Novice Debugging in Block-Based and Hybrid Environments

Phoebe Martinez
DePauw University
Greencastle, Indiana
phoebemartinez_2020@depauw.edu

John Lopez
Loyola Marymount University
Los Angeles, California
jlope100@lion.lmu.edu

Fernando J. Rodríguez
University of Florida
Gainesville, Florida
fjrodriguez@ufl.edu

Joseph B. Wiggins
University of Florida
Gainesville, Florida
jbwiggi3@ufl.edu

Kristy Elizabeth Boyer
University of Florida
Gainesville, Florida
keboyer@ufl.edu

ABSTRACT

Debugging is an important skill for novice programmers to master, but many students struggle to learn how to debug due in part to difficulty with program syntax. Block-based environments provide an alternative to traditional textual programming that reduces syntax errors, and recently hybrid block-based/textual environments have become more common. This poster presents preliminary research to understand how novice debugging strategies differ between block-based and hybrid environments. We assigned seven participants to debug four programs within one of the two environments and conducted interviews about their debugging approaches. Thematic analysis of interview responses suggest that students adjusted their strategies based on their prior experience with textual environments. By understanding novice programmers' strategies in these environments, the field can move toward more effectively supporting productive strategies.

CCS CONCEPTS

• **Social and professional topics → CS1**.

KEYWORDS

Debugging, Block-based programming, Hybrid environment

1 INTRODUCTION AND BACKGROUND

The behaviors and perceptions of novice programmers have been studied extensively, with findings suggesting that novice programmers struggle with understanding programming syntax and exactly what a program does [2]. These difficulties also explain why novices have trouble developing strong debugging skills [1]. This preliminary research investigates the affordances of block-based environments and hybrid block-based/textual environments for supporting novices as they debug. The research question is *how does the programming environment, block-based or hybrid, influence the debugging strategies of novice programmers?*

2 OVERVIEW, METHODS, AND RESULTS

Seven introductory programming students completed a series of debugging tasks in a purely *block-based environment* (n=3) or a *hybrid block-based and textual environment* (n=4). They were given 20 minutes to find and repair six bugs within three short programs and 25 minutes to find and repair six bugs within one longer program, derived from prior debugging studies [4]. After these tasks, each participant was interviewed about their debugging approaches.

Students in the hybrid environment condition found and fixed more bugs than students in the block-based environment condition. We conducted thematic analysis [5] of the interview responses, and the results revealed ten themes related to students' debugging approaches. Several debugging strategies were identified, including the use of 'print' statements, line-by-line analysis, and bottom-up reasoning; these strategies echo those performed in prior studies with textual environments [3]. Additionally, while some students found the block-based environment to be easier to read and utilize ($n = 3$), others found the blocks overwhelming ($n = 4$) and primarily utilized the textual interface from the hybrid environment.

3 CONTRIBUTIONS AND FUTURE WORK

The themes described participants' previous experience, different strategies used within the environments, and the pros and cons of each environment. Some of the themes conflicted with each other, such as reporting that the color coding in block programs were both easy and difficult to follow. Contradicting opinions about interface elements and the role of past exposure to text environments remain to be investigated. By understanding how students implement and adapt their debugging strategies to the interface they are using, the CS education community can enhance features of block-based and hybrid environments to encourage more productive strategies.

REFERENCES

[1] Basma S. Alquadi and Jonathan I. Maletic. 2017. An Empirical Study of Debugging Patterns among Novices Programmers. *Proceedings of the 2017 ACM SIGSCE Technical Symposium in Computer Science Education* (2017), 15–20.

[2] Emde Ducasse. 1981. A Review of Automated Debugging Systems: Knowledge, Strategies and Techniques. In *Proceedings of the 10th International Conference on Software Engineering*. 162–172.

[3] Sue Fitzgerald, Gary Lewandowski, Renee McCauley, Laurie Murphy, Beth Simon, Lynda Thomas, and Carol Zander. 2008. Debugging: Finding, Fixing and Flailing, a Multi-Institutional Study of Novice Debuggers. *Computer Science Education* 18, 2 (2008), 93–116.

[4] W Lewis Johnson, Elliot Soloway, Benjamin Cutler, and Steven Draper. 1983. *Bug Catalogue: I.* Yale University Press.

[5] Steven J Taylor, Robert Bogdan, and Marjorie DeVault. 2015. *Introduction to Qualitative Research Methods: A Guidebook and Resource.* John Wiley & Sons.

SIGCSE '20, March 11–14, 2020, Portland, OR, USA

ACM ISBN 978-1-4503-6793-6/20/03..
https://doi.org/10.1145/1122445.1122456