# Demo: Expanding Blocks4All with Variables and Functions

**Jacqueline Shao Yi Ong, Nana Adwoa O. Amoah, Alison E. Garrett-Engele, Mariella Irene Page, Katherine R. McCarthy, Lauren R. Milne**
Macalester College, Department of Mathematics, Statistics and Computer Science
Saint Paul, MN, USA
{jong, namoah, agarrett, mpage, kmccart3, lmilne}@macalester.edu

## ABSTRACT
Blocks-based programming environments are often inaccessible for children with visual impairments who cannot interact with their visual components. We present our work on expanding Blocks4All, a touchscreen-based BBPE that is accessible with screen readers, to improve its accessibility and expand its functionality. We describe our designs to support more complex functionality such as modifiable blocks, variables and functions and provide our code to encourage others to make their own environments accessible. We plan to evaluate the updated interface with children with low vision participating in a robotics competition.

### Author Keywords
Accessibility; blocks-based programming; visual impairments; motor impairments; educational technology; touchscreen interactions.

### ACM Classification Keywords
Human-centered computing: Accessibility technologies.

## INTRODUCTION
Blocks-based programming environments (BBPEs) are a popular way to introduce computer science to children, because they focus on teaching programming concepts over syntax details [7]. These environments rely heavily on visual elements. This makes them inaccessible for children with low or no vision, as they are generally incompatible with assistive technologies such as screen readers. As a result, these children are unable to use BBPEs and face an early barrier to their computer science education.

In response, Milne et al. created Blocks4All, a BBPE that is accessible for children with visual impairments [6]. Blocks4All was developed as an iOS iPad application to use the functionality of Apple's VoiceOver [10], Dynamic Type [11] and Switch Control [12]. It is used to control the Wonder Workshop robots [13] Dash and Dot.



In this paper, we describe how we have extended Blocks4All since its original presentation. We sought input from a TVI to refine the prototype. Our contributions include: (1) introducing modifiable blocks, variables, functions, and more robot commands to the application, and (2) updating the user interface to better support users with color blindness and motor impairments. These changes are reflected in the newest version of Blocks4All, which is available for download [https://github.com/milnel2/blocks4alliOS]. The added functionalities mean that Blocks4All is ready for evaluation in a real-world setting, which we plan to carry out with a group of children in the Wonder League Robotics Competition (WLRC) [14].

## RELATED WORK
There is related work on the problem of making BBPEs accessible for people with visual impairments. A number of researchers have looked at representing a block-based program structure as a hierarchical tree structure, with options to add and delete blocks, that can be navigated using a screen reader and keyboard controls [1,3,5]. There has also been research on making tangible, accessible environments [2,8] and an audio-based BBPE for people with motor impairments [9].

This work builds on our previous research on Blocks4All [6,7], which focuses on making an accessible BBPE using a touchscreen. Blocks4All was designed for iPads, which have a built-in screen reader called VoiceOver [10]. If a user touches a "focusable" item on the screen when VoiceOver is enabled, VoiceOver will read aloud the label of the item. Users can also iterate through the focusable items on the screen by swiping left or right. To select or interact with the currently focused item, users can double tap anywhere on the screen. In this work, we build upon the original Blocks4All interface to include functions and variables.

## DESIGN ENHANCEMENTS
Blocks4All is designed to empower users by enabling them to independently explore and understand the interface. We envision that our enhancements would facilitate interactive experiences and engage users with disabilities.

Figure 1 shows the layout of Blocks4All. There is a vertically-scrolling *toolbox* on the left side of the screen that lists all the block categories. The center of the screen acts as

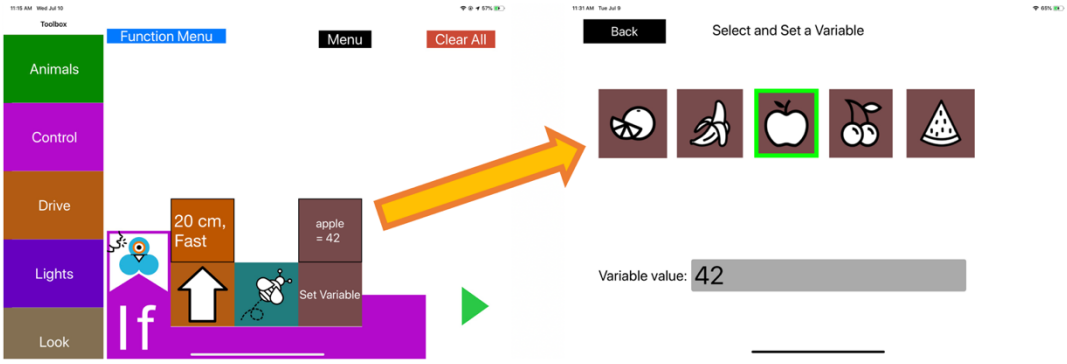


**Figure 1. Blocks4All Interface a) The left side of the screen is a toolbox where users can select blocks to place in the workspace on the bottom of the screen. The program in the workspace reads: "If Dash hears a noise, drive forward 20cm fast, make a buzzing noise and set the apple variable to 42." b) Tapping the modifier button above the "Set Variable" block brings users to the screen on the right, where they can set the icon and value of their variable. The Apple variable is currently set to 42.**

the *workspace* where users can build a stack of blocks that made up their program. This stack runs along the bottom of the screen to make it easier to locate without sight. The play button is located at the bottom right of the screen. To place blocks in the workspace, users first select the block they want to move and then select the place in the workspace where they would like to move it.

We describe the changes we have made to Blocks4All below.

### User Interface

We made changes to the interface based on the principle of universal design [4]; we wanted Blocks4All to be accessible to users with and without sight and users with motor impairments. We used a palette builder [15] and an application [16] to adjust Blocks4All's color schemes and contrast levels to ensure they were accessible for low-vision and red-green and blue-yellow color blindness.

We integrated Apple's Dynamic Type feature into Blocks4All so that the size of blocks and their overlaying text and images can match the user's preferred content size on their device settings. As the application responds to the user's preferences, it caters to all degrees of sight.

While the original Blocks4All was developed for users with visual impairments, we started running the application with Switch Control active to ensure it was also accessible for users with motor impairments.

### Modifiable blocks

Modifiable blocks allow users to more precisely manipulate robot commands. This feature promotes users' experimental learning by letting them test different modifier values for their desired outcome. We implemented modifiers with labelled buttons on top of the modifiable blocks. For example, in Figure 1, the "Drive Forward" block has a modifier button that can be selected to change the distance and speed with which a robot moves. It is currently set to "Drive Forward 20 cm Fast". These buttons lead to screens with custom slider and stepper objects to adjust the values; these support VoiceOver narration by snapping to interval markers so that users can swipe through to set the values.

### Variables

The addition of variables to Blocks4All teaches our users about storing information that allow them construct more flexible programs. In our implementation, we create a new Variables category in the blocks toolbox and list each robot action that can take a variable as an argument within it. Users can add these action blocks to the workspace and tap on their modifier buttons, which segues to a screen where they can select a variable to modify the action. Figure 1 shows the modifier screen to set the "apple" variable to 42 cm. When the user wishes to drive the robot forward by 42 cm, they can select the "apple" variable from the Drive Forward block's modifier button. We chose to keep the design of our variables similar to the Wonder Workshop application [17] to ensure that Blocks4All would fulfill the WLRC requirements.

### Functions

The ability to create and call functions is a key computing concept which we plan to add to Blocks4All. Currently, Blocks4All only supports one stack of blocks at the bottom of the screen to make the stack easier to find for someone who is blind. To allow users to create different functions (essentially different stacks of code), we added a button called "Function Menu" to the top of the workspace (Figure 1). We plan to make selecting that button lead to a screen that (1) lists the workspace and any previously created functions, and (2) allows the user to create a new function. Selecting any of these options take the user to a new workspace where they can build or edit a function. Created functions are then added as blocks in the toolbox menu and can be placed in code like any other block.

## CONCLUSION

Our enhancements to Blocks4All have improved its accessibility for a larger audience of users with low-vision and motor impairments and have incorporated functionalities that enable users to create complex programs while teaching basic computing concepts, such as variables and functions. We plan to evaluate our changes to Blocks4All when it is used by a group of children with low vision in the upcoming WLRC. In the future, we also plan to thoroughly evaluate its accessibility with VoiceOver and Switch Control.

## REFERENCES


1. Google Inc. Accessible Blockly. Retrieved from https://blockly-demo.appspot.com/static/demos/accessible/index.html
2. Varsha Koushik, Darren Guinness, and Shaun K. Kane. 2019. StoryBlocks: A Tangible Programming Game To Create Accessible Audio Stories. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (CHI '19), 492:1–492:12. https://doi.org/10.1145/3290605.3300722
3. Varsha Koushik and Clayton Lewis. 2016. An Accessible Blocks Language: Work in Progress. In *Proceedings of the 18th International ACM SIGACCESS Conference on Computers and Accessibility* (ASSETS '16), 317–318. https://doi.org/10.1145/2982142.2982150
4. Richard E. Ladner. 2015. Design for user empowerment. *interactions* 22, 2: 24–29. https://doi.org/10.1145/2723869
5. Stephanie Ludi and Mary Spencer. 2017. Design Considerations to Increase Block-based Language Accessibility for Blind Programmers Via Blockly. *Journal of Visual Languages and Sentient Systems* 3, 1: 119–124. https://doi.org/10.18293/VLSS2017-013
6. Lauren R. Milne, Catherine M. Baker, and Richard E. Ladner. 2017. Blocks4All Demonstration: A Blocks-Based Programming Environment for Blind Children. In *Proceedings of the 19th International ACM SIGACCESS Conference on Computers and Accessibility* (ASSETS '17), 313–314. https://doi.org/10.1145/3132525.3134774
7. Lauren R. Milne and Richard E. Ladner. 2018. Blocks4All: Overcoming Accessibility Barriers to Blocks Programming for Children with Visual Impairments. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (CHI '18), 69:1–69:10. https://doi.org/10.1145/3173574.3173643
8. Nicolas Villar, Cecily Morrison, Daniel Cletheroe, Tim Regan, Anja Thieme, and Greg Saul. 2019. Physical Programming for Blind and Low Vision Children at Scale. In *Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems* (CHI EA '19), INT003:1–INT003:4. https://doi.org/10.1145/3290607.3313241
9. Amber Wagner, Ramaraju Rudraraju, Srinivasa Datla, Avishek Banerjee, Mandar Sudame, and Jeff Gray. 2012. Programming by Voice: A Hands-free Approach for Motorically Challenged Children. In *CHI '12 Extended Abstracts on Human Factors in Computing Systems* (CHI EA '12), 2087–2092. https://doi.org/10.1145/2212776.2223757
10. Turn on and practice VoiceOver on iPad. *Apple Support*. Retrieved July 10, 2019 from https://support.apple.com/guide/ipad/turn-on-and-practice-voiceover-ipad9a246898/ios
11. Vision accessibility features in iOS. *Apple Support*. Retrieved July 10, 2019 from https://support.apple.com/en-us/HT210076
12. Use Switch Control to navigate your iPhone, iPad, or iPod touch. *Apple Support*. Retrieved July 10, 2019 from https://support.apple.com/en-us/HT201370
13. Wonder Workshop | Home of cue, Dash and Dot, robots that help kids learn to code. Retrieved June 27, 2019 from https://www.makewonder.com/
14. Wonder Workshop | Home of cue, Dash and Dot, robots that help kids learn to code. *Wonder Workshop - US*. Retrieved July 10, 2019 from https://www.makewonder.com/classroom/robotics-competition/
15. Accessible color palette builder. Retrieved July 2, 2019 from https://toolness.github.io/accessible-color-matrix/
16. Sim Daltonism. *Michel Fortin*. Retrieved July 2, 2019 from https://michelf.ca/projects/sim-daltonism
17. Wonder Workshop | Home of cue, Dash and Dot, robots that help kids learn to code. *Wonder Workshop - US*. Retrieved July 8, 2019 from https://www.makewonder.com/apps/blockly/