Accessible Blockly Plugin

Authors: Rachael Bosley, Luna Meier

# 1. Overview

## Goal:

Allow for people with any level of vision to work with Blockly, whether they have no vision, full vision, or somewhere in between. The program must also stay user friendly so that children can use it effectively. In order to allow Blockly to be used by blind users, the entire interface must be hotkey accessible and easily readable by a screen reader.

## Environment (What Blockly is, and what tools that already exist we have to work around):

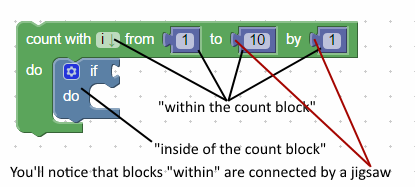
Blockly is a visual coding interface. It allows users to create code by placing blocks together. New blocks can be easily created by developers to enable more functionality within the interface. Blockly code can be exported to XML, Python, Javascript, PHP, and Dart. Basic Blockly does not have the ability to run code, so these exports are necessary to see the final product.

## Glossary:

**User**: Person who is interacting with the finished version of the program, after everything has been put together.

**Developer**: Person or group responsible for integrating our plugin into their own Blockly implementation.

**Within vs. Inside of**: A block or field is considered “within” another block if it is an integral part of the block itself. A block is considered “inside of” another block if it is part of a statement that is run by a block. This concept is rather confusing, so see the picture below for a visual example:



From a coding standpoint, child blocks that are within a parent block output directly into the parent block, while child blocks that are inside of a parent block are code that is run according to the parent block.

# 2. Design

## Features:

### Block Navigation:

The user will be able to navigate throughout the scene with a series of hotkeys. When the user presses a hotkey the selected block should change, and the user will be informed of what they are now selecting. The user should be able to move to any block attached to the current block. Very often in Blockly the user will need to get to a child block of a parent block that is within the parent block. In order to allow the user to select all possible child blocks of a selected parent block, we have implemented a specific mode that allows you to cycle through all possible connections of the selected block. You press a hotkey, which allows you to use the hotkeys you were previously using to navigate through the scene to be used to select a specific connection instead. In addition, this mode will allow you to select any fields that are part of the block, ensuring that the user has full access to the block. Once the user has interacted with the block as they like, the program returns to the normal navigation mode, where pressing the navigation hotkeys allows you to move to attached blocks and inside of blocks.

### Adding Blocks to the Scene:

### Tree Diagram:

### Etc…(Rachael add more of these):

## Philosophy (What are our main goals and how do we plan on achieving them):

## List of Hotkeys:

Alt Shift C: toggle collapse/expand  
Alt Shift D: duplicate  
Alt Shift E: toggle enable/disable  
Alt Shift H: navigate to help page for selected block  
Alt Shift I: toggle inline  
Ctrl Y: redo  
Ctrl Z: undo  
Delete: delete a block  
Enter: leaves editing a field and saves the field  
Escape: exit current field or menu  
A: traverse out  
C: add a comment to selected block  
D: traverse in  
G: jump to associated block or comment  
R: jump to top of section  
S: traverse down  
W: traverse up

# 3. Technical

## Libraries Used:

## Coding Standards:

# 4. Future Plans

## Testing on blind children:

## Building a module for teaching blind children:

# 5. Additional Examples