Teaching Robot Programming Activities for Visually Impaired Students: A Systematic Review

Juliana Damasio Oliveira(✉), Márcia de Borba Campos,
Alexandre de Morais Amory, and Isabel Harb Manssour

Faculty of Informatics (FACIN), Pontifical Catholic University of Rio Grande do Sul (PUCRS), Porto Alegre, Brazil
juliana.damasio@acad.pucrs.br, {marcia.campos, alexandre.amory, isabel.manssour}@pucrs.br

Abstract. This paper presents a systematic review of studies concerning the use of robotics for the programming education of individuals with visual impairment. This study presents a thorough discussion and classification of the surveyed papers, including: different programming teaching methodologies based on robotics for people who are blind; the use of several robotics kits and programming environments; the evaluation procedure for each environment; and the challenges found during the teaching process. Based on these papers we created a guideline to prepare, conduct and evaluate a robot programming workshop for people who are visually impaired. These instructions include, for example, how to train instructors to work in workshops for people with visual disabilities, how to prepare concrete and digital support materials, suggestions of work dynamics for programming teaching, how to conduct collaborative activities, forms of feedback for the student to better understand the syntax and semantics of the language, recommendations for the development of a robotic environment concerning the hardware (robot) and software (programming language to operate the robot). These recommendations were validated with two users with visual impairment.

Keywords: Systematic review · Visual impairment · Teaching robot programming

1 Introduction

The first concepts of educational robotics have emerged with Solomon and Papert [1], with the establishment of the LOGO language which, through commands, allowed for the movement of a graphic turtle. The main idea of LOGO language was to encourage children to learn to program in a motivating and playful way [1].

Traditional teaching techniques mainly rely on visual models to help in the understanding of complex information, such as diagrams, flowcharts, tables, and images. Unfortunately, this type of teaching is not useful for visually impaired students [2]. Robots are being used to assist and to stimulate programming classes [3–5] and several robotics environments have been proposed to facilitate the teaching of programming.

M. Antona and C. Stephanidis (Eds.): UAHCI 2017, Part III, LNCS 10279, pp. 155–167, 2017.
DOI: 10.1007/978-3-319-58700-4_14

However, according to [2, 13, 14], there are few initiatives to involve students who are visually impaired because many programming environments are based on graphical user interface, for instance, using drag and drop features, without satisfying the accessibility and usability criteria. Thus, those users that rely on software assistive technology such as, for example, screen readers or screen magnifiers, experience usability difficulties. Another challenge in the programming activity for people who are blind refers to the syntax of some programming languages. Some languages use unusual operators, commands, expressions, tab, punctuation marks, graphic symbols, characters in lowercase and uppercase, etc., which are not always intuitive for beginner programmers and cannot be read by users who are blind and use assistive technology.

Regarding robotics, some kits require participation from people who are not visually impaired to assemble the robot and sometimes set the scenario where it will move around. Other kits use stylized robots, hindering the recognition of these parts and the robot's front, sides and rear. These characteristics, although visually pleasing, may cause difficulties concerning the robot's recognition and their relation in the space when users make use of touch.

A blind individual needs additional support from non-visual stimuli to perceive the environment and build up mental maps. Also known as cognitive maps, they are defined by Long and Giudice [6] to describe the way people create and mentally remember images of the distances and directions to places beyond the reach of their perception. In this manner, receiving information of space through other senses, such as hearing and touch, collaborates to the creation of mind maps to represent the environment [7]. Multimodal interfaces must be included to increase the user's ability to orient and navigate the robot within an environment [8, 9]. The programming environment must use different sound clues to describe the movements, the location of the robot, and the objects around it. Still, the robot must be easy to handle such that the user who is blind can recognize parts and confirm the sound information through tactile feedback. Moreover, although this was not confirmed in the literature and this evaluation is not the focus of this paper, the authors believe that robotics could help to develop (or improve) orientation and mobility skills in blind users.

In this context, this paper presents a systematic review of works regarding the use of robotics for the programming education of individuals with visual impairment. This review may ease the work of educators, researchers and developers, helping in understanding the teaching methodologies, the robotics and programming kits, and how to prepare, lead and evaluate a programming workshop for people with visual disabilities.

2 Methodology

A systematic literature review (SLR) was performed, following to the protocol of Kitchenham [10]. The goal of the SLR was to identify and understand methodological procedures which make use of robotics as support to the teaching of programming to people who are visually impaired. Through this goal, the primary (1) and secondary (a, b, c, d) research questions were identified:

1. Which methodological procedures are being used in the teaching of programming with robots for people with visual disabilities?
   (a) Which are the methodologies for teaching robot programming to people with visual disabilities?
   (b) What are the characteristics of the programming environments used by people with visual disabilities?
   (c) What are the examples of good practices for teaching robot programming to people who are visually impaired?
   (d) Which were the difficulties/limitations in the use of robotics as support to the teaching of programming to people who are visually impaired?

The SLR was conducted in 4 digital libraries relevant to the area of Computer Science, namely: ACM Digital Library (http://dl.acm.org), ScienceDirect (http://www.sciencedirect.com/), Scopus (https://www.scopus.com/) and IEEE xplore (http://ieeexplore.ieee.org). Based on the research goal, a search expression (or search string) was constructed. First, the keywords related to the research topic were identified, as well as their alternative terms and synonyms. The search expression, which can be seen in Table 1, was adapted according to the search mechanism of each digital library, as to not alter its logical sense. The searches were performed in the abstract, title and keywords fields.

Table 1. Search expression

Keyword	Alternative terms and synonyms
Visual impairment	("blind" OR "visually impaired" OR "visual disability" OR "blindness" OR "student disability" OR "unsighted pupils" OR "visual impairments") **AND**
Programming	("programming" OR "program") **AND**
Robotic	("robot" OR "robotic" OR "robotics")

The articles found by this expression were included for full reading if the title, keyword or abstract met the inclusion criteria. The following inclusion and exclusion criteria were defined:

Inclusion Criteria

I1- The result must be written in the English language.
I2- The result must be fully available.
I3- The result must contain in its title, keywords or abstract some relation to this work's topic (people with visual disabilities, programming, robotics).

Exclusion Criteria

E1- In the case of similar or duplicated results, only the most recent will be considered.
E2- Results which explore topics on programming teaching in a broad manner, and not specifically towards people with visual disabilities.

E3- Results which explore topics on the use of robotics in a broad manner, and not specifically towards people with visual disabilities.
E4- The result is not in the areas of Computer Science or Engineering.

The papers with the "Accepted" status were fully read, as to verify if/how do they answered the research questions.

3 Results

The searches in the digital libraries yielded 125 articles. After application of the selection criteria, 9 articles remained for full reading. Table 2 informs the amount of articles found, duplicated and accepted in each digital library. Articles from the years 2008 to 2015 were identified, and Table 3 shows us the articles, the years and conferences in which they were published.

Table 2. Articles found in the search classified by digital library

Library	Amount of articles found	Amount of articles duplicated	Amount of articles selected
Scopus	66	3	8
ACM	19	4	0
IEEE xplore	7	5	1
ScienceDirect	33	0	0
Total	125	12	9

Table 3. Articles found in the search ordered by year

Reference	Year	Conference
[11]	2008	SIGCSE 2008 - Proceedings of the 39th ACM Technical Symposium on Computer Science Education
[12]	2010	ASSETS 2010 - Proceedings of the 12th International ACM SIGACCESS Conference on Computers and Accessibility
[13]	2011	ACM Transactions on Computing Education
[14]	2012	IEEE Transactions on Learning Technologies
[15]	2013	Proceedings - 2013 Conference on Technologies and Applications of Artificial Intelligence, TAAI 2013
[16]	2013	ASEE Annual Conference and Exposition, Conference Proceedings
[17]	2014	IEEE SSCI 2014: 2014 IEEE Symposium Series on Computational Intelligence - RiiSS 2014: 2014 IEEE Symposium on Robotic Intelligence in Informationally Structured Space, Proceedings
[18]	2014	ASSETS 2014 - Proceedings of the 16th International ACM SIGACCESS Conference on Computers and Accessibility
[19]	2015	IEEE International Workshop on Robot and Human Interactive Communication

The research questions are answered below.

1- Which methodological procedures are being used in the teaching of programming with robots for people with visual disabilities?
a. Which are the methodologies for teaching robot programming to people with visual disabilities?

Of the selected studies, 6 (six) presented the use of robotics workshops [11, 13, 14, 16–18] as the teaching methodology. Although articles [12, 19] present robotics and programming environments for people with visual impairment, the results include only activities with people who can see. In the study of Kakehashi et al. [15], sighted users were blindfolded. The relation between the amount of students, educational level and robotics programming knowledge [11, 13, 14, 16–18], which had participation from people with visual disabilities, are presented in Table 4. The workshop's duration varied between works, as well as the resources and amount of participants, as shown in Table 5.

Table 4. Students' profile

Reference	Educational level	Amount of users	Programming knowledge
[11]	Not cited	14	Not cited
[13]	Not cited	14	Not cited
[14]	Middle school	9	Not cited
[16]	Middle and high school	32	Some had programming experience
[17]	Junior and high school	7	No one had programming experience
[18]	Not cited	10	3 (three) did not have experience and 7 (seven) had some experience

Table 5. Number of workshops vs. duration vs. organization

Reference	Number of workshops	Duration	Organization
[11]	1	4 days	Teams
[13]	3	-	Teams (3 students)
[14]	-	2 weeks	Individual
[16]	-	4 h	Teams
[17]	1	90 min	Individual
[18]	-	4 days (with duration of 3 to 4 h per day)	Teams (2–3 students)

In the study of Kakehashi et al. [15], which was performed with blindfolded sighted people, two experiments were conducted, one to investigate the efficiency of the tactile information provided on the programming blocks, made of Japanese cedar and equipped with RFID readers, and another to verify if the P-CUBE was easier to program than using text programming with the Japanese Beauto Builder2[1] PC software. The first experiment was conducted with 16 participants between 20 and 38 years of age, which were blindfolded. The time necessary for each user to identify the concrete blocks and the accuracy were recorded. In the second experiment, there were 10 sighted users, and the work does not mention if they had prior programming knowledge or if they were blindfolded. The time necessary to complete the task in each programming environment was recorded. These experiments were not performed with people with visual disabilities, but show that the blocks are easily recognizable, just as the P-CUBE proved itself to be a resource that helps to understand how to program.

Some works [11, 13, 14, 16] cite the use of tutorials to present new programming concepts and language syntax, used in the respective studies. In general, the tutorials also contained the description of the activities to be performed. The tasks were designed in a way that each task was based on the knowledge obtained from the previous tasks, systematically increasing in difficulty and, as consequence, the necessary skills to their conclusion [14]. Also, during the tasks, the instructor would ask questions that required for the student to give a verbal answer [14]. According to Demo [20], the verbalization technique allows for an instructor to verify what and how the student is planning to solving a problem, and thus to verify the technical foundation occurring through a formulation which is particular to each student. In the work of Kakehashi et al. [17], the use of tutorials is not explicitly cited, but they describe an experimental programming class in which the participants with visual disability had to solve certain exercises, which included a solution with a sequential program, and then with a program with condition and repetition structures.

There are works that use scale models in different forms and with various materials. In the study of Kakehashi et al. [17], the participants with visual impairment controlled a robot through a course laid out in an E.V.A (Ethylene-vinyl Acetate) material, with a black line indicating the way the robot should follow (Fig. 1a). In the study of Ludi and Reichlmayr [11], a labyrinth was constructed for the robot to navigate (Fig. 1b), set on a table so that the participants could feel the path and the robot with touch, and understand the action performed by the robot through touch. In Park and Howard [16], the environment was modified according to the activity (Fig. 1c).

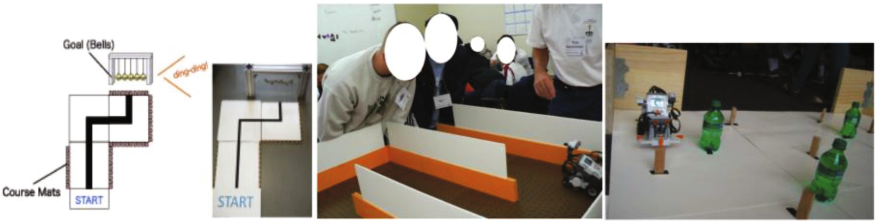

Fig. 1. (a) Trajectory with E.V.A (Source: [17]) (b) Labyrinth (Source: [11]) (c) Challenge (Source: [16])

b. Which are the characteristics of the programming environments used by people with visual disabilities?

Of the 9 (nine) works analyzed, 6 (six) used the LEGO mindstorm NXT [11–14, 16, 18] robotics kit. The others [15, 17, 19] created their own robots using the Arduino board. Figure 2a illustrates the robot used in the study of Ludi and Reichlmayr [13] and Fig. 2b depicts the robot used by Motoyoshi et al. [19].

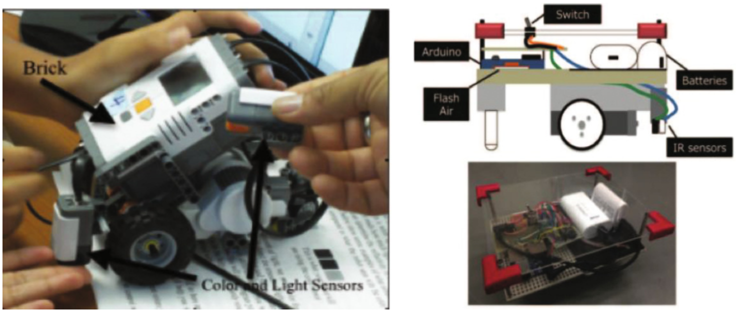

Fig. 2. (a) LEGO mindstorm NXT (Source: [13]) (b) robot with Arduino (Source: [19])

The studies of [11, 13] use the LEGO mindstorm NXT robotics kit composed by three motors (each mounted on a structure with a reduction box and a gyroscope/speed sensor) and touch, light, ultrasonic and sound sensors. The programming environment was developed by the open-source BricxCC[1]. The language used was the NXC[2], similar to the C language. According to the authors, this language was chosen for its syntax's simplicity, ease of learning and using and use with the screen reading software Jaws. The Windows operating system was used because the students were more familiar with it.

The studies of [12, 18] used the LEGO mindstorm NXT kit, switching from the BricxCC programming environment to the JBrick environment. JBrick's menu was simplified to 5 options, to minimize what is spoken by the screen reader, and in the article there is no mention of which options were left; BricxCC had 8 options. The language's commands were not cited.

The studies of [14, 16] used the LEGO mindstorm NXT kit composed by two motors with wheels and internal encoders for odometry calculations, two touch sensors to detect user input and collision events, a light sensor to detect landmarks on the floor, and an ultrasonic sensor to detect objects in front of the robot. The authors chose the BricxCC as the development environment. The commands for robot movement can be seen in Table 6. A Wii remote control (Wiimote) was used for haptic feedback. In the work of Howard et al. [14], the authors included a summarized feedback which was embedded in an intelligent agent called Robbie, which provides audio feedback to the student after his program is executed. Robbie was written in the C++ language.

The robot created by [15, 17, 19] is composed of an Arduino UNO microcontroller board, a microSD wireless card, a buzzer, two motors, a speed box and batteries.

Table 6. BricxCC's basic programming commands

Commands	Description
start_robot()	Switches the robot ON
move_up(X)	Moves the NXT robot forward an X number of times
turnleft(X)	Turns the NXT robot to the left an X number of times
turnright(X)	Turns the NXT robot to the right an X number of times
stop_robot()	Switches the robot OFF

The P-CUBE programming blocks were used (Fig. 3). The blocks used in this study had the following functions: Movement (forward, right, left, backwards), Timer (1st, 2nd, 3rd, 4th), IF (IF START IRsensor (L) END, IF START IRsensor (R) END) and LOOP (mobile robot repeats movements). The execution is performed as follows: information on the block's type is obtained through the block's RFID tag and transmitted to the computer. Then, this information is transmitted to the mobile robot using a microSD card. Thus, the user must manually connects the microSD card into robot to execute its functions. In the study of Kakehashi et al. [17], after testing with visually impaired participants, the robot was remodeled so that the RFID information was sent directly to the robot, eliminating the need for an microSD card.

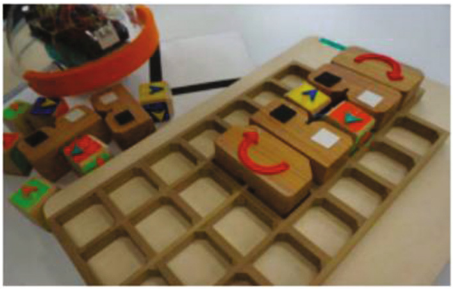

Fig. 3. Programming blocks made of Japanese cedar (Source: [19])

c. Which are the examples of good practices for teaching robot programming to people with visual disabilities?

Thirty-four recommendations for good practices in the teaching of robot programming for visually impaired people were identified. These recommendations were grouped in the following categories, described as: workshop preparation (13), content and activities (12), work dynamics (4) and data acquisition and instruments (5).

Workshop Preparation (13)

- To provide a training section to people who will help in the workshop, so that they know the necessary strategies to work with people with visual impairment. [11]
- Ask previously how each participant would like to receive the tutorial. For example, the material in Braille or with enlarged fonts. [11, 13]

- Orientate the students in regards to the room, the objects' layout, type and location of the equipment. [11]
- Keep ample space for circulation, for example, between tables, chairs, etc., so that the visually impaired participants can move with more safety and autonomy. [11]
- Make screen reader and screen magnifier software available to the participants, which may be configured by them. [11]
- Control the noise level in the place where the activities are conducted. Use an ample room or several smaller rooms. [11]
- Orientate the participants with visual impairment to use headphones. [13]
- Use Braille tags to identify the robot's principal components. [11]
- Orientate the participants in regards to the robot's parts so they can become familiar with the robot, as well as with the download of programs, etc. [13]
- To have monitors (Computing or Engineering students, family members, etc.) to follow the participants' activities, so that they can encourage the active participation of everyone on their work groups. [11, 13]
- To place the labyrinth, or another area in which the robot will operate, in a height where the participants may interact. [13]
- To have tactile information so that the students may identify and recognize the scenario and robot path, for example, using E.V.A materials and tactile models. [17]
- To make available programming environments which can be seen/heard by the participants, such as the interfaces and source code. [13, 16]

Content and Activities (12)

- To provide tutorials with commented code examples. [13, 14]
- Create quick reference sheets for symbols or commands. This is especially useful for students with visual disabilities which have to learn many new commands and syntaxes. Take into consideration that the participants may write down their own notes, for example, using a text editor on the computer. [13]
- To provide a set of simple commands (library), which may be built in stages so that the students may learn more complex coding more easily. [16]
- Put comments on code, watching out for its length so that it does not impair reading when scrolling down the screen. [13]
- Break long lines as to not impair people with low vision which have to increase font size. [13]
- Make activities which can be performed autonomously by the participants, regardless of being blind or having low vision. [13]
- Suit the activities to age and fun. [13] It is possible to perform activities with games, music, geometric shapes drawing, kick-the-can, in which the user programs the robot to kick objects along the way. [16]
- Design activities in a way that participants can learn from their own mistakes and may rethink their solution strategies. [18]
- Design the tutorials and project challenges to facilitate a progression in the student's programming skills. [13]
- Present information through various ways: orally, written on the board for people with low vision and in booklet form, which may be printed in Braille and/or

downloaded into a computer for blind people. For students with low vision, a booklet may be provided with enlarged font, printed or in the computer. [13]
- To provide information through touch and audio. [15, 19]
- To provide multimodal feedback so that the students may easily test their robot and correct/update their codes. [16]

Work Dynamics (4)

- Ask the participants to alternate in programming the robot. The same goes for initiating and stopping the program on the robot itself. [13]
- Keep the participants active in the performance of the activities. In group tasks, allow everyone to have a part in the problem's solution. This diminishes the impact of a dominant personality and it avoids students from been isolated. [13]
- Allow that the participants are capable of interacting with the programming environment, turning the robot on and off, activate motors and sensors as well as executing the desired program. [13]
- Encourage the participants to relate with their own world the skills and challenges of the robot. For example, they may relate the sensors with their own paradigms of navigation. [13]

Data Acquisition and Instruments (5)

- Apply pre and post questionnaires. [11, 13, 14, 16]
- Collect feedback of the participants' family members. [11, 13]
- Perform semi structured interviews after the activities. [18]
- Collect comments from the participants on the workshop experience. [17]
- Use the number of attempts to complete each task as an evaluation measure. [14]

d. Which were the difficulties/limitations in the use of robotics as support to the teaching of programming to people with visual disabilities?

The limitations faced in the works [15, 19] were related to the process of transferring the program, as it was necessary to transfer the program to a microSD card and to connect it to the robot so it could execute the commands. During the activities, a sighted person would help in this task. The students commented that they would like to perform the robot's operations autonomously [17]. Another problem was for the people with visual disabilities to distinguish the start of blocks (IF, LOOP) and the end of blocks (IF, LOOP). The blocks' differences were not sufficiently clear to the users [17].

In the work [11, 13], the BricxCC software was not totally compatible with the screen reader software used, the JAWS. Thus, is was necessary to have the help of a sighted person to perform some activities, such as helping to find a certain line of code according to the compiler's error.

In the study of Howard et al. [14], there was incomprehension from the users regarding the robot`s movement, more specifically to the feedback signals that distinguish between left and right turns.

Some difficulties were also reported by Ludi et al. [18]. There were problems regarding navigation in the code and, as future work, they proposed to perform a study with the use of different audios, such as pitch, earcons, to aid in programming. It was observed that the blind participants which sped up reading of the text (with the screen reader) at times would get lost in the code in terms of construction of IF/THEN and repetition blocks, especially when trying to correct mistakes. Another difficulty was that some participants were not familiar with the use of punctuation, such as brackets and braces, including their location on the keyboard. As for most participants, the possibility of working with a robot was something entirely new, and this may have influenced the study's results.

4 Discussion on the Review's Recommendations with Participants with Visual Disability

The recommendations for good practices were discussed with 2 participants, which have visual disabilities, one with low vision (P1) and another with congenital blindness (P2). P1 has 44 years and P2 has 23 years, both male. P1 and P2 were participating in a research project related to educational robotics, which has as objectives the development of a robot and of a robot programming language. P1 has intermediate knowledge on programming with C, C++, Python, Java, Pascal, Delphi and Logo, and P2 has basic knowledge in programming, having started with an experimental programming language based on Logo.

The discussions occurred individually with each participant. The different recommendation categories were explained and then the 29 recommendations were read and explained. The evaluator used an instrument to register the participants' feedback and to document disagreements and suggestions. It was chosen not to include recommendations related to the category of “Data acquisition and instruments”.

As results, of the 29 presented recommendations, P1 and P2 disagreed on the recommendation to “Control the noise level in the place where the activities are conducted. Use an ample room or several smaller rooms. [11]”, from the “Workshop preparation” category. The participants explained that the room's size may be irrelevant, if the students are using headphones. P1 also questioned the recommendation of “Use Braille tags to identify the robot's principal components. [11]”, also from the “Workshop preparation” category, as he considers that it depends on the robot's size. P1 points out that the robot must be sufficiently large to be able to contain Braille information, when this is the chosen approach. Still, in the case of small robots, if it is important to identify the robot's parts through Braille tags, he suggests that a larger model of the robot be built.

5 Conclusion

In this work, we presented the protocol and results of the Systematic Review on the teaching of programming with the support of robotics for people with visual disabilities, which were published in events relevant to the Computer Science area. The search yielded 125 papers, in which 9 were read in full. The results aimed to answer which methodological procedures are being used in the teaching of programming with robots for people with visual disabilities. There was emphasis on the use of workshops and tutorials, the use of robotics kits such as the LEGO mindstorm NXT, along with the BricxCC programming environment. Good teaching practices were identified, which were categorized in: workshop preparation, contents and activities, work dynamics and data acquisition and instruments.

From a total of 34 recommendations, 29 were discussed with visually impaired participants, which agreed with most of them and proposed suggestions to 2 recommendations.

Acknowledgments. This work was supported by the CNPq/MCTIC/SECIS N° 20/2016, National Council for Scientific and Technological Development – CNPq. JDO is supported by CAPES/PROSUP scholarships.

References

1. Solomon, C.J., Papert, S.: A case study of a young child doing turtle graphics in LOGO. In: Proceedings of the National Computer Conference and Exposition, 7–10 June 1976, pp. 1049–1056. ACM, New York (1976)
2. Al-Ratta, N.M., Al-Khalifa, H.S.: Teaching programming for blinds: a review. In: Fourth International Conference on Information and Communication Technology and Accessibility (ICTA), pp. 1–5 (2013)
3. Benitti, F.B.V., Vahldick, A., Urban, D.L., Krueger, M.L., Halma, A.: Experimentação com Robótica Educativa no Ensino Médio: ambiente, atividades e resultados. In: An. Workshop Informática Na Esc. vol. 1, pp. 1811–1820 (2009)
4. Chetty, J.: LEGO© mindstorms: merely a toy or a powerful pedagogical tool for learning computer programming? In: 38th Australasian Computer Science Conference, vol. 159, pp. 111–118 (2015)
5. Norton, S.J., McRobbie, C.J., Ginns, I.S.: Problem solving in a middle school robotics design classroom. Res. Sci. Educ. **37**, 261–277 (2007)
6. Long, R.G., Giudice, N.A.: Establishing and maintaining orientation for orientation and mobility. In: Foundations of Orientation and Mobility, pp. 45–62. American Foundation for the Blind, New York (2010)
7. Lahav, O., Schloerb, D.W., Kumar, S., Srinivasan, M.A.: BlindAid: a learning environment for enabling people who are blind to explore and navigate through unknown real spaces. In: 2008 Virtual Rehabilitation, pp. 193–197 (2008)
8. Lahav, O., Mioduser, D.: Multisensory virtual environment for supporting blind persons' acquisition of spatial cognitive mapping, orientation and mobility skills. In: Proceedings of Third International Conference on Disability, Virtual Reality and Associated Technologies, ICDVRAT, pp. 53–58 (2000)

9. Yu, W., Kuber, R., Murphy, E., Strain, P., McAllister, G.: A novel multimodal interface for improving visually impaired people's web accessibility. Virtual Reality **9**, 133–148 (2005)
10. Brereton, P., Kitchenham, B.A., Budgen, D., Turner, M., Khalil, M.: Lessons from applying the systematic literature review process within the software engineering domain. J. Syst. Softw. **80**, 571–583 (2007)
11. Ludi, S., Reichlmayr, T.: Developing inclusive outreach activities for students with visual impairments. Presented at the SIGCSE 2008 - proceedings of the 39th ACM technical symposium on computer science education (2008)
12. Ludi, S., Abadi, M., Fujiki, Y., Herzberg, S., Sankaran, P.: JBrick: accessible LEGO mindstorm programming tool for users who are visually impaired. Presented at the ASSETS 2010 - proceedings of the 12th international ACM SIGACCESS conference on computers and accessibility (2010)
13. Ludi, S., Reichlmayr, T.: The use of robotics to promote computing to pre-college students with visual impairments. ACM Trans. Comput. Educ. **11**, 1–20 (2011)
14. Howard, A.M., Park, C.H., Remy, S.: Using haptic and auditory interaction tools to engage students with visual impairments in robot programming activities. IEEE Trans. Learn. Technol. **5**, 87–95 (2012)
15. Kakehashi, S., Motoyoshi, T., Koyanagi, K., Ohshima, T., Kawakami, H.: P-CUBE: block type programming tool for visual impairments. Presented at the proceedings of the conference on technologies and applications of artificial intelligence, TAAI (2013)
16. Park, C.H., Howard, A.M.: Engaging students with visual impairments in engineering and computer science through robotic game programming (research-to-practice). Presented at the ASEE annual conference and exposition, conference proceedings (2013)
17. Kakehashi, S., Motoyoshi, T., Koyanagi, K., Oshima, T., Masuta, H., Kawakami, H.: Improvement of P-CUBE: Algorithm education tool for visually impaired persons. Presented at the Robotic Intelligence In Informationally Structured Space (RiiSS), 2014 IEEE symposium on robotic intelligence in informationally structured space (2014)
18. Ludi, S., Ellis, L., Jordan, S.: An accessible robotics programming environment for visually impaired users. Presented at the ASSETS14 - proceedings of the 16th international ACM SIGACCESS conference on computers and accessibility (2014)
19. Motoyoshi, T., Kakehashi, S., Masuta, H., Koyanagi, K., Oshima, T., Kawakami, H.: The usefulness of P-CUBE as a programming education tool for programming beginners. Presented at the proceedings of the IEEE international workshop on robot and human interactive communication (2015)
20. Demo, P.: Educar pela pesquisa. Autores Associados LTDA, Campinas, SP (2011)