important to follow up this study with a survey administered to a larger group of participants, to verify some of our findings.

The navigation challenges identified in this study illustrate the need for further research on improving the usability and accessibility of current IDEs. For example, participants showed interest in using a new forms of code navigation, e.g. using hierarchical navigation approaches. Participants also indicated a desire for bookmarks (or tags) features that would allow blind programmers to tag specific line of code and return to it later for further modification. They also expressed interest in scope and nesting level indicator, auditory additional feedback, and methods for conveying class relationships, which could make programming more accessible for these users.

Finally, while the participants in our study expressed interest in various technology interventions to address their needs, it would be necessary in future work to conduct formal evaluations of the efficacy of such technology in studies with blind developers. In fact, we are specifically planning, in our future work, to explore some form of auditory feedback which could help convey important information while users are navigating through lengthy codebases. Several participants expressed interest in this technology. Participants also suggested that audio cues could be used in various other programming activities. We plan to conduct participatory design research to understand how to best use auditory cues in a code navigation system.

In summary, the results of this study provide future accessibility researchers a foundation for understanding the needs of blind programmers, which may support their work in creating and evaluating new technologies to address those needs.

7. ACKNOWLEDGMENTS

We acknowledge scholarship support from the King Salman Scholarship Program. We would like to thank our participants and our anonymous reviewers for their valuable feedback and insights.

8. REFERENCES

[1] K. Albusays and S. Ludi. Eliciting programming challenges faced by developers with visual impairments: exploratory study. In *Proceedings of the 9th International Workshop on Cooperative and Human Aspects of Software Engineering*, pages 82–85. ACM, 2016.

[2] C. M. Baker, L. R. Milne, and R. E. Ladner. 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*, pages 3043–3052. ACM, 2015.

[3] H. Beyer and K. Holtzblatt. *Contextual design: defining customer-centered systems*. Elsevier, 1997.

[4] D. B. Boardman, G. Greene, V. Khandelwal, and A. P. Mathur. Listen: A tool to investigate the use of sound for the analysis of program behavior. In *Computer Software and Applications Conference, 1995. COMPSAC 95. Proceedings., Nineteenth Annual International*, pages 184–189. IEEE, 1995.

[5] A. Bragdon, R. Zeleznik, S. P. Reiss, S. Karumuri, W. Cheung, J. Kaplan, C. Coleman, F. Adeputra, and J. J. LaViola Jr. Code bubbles: a working set-based interface for code understanding and maintenance. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 2503–2512. ACM, 2010.

[6] S. A. Brewster. Using nonspeech sounds to provide navigation cues. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 5(3):224–259, 1998.

[7] J. M. Francioni and A. C. Smith. Computer science accessibility for students with visual disabilities. In *ACM SIGCSE Bulletin*, volume 34, pages 91–95. ACM, 2002.

[8] A. Z. Henley and S. D. Fleming. The patchworks code editor: toward faster navigation with less code arranging and fewer navigation mistakes. In *Proceedings of the 32nd annual ACM conference on Human factors in computing systems*, pages 2511–2520. ACM, 2014.

[9] M. Konecki, A. Lovrenčić, and R. Kudelić. Making programming accessible to the blinds. In *MIPRO, 2011 Proceedings of the 34th International Convention*, pages 820–824. IEEE, 2011.

[10] J. Lamping, R. Rao, and P. Pirolli. A focus+ context technique based on hyperbolic geometry for visualizing large hierarchies. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 401–408. ACM Press/Addison-Wesley Publishing Co., 1995.

[11] S. Mealin and E. Murphy-Hill. An exploratory study of blind software developers. In *Visual Languages and Human-Centric Computing (VL/HCC), 2012 IEEE Symposium on*, pages 71–74. IEEE, 2012.

[12] National Science Foundation, National Center for Science and Engineering Statistics. 2017. *Women, Minorities, and Persons with Disabilities in Science and Engineering: 2017*. Special Report NSF 17-310. Arlington, VA. Available at www.nsf.gov/statistics/wmpd/.

[13] D. K. Palladino and B. N. Walker. Learning rates for auditory menus enhanced with spearcons versus earcons. Georgia Institute of Technology, 2007.

[14] C. Parnin and A. Orso. Are automated debugging techniques actually helping programmers? In *Proceedings of the 2011 international symposium on software testing and analysis*, pages 199–209. ACM, 2011.

[15] J. Saldaña. *The coding manual for qualitative researchers*. Sage, 2015.

[16] J. Sánchez and F. Aguayo. Blind learners programming through audio. In *CHI'05 extended abstracts on Human factors in computing systems*, pages 1769–1772. ACM, 2005.

[17] J. Sánchez and F. Aguayo. Apl: audio programming language for blind learners. *Computers Helping People with Special Needs*, pages 1334–1341, 2006.

[18] R. M. Siegfried. Visual programming and the blind: the challenge and the opportunity. *ACM SIGCSE Bulletin*, 38(1):275–278, 2006.

[19] A. C. Smith, J. S. Cook, J. M. Francioni, A. Hossain, M. Anwar, and M. F. Rahman. Nonvisual tool for navigating hierarchical structures. In *ACM SIGACCESS Accessibility and Computing*, number 77-78, pages 133–139. ACM, 2004.

[20] A. C. Smith, J. M. Francioni, and S. D. Matzek. A java programming tool for students with visual disabilities. In *Proceedings of the fourth international ACM conference on Assistive technologies*, pages 142–148. ACM, 2000.

[21] A. Stefik, R. Alexander, R. Patterson, and J. Brown. Wad: A feasibility study using the wicked audio debugger. In *Program Comprehension, 2007. ICPC'07. 15th IEEE International Conference on*, pages 69–80. IEEE, 2007.

[22] A. Stefik, C. Hundhausen, and R. Patterson. An empirical investigation into the design of auditory cues to enhance computer program comprehension. *International Journal of Human-Computer Studies*, 69(12):820–838, 2011.

[23] A. M. Stefik, C. Hundhausen, and D. Smith. On the design of an educational infrastructure for the blind and visually impaired in computer science. In *Proceedings of the 42nd ACM technical symposium on Computer science education*, pages 571–576. ACM, 2011.

[24] S. F. A. Szpiro, S. Hashash, Y. Zhao, and S. Azenkot. How people with low vision access computing devices: Understanding challenges and opportunities. In *Proceedings of the 18th International ACM SIGACCESS Conference on Computers and Accessibility*, pages 171–180. ACM, 2016.

[25] P. Vickers and J. L. Alty. When bugs sing. *Interacting with Computers*, 14(6):793–819, 2002.

[26] B. N. Walker, A. Nance, and J. Lindsay. Spearcons: Speech-based earcons improve navigation performance in auditory menus. Georgia Institute of Technology, 2006

9. ONLINE APPENDIX

Supplemental materials have been uploaded to the ACM Digital Library to accompany this paper, including two comma-separated value (CSV) files: (1) a table of additional quotations from participants and (2) a table listing the operating system, assistive technologies, programming languages, and programming editor uses by each participant (complementing the summarized information found in Table 2 in this paper). In addition, these materials are also available online at our laboratory website at the following URL: http://latlab.ist.rit.edu/assets2017code