

Educational Robots as Pedagogical tools in Computer Science Extended Degree Introductory Course

Josh di Bona
University of Cape Town
Rondebosch, Cape Town
Private Bag X3, 7701
(+27) 836618209
dbnjos001@myuct.ac.za

Gary Stewart
University of Cape Town
Rondebosch, Cape Town
Private Bag X3, 7701
(+27) 216502665
gstewart@cs.uct.ac.za

ABSTRACT

There has been a significant increase in the interest and use of robotics in education over the last sixteen years. Many studies have been done, this paper/review looks at all the current research to analyze what has been done, what was done right and what can be improved in order to provide an effective educational robotic system for the introductory computer science extended degree course at the University of Cape Town. Looking at what robots have been used, or especially designed, for education and/or teaching programming, what features were used, how were the robots used and if there was any software used alongside the robots. It was found that robots are quite effective at increasing the motivation of students in the class and their interest in the subject, but there hasn't been any significant evidence proving that they help the students learn and many studies use questionable evaluation and testing methods. It has become a standard practice to use a simulator or some other sort of software alongside the robot if not every student can be given a robot as the code-test-debug cycle is vital to learning. As well as the use of a framework that provides abstracted high-level methods to teach simple concepts. This paper concludes that the idea of using robots as a pedagogical tool is still a fairly new concept and its effects on education and learning haven't been fully tested to see where its true benefits lie, suggestions are made for further testing.

CCS Concepts

•Social and professional topics→CS1 • Computer systems organization→Robotics

Keywords

Educational Robot; Computer Science, Programming; Pedagogical tools; CS1; Undergraduate; Simulator; Python; UCT.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

Month 4–10, 2016, University of Cape Town (UCT), South Africa

1. INTRODUCTION

This review is being done for a project which involves the creation of an educational robotic system for the purpose of assisting in teaching an introductory computer science extended degree course at the University of Cape Town. In order to design the best system, a review of all the current designs, research and experiences of using robotics in education, robots in teaching computer science and more specifically, robots in CS1 is necessary. This review aims to point out any commonly made errors and help identify what has been proven to work. Also show which areas of using robotics is lacking testing or has had insufficient or incomplete testing.

This review looks at robotics in education as a whole, which involves many subfields. The sections looked at are Robotics in education, which consists of the different robotic systems currently available, the different levels of education and subjects which use robotics as a pedagogical tool [1], the considerations that must be taken into account when using robots, how they are implemented and what benefits are known from using robots. Next the current educational software available, simulators and virtual environments that are used to either teach computer science or used with or instead of the robots are looked at. Robotics in teaching computer science is reviewed and the different concepts that are taught using robots and what methods are used to teach them are observed and finally, robots in CS1, introductory computer science courses. Some of the current applications of robots in CS1 is looked at and how their effects were evaluated and what were the general findings found.

There has been a significant increase in interest in robotics and the use of robotics in education. Robots are currently being used from the ages of nine [9] [14], all the way to students using robots for their master thesis [5]. Robots have the unique ability of being used as an educational tool [3] and being a subject of education themselves [8][4]. In most cases, robots are used for hands-on education, to get a physical, real world experience of a learned concept [1]. Karel the Robot, a simple robotic simulator is used back before robots were a reality and as widespread as today [27]. Soon after came the emergence of robots such as the LEGO RCX and recently the LEGO Mindstorm become available to the public, although at a high cost. With the exponential increase in technology and the reduction in cost over the last couple years' robots are quickly becoming a feasible option as a pedagogical tool, a tool used to learn or teach [1]. Robots are currently being used in many levels of education with a broad range of subjects, although mainly science, technology, engineering and math [9]. Many different robots are used and different features provide many convenient ways to teach basic concepts, features such as different motors on the robot, proximity sensors and cameras.

2. ROBOTS IN EDUCATION

2.1 Robotic Systems

The following covers some of the robotic systems looked at during this review, this won't include every educational robot currently available, but rather get a general look at what is being used, what they are being used for, whether they were bought or designed especially, what features the robot has and how much they cost, if made available.

The first robotic system is one used by faculty members at Tufts University to teach engineering principles to students, such as experimentation techniques, how sensors work and concepts such as control loops. LEGO Bricks are used to build the robots and the LEGO RCX, the old microprocessor, is used as the brain of the robot. Groups of three were formed and each given a set of parts to work with [5].

LEGO kits are used with Handey boards to teach an undergraduate introduction to robotics course. Handey boards take in more external input from sensors than the LEGO Mindstorms and are more capable and general, allowing for more advanced computation. Evolution Robotics SDK with vision capabilities and ActivMedia Pioneer robots are used to test algorithms as they can first be run on a simulator then without much editing, run on the robots [4].

The e-puck is a robot that was designed for teaching a wide range of engineering disciplines. It was designed with many requirements in mind such as size, cost and available features. It's small enough to use on a desk and cost about R4000 to develop (sells at R9000) – with the current exchange rate of R16.24 to the Euro. This proved to be much cheaper than those currently on the market which ranged from R1600 to up to R74000. The e-puck is fitted with many features, such as cameras, speakers, mics, led sets. It also has wired and wireless capabilities and sensors to tell the distance to objects. The e-puck is used for quick high-level disciplines such as behavior based robotics, automatic control, path finding and distributed intelligent systems. [6]

The more modern LEGO Mindstorms are used in secondary school to replace the more traditional educational tools that are used. In an example it is used to teach technology to a group of twenty secondary school students, they were divided into groups of four each with a set of parts. Each set of parts came with a set of motors, sensors and wheels, along with instructions of a desired set of functionality. [8]

Another robot system which was designed for a specific course, is used to teach 3rd and 4th graders. It is made up of multiple robots with infrared sensors so that they can follow lines and detect line crossings, a web-based programming interface accessible with Wi-Fi and a java enabled browser. A Raspberry Pi is used as the multi-robot controller which acts as a wireless access point for the user and the robots (it connects to the robots via Bluetooth). The firmware contains a single communication protocol, allowing students to control the robots with a sequence of simple commands (blocks of code), classes are split into groups of five to six each with a robot [9].

The Khepera II robot was used to teach graphical user interface (GUI) programming to a computer science class at California State University. It's moderately expensive, small and round, with eight infrared proximity detectors. It was used along with a server for communications between the students and the robot. The robot also provides an interface that abstracts the low-level details needed to control the robots, and even a higher-level interface

which is also available which contains commands that need no parameters [12].

The Parallax Scribbler was made with the idea in mind that a course can be taught where every student is given a robot. Because of this, the Scribbler was designed to be inexpensive, portable and robust. There are quite a few features available on the robot, such as multiple motors, proximity sensors, line sensors, programmable LEDs, speaker, Bluetooth device, a color camera and a pen port. The robot works with the Pyro/Myro software to provide a helpful interface/framework to control the robot with simple commands. The robot along with the software is used to teach CS1 much like many others have done. [17] [23].

Pyro, or more recently Myro, although not a physical robot, it is an essential part of a robotic system. Pyro is an API, or an interface framework developed to help students write platform-independent robot programs. It includes a set of robot control functions and additional helper functions to introduce students to more complex concepts as well as the basic concepts. The low-level robot controlling functional are abstracted into high-level simple commands which can be used to control a range of robots with little or no editing of the code. These abstractions also help making the writing of the code, which controls the behavior of the robot, independent to the type, size or weight of the robot. Pyro works with many robots and has even started working with simulators, meaning that you can use the API, code programs, test them on a simulator then run them on the robot. [11]

2.2 Considerations

A few considerations have to be made when choosing a robotic system to use and when designing a course that uses robots as a pedagogical tool. One has to first take into account the age of the users as well as the material that is being taught as mentioned above. Another is to take into account the gender of the student. It was found that both boys and girls are equally as motivated to use robots but they choose to use them in different ways, for example boys are happy to code robots that play soccer while girls would be more willing and motivated to code robots that dance [2]. So careful thought has to go into designing a course with tasks that cater for both genders.

2.3 Implementation

In order to implement an educational robotic system, one needs to take into consideration a few things. After reviewing a few papers and past studies, the following points have been raised. Robots take up space, especially in order to test them correctly. In some cases, a dedicated amount of space was cornered off to test the robot application [12] while in other cases, such as the e-puck, a small enough robot was designed so that it could theoretically be tested on the student's desk [6].

It is clear that already built educational robots can range from anywhere around R4000 to up to R70000. So decisions have to be made in terms of the level of features, amount of features and the number of robots needed. If the goal is to give every student in a class of 500 a robot, a huge budget or a really cheap robot will be needed.

One of the most important things to look at when implementing a robotic system is how the student is going to interact with it. If every student can't be given their own robot kit, then an alternative way to give access to the students outside of dedicated lab times must be found [19] [26]. One solution is to provide the students with a simulator to run their code before they run it on the robots, or alternatively, let them be able to access and run code on the robots from a remote location, using a server and a web-

cam [12]. At minimum, implement a way of controlling the robot wirelessly, getting the robot to run the code without having to physically put it on the robot, using either Bluetooth or Wi-Fi [18] [13].

One of the things which has an effect on the implementation of the robot system is how it is going to be graded, keep in mind that because robots run out of battery, they constantly perform differently, which could teach students at a young age about noise and external effects on their programs. One the same note, one must implement the system in such a way that batteries don't have to be regularly replaced [23].

Choosing the correct framework or API to use with your robot is also an important decision when setting up the system. The API allows for the abstraction of all the low-level code and commands that moves the robot's motors and retrieves information from the sensors and other features. A few frameworks are discussed in this paper, the most popular being Pyro or Myro, which provides an API which can be used to run the same code on multiple robots. It also provides additional features that start to introduce the students to more complex concepts. So the level of education has to be evaluated and the types of concepts that are trying to be taught, to make sure that the API has functions and methods that will allow for the concept to be taught effectively.

2.4 Benefits

Below the benefits of using robots in teaching that have been found in the past are observed and analyzed.

Most papers easily show through either observation or student surveys that the use of robots increases the motivation of the students in class and in the particular subject being taught [9] [5]. But aside from the benefit of extra motivation, the question still remains about what other benefits there are, and what benefits haven't yet been proven to be true. Robotics is suggested to work hand in hand with problem based learning [8]. It can be used to help recruit students into the respective STEM fields [4]. One of the most significant questions proposed in multiple papers was whether robots are just a temporary interest that will soon get replaced, and whether the idea that they actually help with learning is true [3] [2]. This question is difficult to answer, and will probably only be answered in a few years when they are still around. Although it was difficult to answer, it was possible to conclude that robots have a clear benefit as a tool in some subjects, such as AI and multi-agent systems, where problems and solutions can be viewed in the real world [3]. It was also possible to come to the conclusion that yes robots might get replaced, but it is clear that they have a sustained potential in education and have a unique hands-on experience that not many other tools have [2]. Robotics has only recently gone public and it is still the early stages of using robotics in education [4].

Using robotics also seemed to create an open environment in which students were more willing to discuss issues and ask for advice [5]. Robots can also be used to teach with new methods, such as seminar courses instead of lecture courses where students are encouraged to do their own research and exploration into the topic and learn by doing [4].

3. EDUCATIONAL SOFTWARE

3.1 Simulators & Visual Environments

Aside from the growing rate of robots in teaching STEM subjects, many universities and curriculums use 2D or 3D simulators, either with or without physical robots [28]. This is either because robots weren't accessible yet [27] or because the simulator seems to be

doing a good enough job at visualizing the concepts being learnt. In some cases, simulators are even used to test the students code before they run it on the actual robot [26]. A few simulators are covered below.

Scratch is a visual programming environment, usually used by children aged eight to sixteen or student's who's major isn't in CS. It allows students to learn the basic concepts of programming using interactive animated stories and games. It is used to introduce programming to those who have no previous coding experience [30].

USARSim is a high fidelity robot simulator made for the use in research and well as education, and more commonly used in competitions such as Robocup, a competition where teams program robots to play soccer against one another. The simulator is a set of models and classes that define the robot, sensors and actuators. Its built in Unreal Engine, a game engine, and provides a 3D environment which the user can make and test their robots. This type of simulator is suggested to be better than regular 2D visual environments such as Karel, as it provides much more detail, in the models and in that of the environment, and their physical interactions. Which make it an ideal simulator to learn about robots and the use of sensors in an environment [28].

Alice is a 3-D interactive animation environment, created to teach introductory programming concepts. It was found that aside from students that are new to programming, some students struggle to understand the simple concepts of problem-solving using a computer. It was suggested that a visual environment might help them to understand the sequence of code and how their program is being executed, and help them to better debug the code. Alice comes with many abstracted built in commands much like many educational robot systems do. Alice can help teach named instructions, functions, decisions, recursion/looping, events/interactions and program language constructs. Students were found to be comfortable with using objects and invoking methods and were able to easily watch their code run and effectively debug it [29].

The python turtle module is currently being used at UCT in the extended degree program for computer science first year. It is a 2D graphical interface in which students can easily observe their code. It contains methods such as `turtle.forward(dist)` and `turtle.left(90)` which move the turtle forward and turn the turtle 90 degrees. Students can draw patterns using the turtle which quickly helps to see visually where their code went wrong [31].

4. ROBOTICS IN TEACHING COMPUTER SCIENCE

4.1 Concepts

Computer science can be abstracted down into many basic concepts, which can be taught using an educational robot as a tool. This is usually done in introductory computer science courses at undergraduate level or at school level. Some of these concepts are, variables, constants, iteration statements, control flow statements, conditionals, functions and basic data types [25] [23] [19]. Besides from teaching the basic concepts of programming, in an introductory programming course you also want to teach things such as problem-solving skills and critical thinking [22].

Other than teaching basic concepts, it was found that robots also come in handy when teaching more complex concepts such as AI and multi-agent learning [10] [11]. In most cases, the robots and the frameworks, if any, are built in such ways that they can be

used for learning more complex concepts, or even come with methods in the framework to help simplify the process [15]. Modules that enable and encourage the exploration of more complex robot control paradigms, such as robot learning and robot vision [11].

4.2 Methods

When designing an educational system, it involves not only the robot, its features and a framework but also how this system is going to be used and how it will fit into the curriculum. Below are a few ways in which robots are used to teach basic concepts.

One of the main methods, and probably the most important in a hands-on, problem-based subject that computer science is, is the design, implementation, testing and debugging of the code. Or better known as the write-run-debug feedback loop [21] or the compile-run-debug cycle [19] when learning. Therefore, this method must either be enhanced or retained when robots are added. The negative findings made by Fagin in his study are shown to be because of the fact that students couldn't work on their code at home or back in their rooms, only during lab sessions, so the whole process of writing code, testing it and debugging it was taken away from them. It was found that students either asked to have robots to take home or to be given some sort of simulator to use at home [19] [21]. These findings are eventually taken into account, where there are robots and a simulator, and the code written on/for the simulator will run without much editing on the robot [4]. Thus allowing students to test their code without the need of the robot.

Summit talks about the approach they took in using robots in CS1, which is a similar approach taken by most curriculums using robots. Classes or lectures are given to students on a concept, such as sequential ordering of statements, program flow, and encapsulation in functions. After which they are given an assignment which requires programming some sort of behavior for the robot. While they are busy on this assignment, they are given a lecture on the next concept and so on. At the end of the semester they are given a project which encapsulates all the learned concepts. [17]

It was mentioned already that using robots can also lead to seminar lectures on robotics, which encourages learners to choose their own project and do their own research into the topic. However, this may not be suitable to use in introductory programming courses but rather in an engineering course where the students can build and code the robots. [4]

4.3 Current Applications in CS1

In a study that is looking at using robots for education in a simplified introductory programming course, it is important to consider some of the literature on previous experiences of using robots in CS1 and the various theories that have been tested. The past and current studies look at using either software or robots, or both to either improve student motivation, or student learning or encourage students to take up CS as a major.

Fagin and Merkle present an approach to teaching the basics of computer science using robots, specifically LEGO Mindstorms. An interface was made providing a high-level language to the user for controlling the robot and methods made to teach each basic concept. The study was a year-long experiment and compared the results of over 800 students from both robotic and non-robotic lab-sessions. Students were given lab-sessions where they were allowed to work on the robots, thus preventing them from being able to work on them at home. The results from this study and mentioned and discusses later in this study. [20]

Summet et al. developed a CS1 curriculum that uses robotics, the Scribbler robot is used along with a range of features. The curriculum was developed to correct the findings made by Fagan in his study of the effect of robots. Every student was given a robot with all the features and the Myro (Pyro) API software.

Saad et al. introduced robots into the curriculum in order to increase the number of students taking CS as a major, as well as increase student motivation. The Ridgesoft Intellibrain-Bot, which is programmable using Java was chosen. [25]

4.4 Evaluation Methods

After reading a few studies that were done on introductory computer science courses to test the effect that introducing the use of robots has, some of the main methods used in evaluating this effect are identified. This is an important part of a study to look at, as it can determine whether the data and conclusions made are completely valid or not, as well as to see if there are any discrepancies in findings when different evaluation methods were used.

In some cases, a survey or questionnaire is given to a group of students before and after a course or task using the robots [23], in others a survey is just given afterwards [15]. In the majority of studies, the marks of the students are looked at in the previous years that didn't use robots to the year that did [19], and conclusions are made based on these changes. Another evaluation method, for those studies testing whether the use of robots can convince students to take up CS as a major, looks at the number of students who declared CS as a major before and after the use of robots in the course.

M. McGill goes into a depth analysis that the use of robots in education increases the motivation of the students involved, the theory is put to test, other papers analyzed for flaws and areas for future work surrounding this topic are suggested. It is pointed out that when looking at the effects on motivation, it is important to take into account the different characteristics such as gender, technical self-perception, and interest in development. The study points out several indicators used to measure motivation that are tested across different studies, including things such as attention, the value the learner places on a task and the perceived relevancy of the task [24]. One main critique she has concerning other papers, is that none of them seem to use a standard way of measuring motivation, and none of them use accredited questionnaires. Many tools do exist to measure motivation, but only a few of them meet the criteria for being valid and reliable. The Computer Programming Self-Efficiency Scale is an example of a validated instrument used to measure factors related to student success in introductory computer science courses.

4.5 Findings

Many different studies have been analyzed and studies found different results. The following can be concluded about the use of robotics in university introductory computer science courses (CS1).

Students spent extra time on class-related work that wasn't required, which was self-directed and showed an increase in motivation and interest in the subject [23].

Mcgill found that that the use of robots had a positive effect and influenced the student's attitudes towards learning but found it had little or no effect on relevance, confidence or satisfaction. Fagin found negative results in his study which is believed to be because of the lack of a sufficient code-test-debug loop, as the students couldn't test the robots at home or in their free time.

However, both found that the idea of programming with robots captured the attention of the students and this in turn increased their motivation in the classroom [19] [24].

It must also be noted that most papers give results such as, most students enjoyed the course or x % of students found the course and use of robots helpful. This gives no insight or helpful information when doing a validated in depth analysis of the effects of robots on teaching and learning.

5. DISCUSSION

The use of robotics in education but especially in introductory computer science courses has many benefits, some validated and clear and others not. But overall it is clear that there are benefits in using robotics, when used correctly. The reported benefits consist of effects such as an increase in motivation, encouraging students to take up CS as a major, increasing student interest and encouraging self-learning. There is however not much evidence of robotics actually increasing the effectiveness of how the concepts are presented to the students and it increasing their understanding and marks.

Because of the inconsistencies in the way studies have been done and the evaluation of the effect of robots has been done, it is not known whether software has the same effects on motivation as physical robots. It was found that there was only a slight increase in student motivation, no long term sustained effect with non-majors. Measuring the actual effects of robots is difficult, some papers have looked at student marks over years, those taught with and without robots, while others looked at the number of students who declared a CS major. Other studies used questionnaires or surveys to measure the effect that robots had, some did this at the start and end of a course and others did this after using the robots for a single workshop. It has been noted that there have been biased questionnaires and questionnaires and surveys made that weren't ever validated, and on top of that no control group used when comparing the results of the surveys. McGill suggests that motivation is a complex notion, consisting of attention, relevance, confidence and satisfaction, and suggests a validated questionnaire to test all of these concurrently to get more accurate results. Creating your own validated questionnaire is too complex of a task. [24]

The motivation for this paper is to look into current research for the use of robots in education and specifically introductory computer science courses, in order to create an educational robotic system for the use in a CS1 extended degree course. This course is aimed at students struggling to understand simple basic concepts on a deeper level [32] and rather than teaching programming, students should come from the course with a solid understanding of the scope of computing and the role of programming [16]. There is a lack of a robot and software that cater for struggling students specifically. Students bring with them a wide spectrum of backgrounds [29] and this causes CS1 general courses to contain a broad mixture of students with varying abilities. All of the current research, and those analyzed in this paper, have been done on CS1 or similar populations. There is a lack of testing to see the effects of using robots to teach basic concepts to those struggling to understand them and it gives the opportunity to be able to see the true benefits in a focused test group.

When looking at the different features that are used throughout all the studies that have been done, there are many different features used on the different devices. Features such as proximity sensors, cameras, mics, speakers, line detectors and others. These features are generally used to create more abstracted methods used in

teaching the programming concepts. Unfortunately, there doesn't seem to have been any research done into looking which features are the best to use for teaching each concept and which few would be the most beneficial for a low cost robot on a budget.

It would appear that most institutions teaching introductory computer science courses are either using a robot simulator or some other software visualization tool to assist with the teaching of basic concepts.

It has been found that using robots in a class where the students can only test their code when they have a lab session didn't work as it removed the code-test-debug cycle [19]. Some researchers chose to use a simulator along with a robot, that the students can use to test their code at home, and run the same code on the robot when available [12]. One interesting approach is the use of a Khepera II robot, where a server and webcam are set up for the students to run their code at home wirelessly on the robot.

6. CONCLUSIONS

From the different areas that make up an educational robotic system, it can be concluded that there are areas where best practice has been found, such as having a simulator or some sort of set up that allows students to test their code in their free time, if every student can't be given their own personal robot. The ways in which robots are used as a pedagogical tool has been well defined, the only thing that hasn't been explored in full, is the different available features and what benefits each could have. It is now known that an API or some sort of framework has to be used or made in order to abstract low-level commands and create simple methods for teaching concepts, but other methods can be made to introduce more complex concepts as well.

Robotics as a pedagogical tool in education is still a fairly new concept, it doesn't have well defined benefits but seems to be a worthwhile investment to look into. It is clear from this study that there has been a lot of research done in the area and many experiences had. But because it is still fairly new, there is no agreed upon method yet about testing its effects and benefits and how testing should take place. This study has provided a good insight into a few positive and negative approaches to using robots to teach basic computing concepts and suggestions have been made. This knowledge will be used to develop an all rounded educational robotic system to be used in an CS1 extended degree program, and perhaps look into the effects that it has on this focused group using more validated testing and evaluation methods.

7. REFERENCES

- [1] Alimisis, D. 2012. Robotics in education & education in robotics: Shifting focus from technology to pedagogy. In *Proceedings of the 3rd International Conference on Robotics in Education.*, 7-14.
- [2] Johnson, J. 2003. Children, robotics, and education. *Artificial Life and Robotics*, 7, 1-2 (Jan. 2003), 16-21.
- [3] Malec, J. 2001. Some thoughts on robotics for education. In *2001 AAAI Spring Symposium on Robotics and Education.*
- [4] Mataric, M. J. 2004. Robotics education for all ages. In *Proc. AAAI Spring Symposium on Accessible, Hands-on AI and Robotics Education.*
- [5] McNamara, S., Cyr, M., Rogers, C. and Bratzel, B. 1999. LEGO brick sculptures and robotics in education. In *ASEE Proc.*

- [6] Mondada, F., Bonani, M., Raemy, X., Pugh, J., Cianci, C., Klapotocz, A., Magnenat, S., Zufferey, J., Floreano, D. and Martinoli, A. 2009. The e-puck, a robot designed for education in engineering. In *Proceedings of the 9th conference on autonomous robot systems and competitions*. IPCB: Instituto Politécnico de Castelo Branco, 59-65.
- [7] Mubin, O., Stevens, C. J., Shahid, S., Al Mahmud, A. and Dong, J. 2013. A review of the applicability of robots in education. *Journal of Technology in Education and Learning*, 1(Jan. 2013), 209-0015.
- [8] Pastor, J. 2012. Robotics in education. *Proceedings in ARSA-Advanced Research in Scientific Areas*, 1 (Jan. 2012).
- [9] Saleiro, M., Carmo, B., Rodrigues, J. M. and du Buf, J. H. 2013. A low-cost classroom-oriented educational robotics system. *Social Robotics*, 74-83.
- [10] Blank, D. 2006. Robots make computer science personal. *Commun ACM*, 49, 12 (Dec. 2006), 25-27.
- [11] Blank, D., Kumar, D., Meeden, L. and Yanco, H. 2004. Pyro: A python-based versatile programming environment for teaching robotics. *Journal on Educational Resources in Computing (JERIC)*, 4, 3 (Mar. 2004), 3.
- [12] Challinger, J. 2005. Efficient use of robots in the undergraduate curriculum. *ACM SIGCSE Bulletin*, 37, 1 (Jan. 2005), 436-440.
- [13] Goh, H. and Aris, B. 2007. Using robotics in education: lessons learned and Learning experiences. In *1st International Malaysian Educational Technology Convention.*, 1156-1163.
- [14] Magnenat, S., Shin, J., Riedo, F., Siegwart, R. and Ben-Ari, M. 2014. Teaching a core CS concept through robotics. In *Anonymous Proceedings of the 2014 conference on Innovation & technology in computer science education*. ACM, 315-320.
- [15] Saad, A. and Kroutil, R. M. 2012. Hands-on learning of programming concepts using robotics for middle and high school students. In *Proceedings of the 50th Annual Southeast Regional Conference*. ACM, 361-362.
- [16] Xu, D., Blank, D. and Kumar, D. 2008. Games, robots, and robot games: complementary contexts for introductory computing education. In *Proceedings of the 3rd international conference on Game development in computer science education*. ACM, 66-70.
- [17] Summet, J., Kumar, D., O'Hara, K., Walker, D., Ni, L., Blank, D. and Balch, T. 2009. Personalizing CS1 with robots. *ACM SIGCSE Bulletin*, 41, 1 (Jan. 2009), 433-437.
- [18] Burbaite, R., Stukys, V. and Damasevicius, R. 2013. Educational robots as collaborative learning objects for teaching Computer Science. In *System Science and Engineering (ICSSE), 2013 International Conference on*. IEEE, 211-216.
- [19] Fagin, B. S. and Merkle, L. 2002. Quantitative analysis of the effects of robots on introductory Computer Science education. *Journal on Educational Resources in Computing (JERIC)*, 2, 4 (Apr. 2002), 2.
- [20] Fagin, B. S., Merkle, L. D. and Eggers, T. W. 2001. Teaching computer science with robotics using Ada/Mindstorms 2.0. *ACM SIGAda Ada Letters*, 21, 4 (Apr. 2001), 73-78.
- [21] Fagin, B. and Merkle, L. 2003. Measuring the effectiveness of robots in teaching computer science. *ACM SIGCSE Bulletin*, 35, 1 (Jan. 2003), 307-311.
- [22] Goldweber, M., Congdon, C., Fagin, B., Hwang, D. and Klassner, F. 2001. The use of robots in the undergraduate curriculum: experience reports. *ACM SIGCSE Bulletin*, 33, 1 (Jan. 2001), 404-405.
- [23] Markham, S. A. and King, K. 2010. Using personal robots in CS1: experiences, outcomes, and attitudinal influences. In *Proceedings of the fifteenth annual conference on Innovation and technology in computer science education*. ACM, 204-208.
- [24] McGill, M. M. 2012. Learning to program with personal robots: Influences on student motivation. *ACM Transactions on Computing Education (TOCE)*, 12, 1 (Jan. 2012), 4.
- [25] Saad, A., Shuff, T., Loewen, G. and Burton, K. 2012. Supporting undergraduate computer science education using educational robots. In *Proceedings of the 50th Annual Southeast Regional Conference*. ACM, 343-344.
- [26] Schumacher, J., Welch, D. and Raymond, D. 2001. Teaching introductory programming, problem solving and information technology with robots at West Point. In *Frontiers in Education Conference, 2001. 31st Annual*. IEEE, F1B-2-7 vol. 2.
- [27] Becker, B. W. 2001. Teaching CS1 with karel the robot in Java. *ACM SIGCSE Bulletin*, 33, 1 (Jan. 2001), 50-54.
- [28] Carpin, S., Lewis, M., Wang, J., Balakirsky, S. and Scrapper, C. 2007. USARSim: a robot simulator for research and education. In *Robotics and Automation, 2007 IEEE International Conference on*. IEEE, 1400-1405.
- [29] Cooper, S., Dann, W. and Pausch, R. 2000. Alice: a 3-D tool for introductory programming concepts. *Journal of Computing Sciences in Colleges*, 15, 5 (May. 2000), 107-116.
- [30] Maloney, J., Resnick, M., Rusk, N., Silverman, B. and Eastmond, E. 2010. The scratch programming language and environment. *ACM Transactions on Computing Education (TOCE)*, 10, 4 (Apr. 2010), 16.
- [31] Sanders, I. and Scholtz, T. 2012. First year students' understanding of the flow of control in recursive algorithms. *African Journal of Research in Mathematics, Science and Technology Education*, 16, 3 (Mar. 2012), 348-362.
- [32] Yadin, A. 2011. Reducing the dropout rate in an introductory programming course. *ACM inroads*, 2, 4 (Apr. 2011), 71-76.