alves, M.; Sousa, A.; Cardoso, Â. Web Based Robotic Simulator for Tactode Tangible Block Programming System. In *Proceedings of the Advances in Intelligent Systems and Computing*; Springer: Porto, Portugal, 2020; Volume 1092, pp. 490–501.
49. AST (Abstract Syntax Tree). AST (Abstract Syntax Tree) is a graph ... | by Dinis Cruz | Medium. Available online: <https://medium.com/@dinis.cruz/ast-abstract-syntax-tree-538aa146c53b> (accessed on 9 November 2020).
50. De, A.; Do, N. Robotic Simulator for the Tactode Tangible Block Programming System. 2019. Available online: <https://repositorio-aberto.up.pt/bitstream/10216/123177/2/361366.pdf> (accessed on 1 January 2021).
51. De Lima, P.V.S.G.; Bezerra, M.H.R.A.; De Sousa Tavares, A.C.; José Roberto Fonseca, J.; Teixeira, J.M.X.N.; Cajueiro, J.P.C.; Melo, G.N.; Henriques, D.B. Improving Early Robotics Education Using a Line-Following Robot Simulator. In *Proceedings of the 15th Latin American Robotic Symposium 6th Brazilian Robotic Symposium 9th Workshop on Robotics in Education LARS/SBR/WRE 2018*, Paraiba, Brazil, 6–10 November 2018; pp. 554–561. [CrossRef]
52. GitHub—Maracatronics/3pisimulator: A Pololu’s 3pi Simulator Fully Compatible with Arduino’s IDE Code. Available online: <https://github.com/maracatronics/3pisimulator> (accessed on 19 December 2020).
53. Karalekas, G.; Vologiannidis, S.; Kalomiros, J. EUROPA—a ROS-based open platform for educational robotics. In *Proceedings of the 2019 10th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, IDAACS 2019*, Metz, France, 18–21 September 2019; Institute of Electrical and Electronics Engineers Inc.: Piscataway, NJ, USA, 2019; Volume 1, pp. 452–457.
54. Buy a Raspberry Pi 3 Model B+—Raspberry Pi. Available online: <https://www.raspberrypi.org/products/raspberry-pi-3-model-b-plus/> (accessed on 19 October 2020).
55. ROS/Introduction—ROS Wiki. Available online: <http://wiki.ros.org/ROS/Introduction> (accessed on 18 October 2020).
56. Costa, V.; Cebola, P.; Tavares, P.; Morais, V.; Sousa, A. Teaching Mobile Robotics Using the Autonomous Driving Simulator of the Portuguese Robotics Open. In *Proceedings of the Advances in Intelligent Systems and Computing*; Springer: Porto, Portugal, 2020; Volume 1092, pp. 455–466.
57. Portuguese Robotics Open—INESC TEC. Available online: <https://www.inesctec.pt/en/events/portuguese-robotics-open#about> (accessed on 21 November 2020).
58. GitHub—ee09115/conde_simulator: Autonomous Driving Simulator for the Portuguese Robotics Open. Available online: https://github.com/ee09115/conde_simulator (accessed on 19 December 2020).
59. Bocca, L.F.; Leite, J.B.; Mantovani, S.C.A. Simulator of mobile robots controlled by Artificial Neural Networks to learning courses in robotics. In *Proceedings of the 2020 XIV Technologies Applied to Electronics Teaching Conference (TAEE)*, Porto, Portugal, 8–10 July 2020; pp. 1–7. [CrossRef]
60. Bermúdez, A.; Casado, R.; Fernández, G.; Guijarro, M.; Olivas, P. Drone challenge: A platform for promoting programming and robotics skills in K-12 education. *Int. J. Adv. Robot. Syst.* **2019**, *16*, 1–19. [CrossRef]
61. Drone Challenge—Escuela Superior de Ingeniería Informática. Available online: <http://blog.uclm.es/esiidronechallenge/> (accessed on 21 November 2020).
62. Simulink—Simulation and Model-Based Design—MATLAB & Simulink. Available online: <https://www.mathworks.com/products/simulink.html> (accessed on 19 October 2020).
63. Vega, J.; Cañas, J.M. PiBot: An open low-cost robotic platform with camera for STEM education. *Electronics* **2018**, *7*, 430. [CrossRef]
64. Base/src/drivers/gazebo/plugins/pibot at Master JdeRobot/base GitHub. Available online: <https://github.com/JdeRobot/base/tree/master/src/drivers/gazebo/plugins/pibot> (accessed on 19 December 2020).
65. Maloney, J.; Resnick, M.; Rusk, N.; Silverman, B.; Eastmond, E. The scratch programming language and environment. *ACM Trans. Comput. Educ.* **2010**, *10*, 1–15. [CrossRef]

66. Rafael, A.; Santos, C.; Duque, D.; Fernandes, S.; Sousa, A.; Reis, L.P. Development of an AlphaBot2 Simulator for RPi Camera and Infrared Sensors. In *Proceedings of the Advances in Intelligent Systems and Computing*; Springer: Porto, Portugal, 2020; Volume 1092, pp. 502–514.
67. GitHub—Ssscassio/alphabot2-simulator: Alphabot2 simulator using ROS and Gazebo Simulator. Available online: <https://github.com/ssscassio/alphabot2-simulator> (accessed on 19 December 2020).
68. Papelis, Y. *Experimental Validation of a Ground Robot Simulation Model During line Following Task*; MODSIM: Norfolk, VA, USA, 2017; pp. 1–11.
69. Available online: <https://www.parallax.com/sites/default/files/downloads/28985-Ping-Kit-for-Stingray-Documentation.pdf> (accessed on 18 October 2020).
70. Peralta, E.; Fabregas, E.; Farias, G.; Vargas, H.; Dormido, S. Development of a Khepera IV Library for the V-REP Simulator. *IFAC PapersOnLine* **2016**, *49*, 81–86. [[CrossRef](#)]
71. Inventor | Mechanical Design & 3D CAD Software | Autodesk. Available online: <https://www.autodesk.com/products/inventor/overview?plc=INVPROSA&term=1-YEAR&support=ADVANCED&quantity=1> (accessed on 19 October 2020).
72. Fabro, J.A.; Junior, E.T.P.; Dias, A.F.G.P.; Skora, L.E. Programming teaching using flowcharts in a simulated environment focused on introducing practical OBR. In *Proceedings of the 2019 Latin American Robotic Symposium 2019 Brazilian Robotic Symposium 2019 Workshop on Robotics in Education LARS/SBR/WRE*, Rio Grande, Brazil, 22–26 October 2019; pp. 453–458. [[CrossRef](#)]
73. OBR—Olimpíada Brasileira de Robótica 2020. Available online: <http://www.obr.org.br/> (accessed on 21 November 2020).
74. Aroca, R.V.; Pazelli, T.F.; Tonidandel, F.; Filho, A.C.A.; Simes, A.S.; Colombini, E.L.; Burlamaqui, A.M.F.; Goncalves, L.M.G. Brazilian Robotics Olympiad: A successful paradigm for science and technology dissemination. *Int. J. Adv. Robot. Syst.* **2016**, *13*, 1–8. [[CrossRef](#)]
75. Rahul, R.; Whitchurch, A.; Rao, M. An open source graphical robot programming environment in introductory programming curriculum for undergraduates. In *Proceedings of the 2014 IEEE International Conference on MOOC, Innovation and Technology in Education (MITE 2014)*, Patiala, India, 19–20 December 2014; pp. 96–100. [[CrossRef](#)]
76. Minibloq: Download. Available online: <http://blog.minibloq.org/p/download.html> (accessed on 19 December 2020).
77. Barbosa, A.S.; De Oliveira, A.P.; Silva, A.R.; Boas, A.S.C.A.V.; Boas, E.R.V.; Felipe De Almeida, F.D.S.A.; E Souza, J.V.A.; Boas, M.A.V. Design and development of a manipulator for educational Robotics. In *Proceedings of the 15th Latin American Robotic Symposium 6th Brazilian Robotic Symposium 9th Workshop on Robotics in Education LARS/SBR/WRE 2018*, Paraiba, Brazil, 6–10 November 2018; pp. 522–527. [[CrossRef](#)]
78. Available online: <http://sistemaolimpo.org/midias/uploads/00e350e39d80891c6e832bb76eda42b8.pdf> (accessed on 31 December 2020).
79. Portal Unicamp | Unicamp. Available online: <https://www.unicamp.br/unicamp/> (accessed on 20 December 2020).
80. Unesp—Universidade Estadual Paulista—Portal. Available online: <https://www2.unesp.br/> (accessed on 20 December 2020).
81. Parrot Developers. Available online: <https://developer.parrot.com/> (accessed on 20 December 2020).