

Abstract

- Students who are visually impaired face barriers in higher-level computing courses.
- A common task is for students to design digital circuits, through writing code or creating schematics through a drag-and-drop interface.
- For students with visual impairments, drag-and-drop interfaces are not accessible, forcing them to only use code.

- However, exclusively using code is insufficient for learning digital circuit design.
- To address these problems, we are reviewing the literature to determine how current programming software is inaccessible and how to develop software that enables students who are visually impaired to create and interact with schematics.

Barriers That Blind Programmers Face

Textual Navigation

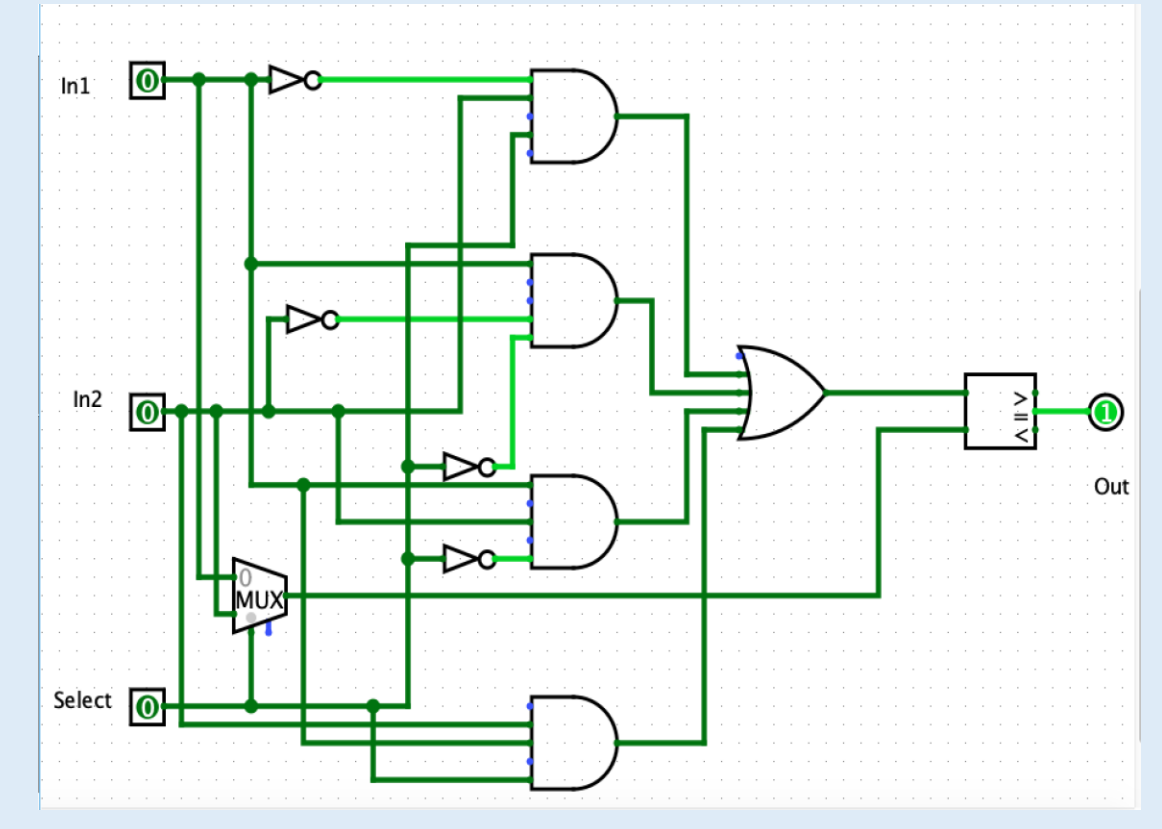
- Programmers and students usually use visual aids called Integrated Development Environments (IDE) to help write programs, but blind programmers might not get as much information as their sighted counterparts. (StructJumper)
- Context switching between lines of code or debugging might be inefficient for blind students. (StructJumper)
- Screen reader cannot help generate a general picture of the code structure for blind programmers. (StructJumper)
- Complicated syntax of certain programming language. (A Comparison of Program Strategies)

```
always_ff@(posedge Clock) begin
    if (Areset == 1'b0) begin
        state <= s00;
    end else begin
        state <= nextState;
    end
end
end
```

<https://github.com/uiowa.edu/bdmyers/public/blob/master/Lab3/fsm/OnesDetector.v>

Spatial/Graphical Navigation

- Screen reader cannot “read” graphs. (Blocks4All)
- Drag-and-drop programming is highly visual; therefore unavailable to blind learners. (Blocks4All)
- Some important graphs, often as node-link diagrams, are widely used in Computer Science, but they are often inaccessible to students and professionals. (GSK)



CS-2630 Computer Organization

Existing Solutions

Textual programming needs spatial navigation, too!

Existing Solutions

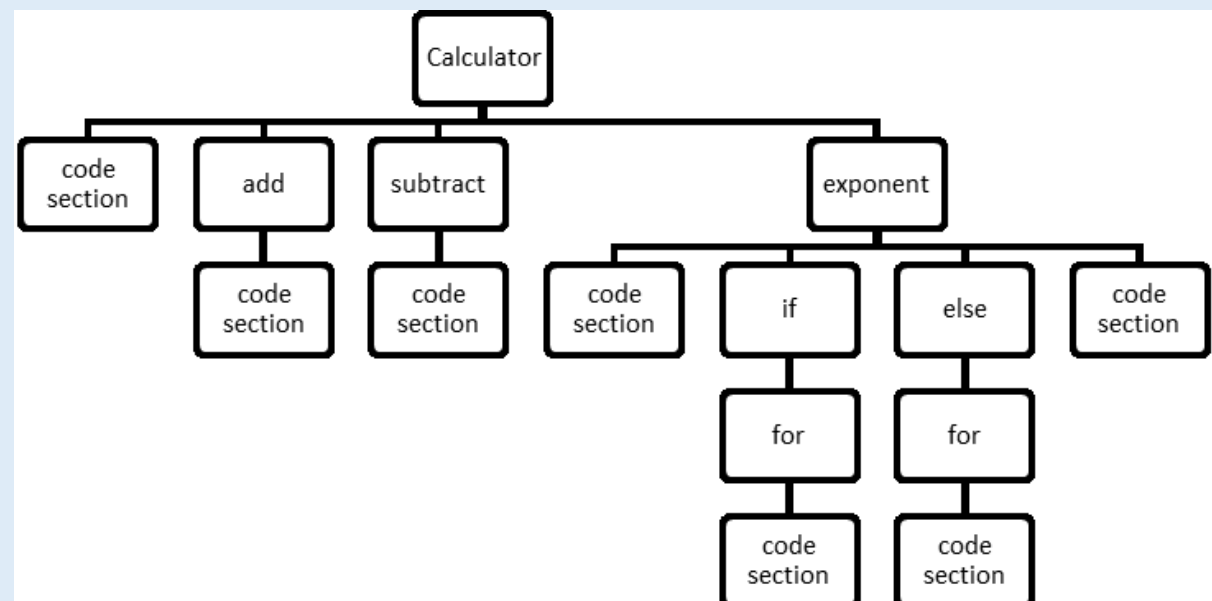
Textual Navigation

- StructJumper:**
 - A plugin to existing programming tool that creates hierarchical tree to help blind programmers navigate the code.
 - It is compatible with one of the popular programming visual aids called Eclipse.
 - It makes both navigating the code and understanding where they are within the code easier for blind programmers
 - Complicated textual code will eventually converted to a hierarchical structure to make navigation and debugging easier.

```
public class Calculator {
    private String display;
    public int add(int a, int b) {
        return a+b;
    }
    /**This method subtracts b from a*/
    public int subtract(int a, int b) {
        return a-b;
    }
    public double exponent(int a, int b) {
        double answer = 1.0;
        if(b>=0){
            for(int i = 0; i<b; i++){
                answer = answer*a;
            }
        }
        else{
            for(int i = 0; i<-b; i++){
                answer = answer/a;
            }
        }
        return answer;
    }
}
```

StructJumper: A Tool to Help Blind Programmers Navigate and Understand the Structure of Code. 2013

What sighted people see



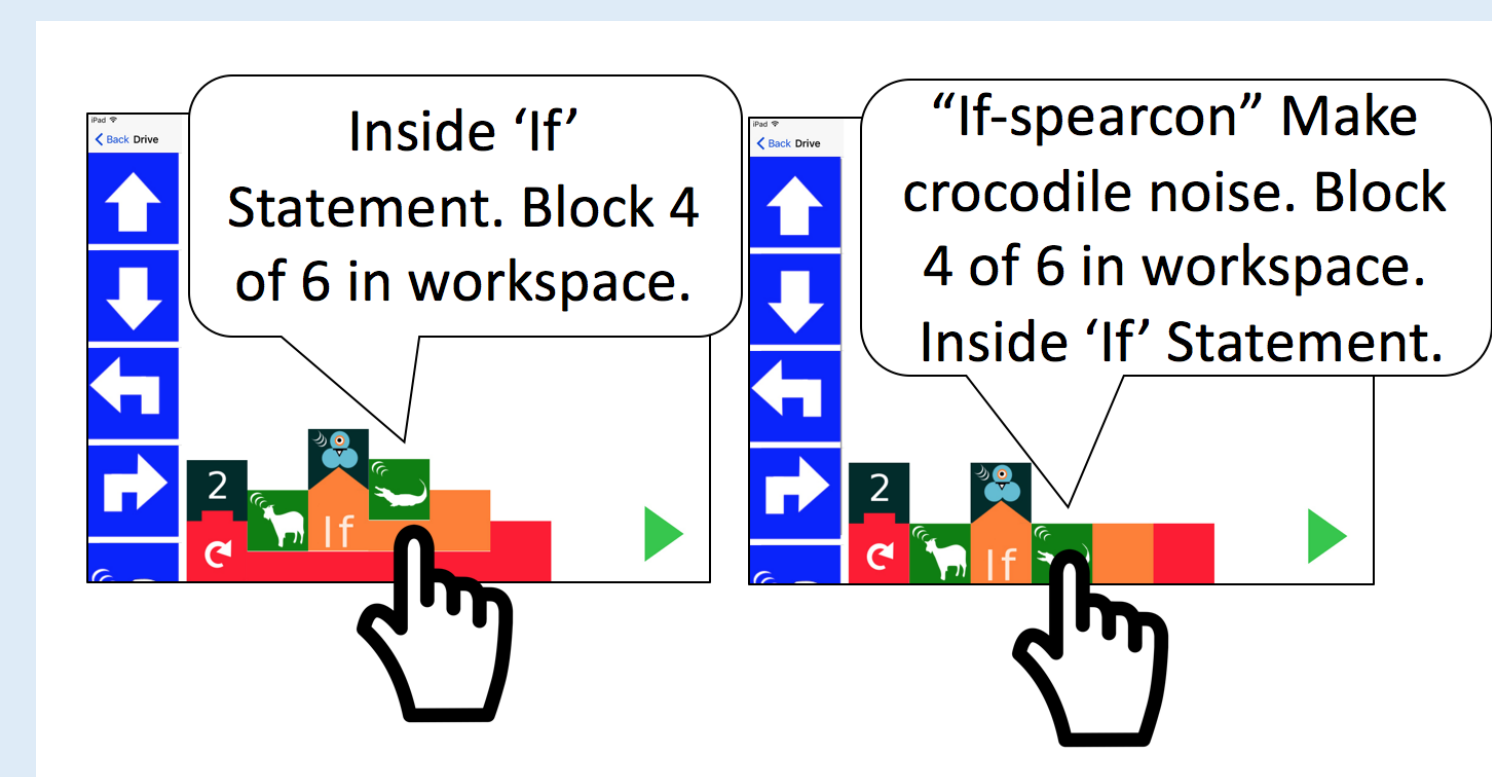
StructJumper: A Tool to Help Blind Programmers Navigate and Understand the Structure of Code. 2013

- Quorum Programming Language:**
 - A programming language designed to help people with different disabilities.
 - It has tutorial for novice programmers.
 - It uses screen reader in order to create an audio programming language environment.
 - The developers of this language simplified the syntax to make it more accessible for blind students.

What blind programmers navigate aurally

Spatial/Graphical Navigation

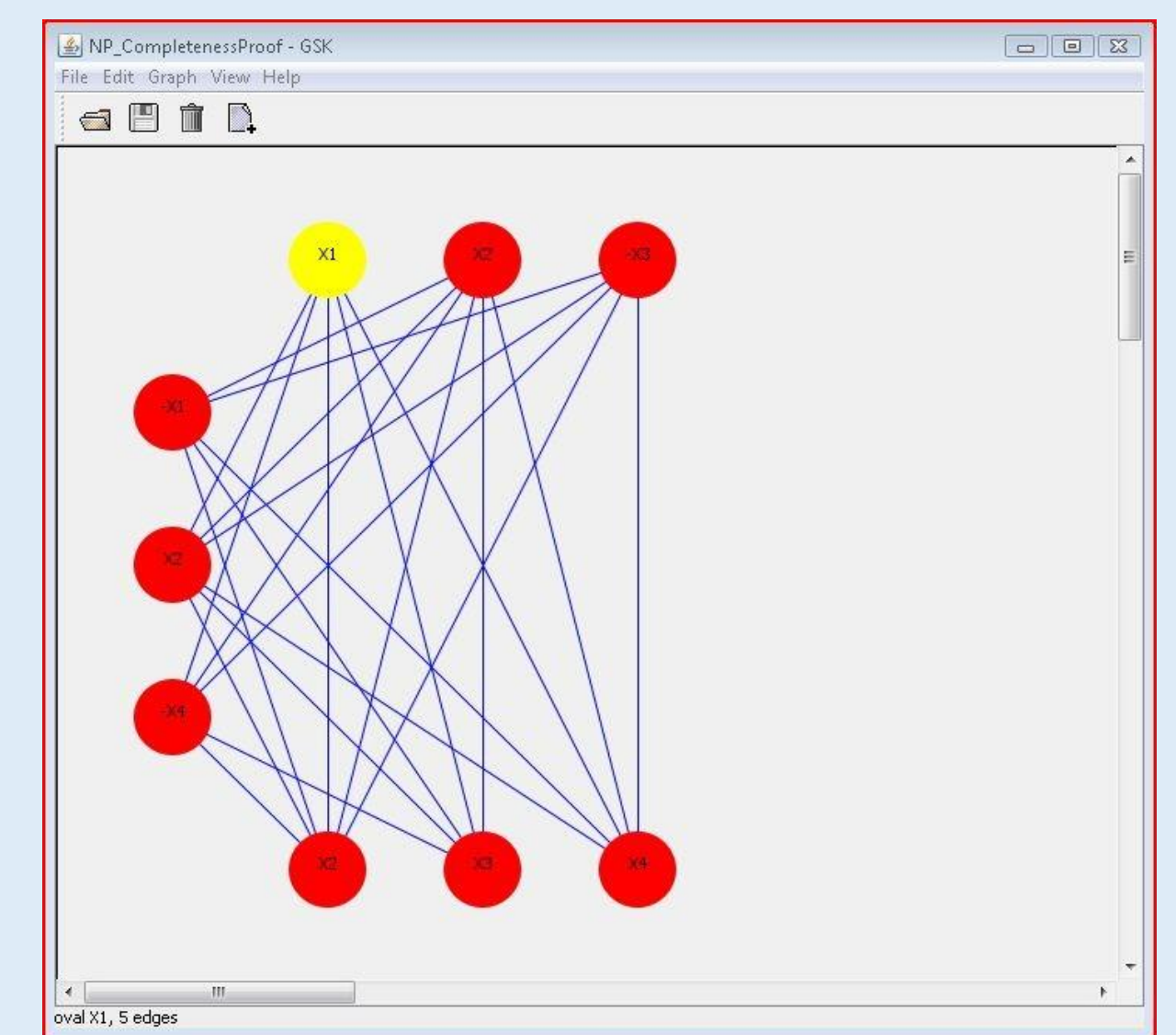
- Blocks4All:**
 - A drag-and-drop programming language primarily designed for children.
 - It simplifies the syntax of programming languages.
 - Universal designs means it is accessible for both sighted and blind children.
 - It is based on touch screen, which can be explored using speech and sound and compatible to screen reader.



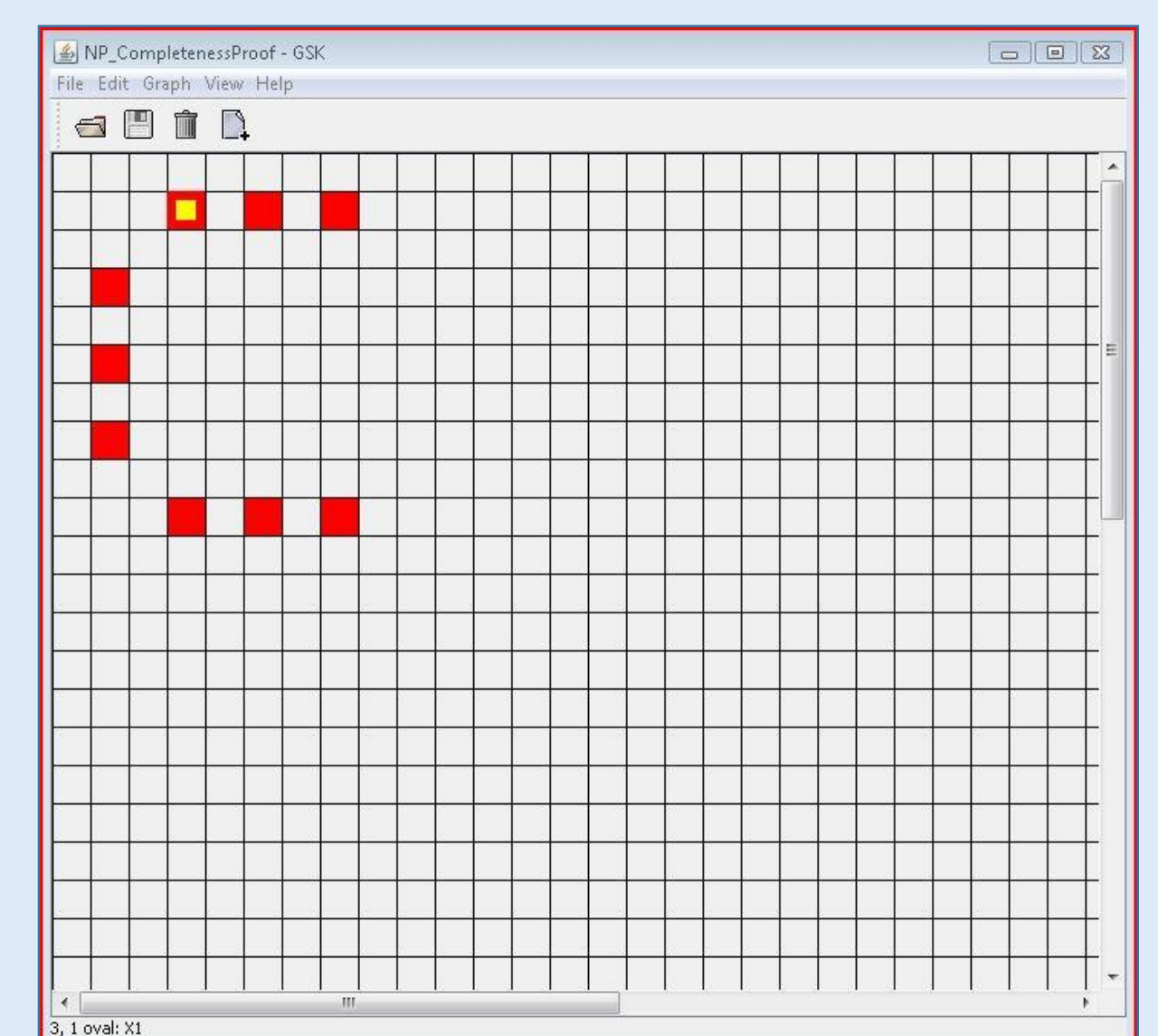
<https://miniel2.github.io/>

- Graph Sketching (GSK):**
 - It has two views – **Connection View** and **Grid View**
 - The connection view uses graph – a data structure.
 - Its connection view is accessible for both sighted and blind programmers.
 - The grid view adopts chessboard idea.
 - Both connection and grid view can guide blind people to add/remove/edit a node or a connection.
 - Its properties windows can help turn the graph into text to make it readable for screen reader.

Connection view



Grid view



GSK: universally accessible graph sketching. 2013

What's Next?

- Solutions above have some shortcomings when it comes to circuits design:**
 - Hierarchical trees usually have limited amount of layers; therefore when dealing with large chunk of code, StructJumper might be inefficient.
 - Blocks4All might not be helpful when it comes to complex circuit design, since it is designed for children.
 - GSK treats nodes and connections equally, but in circuit design, wires and endpoints/nodes might have different functions and should not be treated equally.
- We plan to focus on improving the accessibility of spatial/graphical navigation, because:**
 - Programmers tend to come up with schematics to help them structure their textual code.
 - In circuit design, textual programming like Verilog also has outputs in graphical form.
 - Graphical programming tends to be easier for beginners to understand.

- We choose Logisim, a digital circuits design language, as our primary language to modify.**
- We must make Logisim compatible with screen readers.**
- We plan to adopt some of the ideas of GSK, specifically the chessboard idea from the grid view.**
- We must modify GSK to distinguish different types of connections and nodes.**
 - In logic circuits we need a different type of node for each component (e.g., AND, OR, NOT).
 - Since each port of a component has a specific purpose, a node's connections are not interchangeable and so connections must be named or ordered.

References

- Ameer Armaly, Paige Rodeghero, and Collin McMillan. 2018. A comparison of program comprehension strategies by blind and sighted programmers. In Proceedings of the 40th International Conference on Software Engineering (ICSE '18). ACM, New York, NY, USA, 788-788. DOI: <https://doi.org/10.1145/3180155.3182544>
- Catherine M. Baker, Lauren R. Milne, and Richard E. Ladner. 2015. StructJumper: A Tool to Help Blind Programmers Navigate and Understand the Structure of Code. In Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15). ACM, New York, NY, USA, 3043-3052. DOI: <https://doi.org/10.1145/2702123.2702589>
- 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). ACM, New York, NY, USA, Paper 69, 10 pages. DOI: <https://doi.org/10.1145/3173574.3173643>
- Suzanne P. Balik, Sean P. Mealin, Matthias F. Stallmann, and Robert D. Rodman. 2013. GSK: universally accessible graph sketching. In Proceeding of the 44th ACM technical symposium on Computer science education (SIGCSE '13). ACM, New York, NY, USA, 221-226. DOI: <https://doi.org/10.1145/2445196.2445266>
- Quorum: <https://quorumlanguage.com/>

Acknowledgement

- This research is supported by ICRU.
- Special thanks to Professors Kyle Rector and Brandon Myers for the support, advice, and help.
- Special thanks to Dr. Catherine Baker for the helpful suggestions on the research.