Coding Non-visually in Visual Studio Code: Collaboration Towards Accessible Development Environment for Blind Programmers

JOOYOUNG SEO, School of Information Sciences, University of Illinois at Urbana-Champaign, USA
G.K.M. TOBIN, Institute for Clarity in Documentation, USA

In this paper, we will showcase a few examples of how blind and sighted developers have been working together to make Visual Studio Code more accessible. Through our collaboration, which is evident in the GitHub issues, pull requests, review process, and insiders releases we've created together over the past few months, we hope to help other open source developers use these methods to create more accessible development environments.

CCS Concepts: • **Computer systems organization** → **Embedded systems**; *Redundancy*; Robotics; • **Networks** → Network reliability.

Additional Key Words and Phrases: datasets, neural networks, gaze detection, text tagging

ACM Reference Format:
JooYoung Seo and G.K.M. Tobin. 2018. Coding Non-visually in Visual Studio Code: Collaboration Towards Accessible Development Environment for Blind Programmers. In . ACM, New York, NY, USA, 3 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

An integrated development environment (IDE) is an application that conveniently provides essential functions for the entire programming process, including source editing, compiling and interpreting, and debugging. IDEs have become an essential tool for not only software developers, but also STEM engineers and data scientists in many fields to efficiently manage their computing environments. However, blind developers are not able to take advantage of the many features that graphical user interface (GUI)-based IDEs offer. For example, syntax highlighting, code autocompletion and autosuggestion, diagnostics and linting, variable watches and breakpoints are underutilized even among experienced blind programmers, and many blind developers are still working manually with simple text like Notepad, along with runtime and compile terminals. Behind this problem are intertwined issues of accessibility and learnability. Because different IDEs use different architectures and have different levels of accessibility compliance, blind developers face a new learning curve each time they use an IDE. Blind developers also face the additional challenge of learning the non-visual workaround of accessing an IDE with a screen reader. Although there is a community of blind programmers called Program-L that helps each other with their struggles, IDEs remain a daunting barrier for blind people.

These difficulties are a major socio-technical barrier to blind developers reaching their full potential in the computing field and to social and professional participation. From the perspective of the social model, which recognizes that an individual's disability may stem from structures and cultures that sociotechnically limit their access, rather than from physical, sensory, cognitive, or emotional issues, we can see that IDE accessibility issues are no longer a group-specific problem that blind people must endure, but a collective task for the technology community to reduce barriers together. Specifically, to address these issues, blind and sighted developers need to work together to understand the challenges that blind developers face in using IDEs, and then collaboratively find ways to address those challenges.

This paper is the empirical product of blind and sighted developers who have thought deeply about these issues and actively collaborated. We describe how the first author, who is blind, and the second author, who is sighted, have been working together to make the open source IDE Visual Studio Code (VSCode) non-visually accessible, and what specific accessibility features have been implemented as a result of our collaboration.

In the following sections, we start with some background on how our collaboration began, then present our methods and deliverables. Finally, we'll share some insights from our collaboration.

2 BACKGROUND

2.1 Visual Studio Code and Accessibility

- Brief history of VSCode
- How accessibility commitment got started

2.2 Biographies of the Authors

The first author of this paper is blind with only light perception, currently working as an assistant professor in the School of Information Sciences at the University of Illinois at Urbana-Champaign. At the university, he teaches introductory data science courses using R and Python to undergraduate and graduate students. As a lifelong non-visual programmer, he has experience with a variety of IDEs including Visual Studio, Eclipse, and Net Bean, including text editors such as Emacs/Emacspeak, VIM, and NotePad++, on Linux, Mac, and Windows operating systems, using a variety of screen readers (e.g., JAWS, NVDA, Narrator, VoiceOver, and Orca) and refreshable braille displays. He is a certified professional in accessibility core competencies (CPACC) from the International Association of Accessibility Professionals and has contributed code to a number of open source data science projects to improve screen reader accessibility, including RStudio IDE Server and the web-based data science dashboard Shiny, reproducible tehnical publishing systems (e.g., R Markdown, bookdown, and Quarto), and the data table package gt.

- About Megan

2.3 Collaboration Methods

- How we met and collaborated
- gitHub issues/PR, email, test.

3 CASE STUDIES

3.1 VSCode Terminal Buffer

3.2 Git Diff and Audio Cues

3.3 Verbosity Settings

3.4 Copilot and Inline Suggestions

4 DISCUSSION AND CONCLUSION

5 ACKNOWLEDGMENTS

ACKNOWLEDGMENTS

We thank ...

REFERENCES