# BLOCKS4ALL: MAKING BLOCK PROGRAMMING LANGUAGES ACCESSIBLE FOR BLIND CHILDREN

Lauren R. Milne
Computer Science and Engineering
University of Washington
milnel2@cs.washington.edu

## Abstract

Block programming languages, such as Scratch and Blockly, are being used as to introduce children to programming, but because they rely heavily on visual aspects, blind children are being left behind their peers in access to computer science education. We propose finding new techniques to make these types of programs accessible to blind children. We plan to use an iterative design process to create a web-based application on a touchscreen laptop, where children can synthesize music using different instruments and recordings of themselves. We plan to work with students at a local school to test and refine initial prototypes in a workshop setting. We will then evaluate the final prototype in a longitudinal study with students: collecting the programs that they create over a two-week period, and conducting observations and interviews throughout that period in order to evaluate Blocks4All.

## Introduction

Block programming languages, such as Scratch [4], ScratchJR [1], and Blockly[3], are becoming increasingly popular learning tools for children in elementary, middle and high school, as they allow children to avoid learning syntax and focus instead on logic and common programming concepts. These languages are composed of blocks that represent programming control structures as well as data structures and values. Each type of block has a specific shape, so that when blocks are joined together in a script, only legal syntax is allowed. This allows children to learn about logic and problem solving without worrying about the syntax and details (e.g. brackets and semi-colons) involved in most programming languages. Children can quickly build working programs without having to perform a lot of debugging just to get a program to run. However, because of the visual puzzle-piece nature of the blocks, block languages today are largely inaccessible to blind children, which puts these children at a disadvantage in learning computer science as compared to their sighted peers.

We propose developing an accessible block language that can be used by blind children, specifically 5-7 year olds, so they can learn high-level programming concepts like their peers without focusing on the low-level syntax. As part of this solution, we will develop non-visual techniques to help children understand which different pieces of code (or blocks) fit together, paralleling the visual affordances of the "puzzle-piece" blocks. We will also develop techniques that allow children to easily move and place the blocks, and techniques that allow children to understand the runtime execution of their code. Additionally, we will develop blocks that allow the children to create rich non-visual programs that are fully accessible, as the current choices (traversing mazes, animating stories) are generally not rich experiences for blind children. Our anticipated contributions are:

[3] https://developers.google.com/blockly/

1. The open-source code of our accessible block language,
2. Design considerations for accessible programming languages, and
3. Empirical findings on how the children interact with the games.

Ideally we will use the principles of universal design to create a block programming language that can be used and enjoyed by children whether they are blind, have low vision or are sighted.

## Related Work

There are many block languages today that have their basis in visual programming languages. These include Scratch [4] and ScratchJR [1], both adaptations of the Logo language created at MIT, and the Blockly[1] library, which allows developers to quickly add block-based representations of code to their applications. Unfortunately, these languages are generally not accessible for blind children. The standard blocks in both Scratch and Blockly are not accessible to screen readers and the "drag and drop" paradigm is not feasible for children with either visual or motor impairments. There are exceptions to the general state of inaccessibility. Lewis [3] is in the early stages of creating an accessible language, Noodle, a nonvisual dataflow programming system, where functional units of code are strung together and the output of one piece of code is passed in as the input of another. There is also a version of Blockly that is accessible to screen readers, which allows users to access and place the blocks through hierarchical menu-based control, and navigate these using the principles of accessible web navigation.[4] Wagner et al. have also created a tool called Myna, which allows people with motor-impairments to use voice controls to program in Scratch [6].

There have also been tactile versions of block languages, including the commercial product Osmo[5], and the closely related Strawbies project [2], which both rely on tangible building blocks that can be pieced together and interpreted using an iPad to guide a monster on his quest to find strawberries, as well as the KIBO robotics kit, which allows young children to programmatically control a robot using wooden blocks [5]. While the tangible, tactile languages hold a lot of promise, they are not currently designed with blind children in mind and rely on a lot of visual aspects to convey what the blocks do.

Additionally, the output of both the tactile and digital block programming languages is generally not very accessible, as the programs created are visual: often animating a character to interact with others or traverse a maze with few audio cues. There are exceptions to this: Scratch includes blocks that allow for audio to be included (ScratchJR has much more limited support for this), and many of the languages support controlling robots, which is inherently accessible for blind children. We intend to build on this work and use the design principles these researchers have distilled to inform an accessible language: Blocks4All.

## Blocks4All Design

We plan to develop a prototype using Google's open source Blockly code.[6] We will make the blocks accessible and design for touch interactions. Specifically, we plan to incorporate the following design principles:

[4] https://blockly-demo.appspot.com/static/demos/accessible/index.html
[5] https://www.playosmo.com/en/
[6] https://github.com/llk/scratch-blocks

1. *Universal design:* designed to be usable by both sighted and blind children,

2. *Designed for young children:* as block-based languages have been most successful with young children in K-5,

3. *Touchscreen based:* using similar touchscreen interactions as used by the screen readers on iOS and Android,

4. *Allows for non-visual program exploration and construction:* Touchscreen elements can be explored using speech and sound output to give spatial information about program structure to blind children. The "drag and drop" action can be replaced with a more accessible "select, select, drop" action, which will also benefit children with limited mobility,

5. *Support for audio:* Programs that produce audio output can be enjoyable for all children.

We will develop a number of interactions to indicate how blocks can be put together using sound and allow users to make small customizations to blocks (e.g. change the number of times a loop will iterate). We will also design interactions so blind children can explore program structure and understand the state of the program during runtime. For the output, we plan on creating blocks that allow children to synthesize music by using different instruments, recordings of themselves and animal noises, as well as control structures such as loops and event-driven programming.

## Evaluation

We will develop the initial prototypes iteratively with children at a workshop at a local school for the blind, taking note of challenges the students encounter and suggestions from both the students and teachers.

We will refine the prototypes, and test the final design in a longitudinal study over two weeks with children at the school. In the study, we will meet with the children and watch them interact multiple times a week, as well as allow them to use the system unsupervised and collect saved versions of their programs. We will report on qualitative results: how much they liked the system, and what they found challenging, as well as more quantitative metrics based on the collected programs: on how complex their programs were, how this changed over the two-week period, and how much the children interacted with the system.

## Acknowledgments

Thanks Richard Ladner for his input and ideas in this space. This work supported by the National Science Foundation Graduate Research Fellowship under Grant No. DGE-1256082.

## References

1. Flannery, L.P., Silverman, B., Kazakoff, E.R., Bers, M. U., Bontá, P., and Resnick, M. 2013. Designing ScratchJr: Support for Early Childhood Learning Through Computer Programming. *Proc. of the 12th International Conference on Interaction Design and Children*, ACM, 1–10. http://doi.org/10.1145/2485760.2485785

2. Hu, F., Zekelman, A., Horn, M., and Judd, F. 2015. Strawbies: Explorations in Tangible Programming. *Proc. of the 14th International Conference on Interaction Design and Children*, ACM, 410–413. http://doi.org/10.1145/2771839.2771866

3. Lewis, C. 2014. Work in Progress Report: Nonvisual Visual Programming. *Psychology of Programming Interest Group*. Retrieved from http://users.sussex.ac.uk/~bend/ppig2014/14ppig2014_submission_5.pdf

4. Resnick, M., Maloney, J., Monroy-Hernández, A., et al. 2009. Scratch: Programming for All. *Commun. ACM* 52, 11: 60–67. http://doi.org/10.1145/1592761.1592779

5. Sullivan, A., Elkin, M., and Bers, M.U.. 2015. KIBO Robot Demo: Engaging Young Children in Programming and Engineering. *Proc. of the 14th International Conference on Interaction Design and Children*, ACM, 418–421. http://doi.org/10.1145/2771839.2771868

6. Wagner, A., Rudraraju, R., Datla, S., Banerjee, A., Sudame, M. and Gray, J. 2012. Programming by Voice: A Hands-free Approach for Motorically Challenged Children. *CHI '12 Extended Abstracts on Human Factors in Computing Systems*, ACM, 2087–2092. http://doi.org/10.1145/2212776.2223757

## About the Author:

Lauren Milne is a fifth year PhD student in Computer Science and Engineering at the University of Washington. She is interested in accessibility and providing equal access to information for everyone.