GoDonnie: A Robot Programming Language to Improve Orientation and Mobility Skills in People Who are Visually Impaired

Juliana Damasio Oliveira
Pontifical Catholic University
of Rio Grande do Sul
Porto Alegre, Brazil
juliana.damasio@acad.pucrs.br

Márcia de Borba Campos
Inedi College - CESUCA
Porto Alegre, Brazil
marciabcampos@hotmail.com

Alexandre Amory
Pontifical Catholic University
of Rio Grande do Sul
Porto Alegre, Brazil
alexandre.amory@pucrs.br

Rafael H. Bordini
Pontifical Catholic University
of Rio Grande do Sul
Porto Alegre, Brazil
rafael.bordini@pucrs.br

ABSTRACT
This work presents GoDonnie a programming language to command a robot to improve orientation and mobility (O&M) skills in people who are visually impaired (PVI). The GoDonnie programming language is based on the Logo language. GoDonnie runs in a programming environment called Donnie. This environment has a 2D graphic simulator with a virtual robot, in which one can visualise and receive sound feedback from the execution of the language commands for moving the virtual robot in the environment. GoDonnie has been evaluated with PVI to verify its usability and support to O&M. The results indicate that GoDonnie has good usability, supports the development of O&M in PVI and meets the expectations regarding the programming environment. A video of GoDonnie execution is available in https://youtu.be/HE__sAgfNBo

Author Keywords
Programming language; orientation and mobility; visual impairment; usability.

CCS Concepts
•**Human-centered computing** → **User studies; User interface programming;** *Accessibility systems and tools;* •**Social and professional topics** → **People with disabilities;**

INTRODUCTION
The skills of orientation and mobility (O&M) are indispensable for people who are visually impaired (either blind or low vision) because such skills increase their autonomy, their independence, and inclusion in school and society [11, 3]. Previous papers show that robotics programming can be a useful tool for the development of computational thinking for solving challenges in the field of motor skills [1, 2, 5]. Several robotic environments are being proposed to facilitate the teaching of computer programming such as Lego mindstorms [7] and NaoBlocks [12]. However, these initiatives do not address all students, especially those who have visual impairment [10]. This is because many programming environments are based on graphical interfaces, which makes them inaccessible to this group of users [8, 4, 6, 10].

Furthermore, it has been reported that some languages are difficult to use by programmers who are visually impaired [6, 9]. Among the programming languages that present such difficulties, we mention Python, which uses white spaces to delimit code blocks, which is hard to navigate with screen readers [6, 9]. Consider also languages such as C and Java, which use braces to delimit code blocks. In this case, the obstacle is when there is a block with sub-blocks because students who have visual impairment tend to have difficulty in figuring out the matching braces.

We created a programming language called GoDonnie, based on the Logo language. GoDonnie was created to be used as a tool to aid in the resolution of spatial problems involving O&M. It was conceived in accordance with criteria of accessibility and usability to be used by PVI. The assumption is that using GoDonnie and the virtual environment, the PVI can improve their analysis, programming, and O&M skills. In addition, the playful aspect of robotics may encourage, especially young students, to use the proposed environment and acquire more programming skills and O&M from their use. We evaluated the GoDonnie language with PVI.

ACM ISBN 978-1-4503-6676-2/19/10.
https://doi.org/10.1145/3308561.3354599

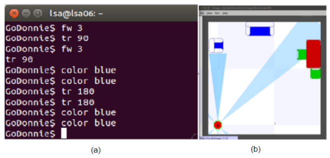

Figure 1. Donnie Programming Environment (a) editor, (b) 2D simulator

GODONNIE

We have developed GoDonnie language[1] that students use to simulate a robot's behavior in a virtual environment; the behavior is described to the user through audible messages. GoDonnie has commands to move the robot forward and backward (FW, BW) and to rotate it to the right and to the left (TR, TL). Also, there are selection (IF), repeat (REPEAT, WHILE, FOR), procedure, and assignment (VAR) commands that are common to general-purpose programming languages. However, it lacked a better integration with screen readers, and commands that avoided the use of simultaneous keys, as well as commands that would allow the user to better understand the scenario, how the robot relates to that environment and moves in it. From these limitations, specific commands were defined for the end-users such as SCAN, COLOR, DISTANCE, POSITION, and STATE. Some commands combine feedbacks of spoken messages and iconic sounds. In addition, the Orca screen reader reproduces what the user types in the terminal.

- COLOR c: returns the number of objects of a given color *c* (blue, red, green).
- SCAN: scans objects within 180^o in front of the robot, then it returns the color, the distance, and the angle to the detected objects.
- DISTANCE d: the robot has six range sensors to measure the distance from obstacles; this command returns the distance of the sensor *d* (front; frontal right; frontal right; back; back left; back right) to an object in steps (one robot step equals 5 cm).
- STATE: returns a series of current information about the robot such as its current pose and the last instruction.
- POS or POS k: returns all coordinates (X,Y, and angle A) or only the indicated *k* (*X*, *Y*, or *A*).

The Donnie Programming Environment (DPE) includes an interpreter (Client), an editor where GoDonnie is typed, and an 2D robotics simulator (Fig. 1). The 2D simulator, called Stage [13], is configurable, we can add new objects, and change the arrangement of objects.

EVALUTATION

This evaluation was attended by one participant who is blind (P1) and one who is low-vision (P2). P1 is a non-programmer. P2 has prior experience in programming. We designed 6 programming activities, 1 activity to evaluate the GoDonnie guide, and a questionnaire for this experiment. The questionnaire contained questions related to the following criteria: ease of use; utility; error prevention; sound interface; O&M; programming; help and documentation, and satisfaction. In total, we had 48 closed questions. Evaluations occurred individually with each user.

[1]The language and system are available at https://donnie-user-manual.readthedocs.io/en/latest/docs/godonnie/index.html.The language commands are in Portuguese and English.

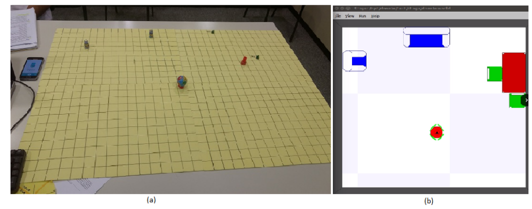

Figure 2. (a) Tactile map mounted by P1, (b) virtual environment

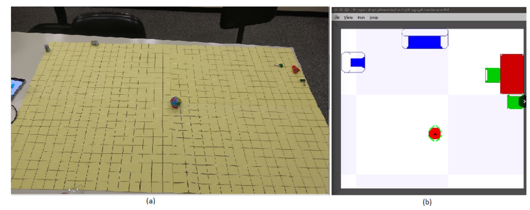

Figure 3. (a) Tactile map mounted by P2, (b) virtual environment

Results

P1 and P2 performed all tasks successfully. In the first programming activity the users had to explore the virtual environment with GoDonnie and then representing in the tactile map that environment. Users have used the SCAN and COLOR commands. Figure 2 depicts the physical environment constructed by participant P1 as well as the virtual environment. Figure 3 depicts the tactile map that the P2 set up and the virtual environment. We verify that the user was able to mount a map that is similar to the virtual environment. In this sense, it demonstrates that he has been able to make a mental map of the environment according to the sound feedback provided by GoDonnie. The participants strongly agreed that GoDonnie is ease of use, utility, prevents mistakes from occurring, help and documentation were adequate. The users suggest that the GoDonnie guide should contain more examples of using the commands (P1), and the COLOR command should be changed to inform the location (P1) and size of the objects (P2).

CONCLUSION

GoDonnie's evaluation in the DPE was done with two users. As a result, GoDonnie's commands were found to have good usability with ease of use, utility, error prevention and handling, and users were satisfied with the usage. In addition, it has a nice sound interface. It has also been found that the use of GoDonnie assists in the development of O&M, because participants were able to make a mental map of the environment and related objects. This map was externalised with the use of a tactile map, in which the participants distributed the objects representing the virtual scenario.

ACKNOWLEDGMENTS

This study was financed by the CAPES, Finance Code 001.

REFERENCES

[1] Samuel Azevedo, Akynara Aglaé, and Renata Pitta. 2010. Minicurso: Introdução a Robótica Educacional. In *62o reunião anual da Sociedade Brasileira Para o Progresso da Ciência (SBPC)*.

[2] Rayshun Dorsey, Chung Hyuk Park, and Ayanna M. Howard. 2013. Developing the Capabilities of Blind and Visually Impaired Youth to Build and Program Robots. In *Proceedings of the CSUN Annual International Technology and Persons with Disabilities Conference*. 55–67.

[3] Claudia Pereira Dutra. 2003. *Orientação e Mobilidade: Conhecimentos básicos para a inclusão do deficiente visual*. MEC/SEESP. 165 pages.

[4] A. M. Howard, Chung Hyuk Park, and S. Remy. 2012. Using Haptic and Auditory Interaction Tools to Engage Students with Visual Impairments in Robot Programming Activities. *IEEE Transactions on Learning Technologies* 5, 1 (Dec. 2012), 87–95.

[5] L'udmila Jašková and Mária Kaliaková. 2014. Programming microworlds for visually impaired pupils. In *Proceedings of the 3rd internatiobal constructionism conference*. Oesterreichische Computer Gesellschaft.

[6] Shaun K Kane and Jeffrey P Bigham. 2014. Tracking stemxcomet: teaching programming to blind students via 3D printing, crisis management, and twitter. In *ACM technical symposium on Computer science education*. 247–252.

[7] Frank Klassner and Scott D Anderson. 2003. Lego MindStorms: Not just for K-12 anymore. *IEEE Robotics & Automation Magazine* 10, 2 (2003), 12–18.

[8] Stephanie Ludi and Tom Reichlmayr. 2011. The Use of Robotics to Promote Computing to Pre-College Students with Visual Impairments. *Trans. Comput. Educ.* 11, 3, Article 20 (Oct. 2011), 20 pages. DOI: http://dx.doi.org/10.1145/2037276.2037284

[9] Sean Mealin and Emerson Murphy-Hill. 2012. An exploratory study of blind software developers. In *IEEE Symposium on Visual Languages and Human-Centric Computing*. IEEE, 71–74.

[10] R. Pitta Barros, A. Medeiros Filgueira Burlamaqui, S. Oliveira de Azevedo, S. Thomaz de Lima Sa, L. Marcos Garcia Goncalves, and A. Aglae R S. da Silva Burlamaqui. 2017. CardBot - Assistive Technology for Visually Impaired in Educational Robotics: Experiments and Results. *IEEE Latin America Transactions* 15, 3 (March 2017), 517–527. DOI: http://dx.doi.org/10.1109/TLA.2017.7867603

[11] Juliet Stone. 1995. *Mobility for special needs*. Burns & Oates.

[12] Craig J Sutherland and Bruce A MacDonald. 2018. NaoBlocks: A Case Study of Developing a Children's Robot Programming Environment. In *2018 15th International Conference on Ubiquitous Robots (UR)*. IEEE, 431–436.

[13] Richard Vaughan. 2008. Massively multi-robot simulation in stage. *Swarm Intelligence* 2, 2 (Dec 2008), 189–208.