DOI: https://doi.org/10.3217/978-3-85125-609-3-23

Debugging Computational Design: A Pedagogical Approach to Introducing a Visual Programming Platform

Julie Lamarra[1], Breanne K. Litts[1], Whitney E. Lewis[1], & Chase K. Mortensen[1]

[1]Utah State University
{Julie.Lamarra, Breanne.Litts, Whitney.Lewis, Chase.Mortensen}@usu.edu

Abstract. Debugging or encountering and resolving bugs is a practice embedded in computational design processes. In this study, we explored how debugging can be leveraged as a pedagogical approach to introduce youth to a visual programming tool as part of a computational design process. In partnership with a local makerspace, we conducted a workshop with seven youth (6 girls, 1 boy) ages 9-15 during which they created their own mobile games. Our findings offer insights for thinking about debugging as a productive tool not only for learning and assessment but also for instruction.

Keywords: debugging, computational thinking, visual programming, pedagogy, mobile games

1 Introduction

Encountering and resolving bugs is a practice embedded in the computational design processes. A “bug” can be defined as a minor problem that is wrong within the computer code which is easily fixable [1]. Griffin and colleagues [2] introduced “debug’ems” to describe a process of intentionally creating a bug or error in a coding language as a tool to scaffold students’ learning. By intentionally creating a known error the teacher can pinpoint a desired competency that will ensure a student will practice and eventually gain mastery over the skill set, which might otherwise be overlooked during exploratory experiences [2]. In this study, we explored how debugging can be leveraged as a pedagogical approach as part of a computational design process. Specifically, we examined how a debugging activity can introduce and orient learners to the Augmented Reality and Interactive Storytelling (ARIS) platform [3], a narrative-based visual programming tool for non-programmers to build their own mobile games and experiences. ARIS consists of a web-based editor with which games are created and a client-based app with which games are played. Our inquiry was guided by the following research question: How can we leverage debugging activities as a pedagogical strategy to introduce novice programmers to a visual programming tool?

DOI: https://doi.org/10.3217/978-3-85125-609-3-23

2 Methods

In partnership with a local makerspace, we conducted a workshop with seven youth (6 girls, 1 boy) ages 9-15 during which they created their own mobile games. We introduced participants to the ARIS platform by giving them an example game to explore on an iPad, which was embedded with strategic bugs predominately focused on “locks” a key element of ARIS that controls logic. After playing this game, we introduced youth to the ARIS editor by showing them the code to the game and challenging them to debug the logic issues. We collected a range of qualitative data including field notes, interviews, video observation, screen recordings, and audio recordings. In our analysis, we examined the method of creating intentional bugs for students to resolve and how that influenced their understanding of the platform of ARIS. We collaboratively developed a codebook that utilizes iterative open (bottom-up) and elaborative (top-down) coding methods [4]. In this poster, we present key themes that we identified across participants to help shed light on debugging as a pedagogical tool for introducing novices to visual programming.

3 Results

Our findings indicate the successes and challenges of leveraging debugging as a pedagogical tool. On the one hand, participants reported that the debugging activity was an effective tool with which to familiarize themselves with the ARIS jargon and interface. For example, when asked about debugging, Jane, an 11-year-old girl, elaborated, “They [bugs] were easy to fix once you figured out what the problem was, and it was really rewarding to fix them” (Interview, December 7, 2017). Moreover, MacKenzie, a 12-year-old girl, reflected on this process and claimed, “that debugging was a good way to get introduced to ARIS” (Interview, November 28, 2017). On the other hand, some participants found the activity overwhelming. For example, Sarah, a 10-year-old girl, further expounds this idea, “Um, I didn't know where to find anything because it was my first time using ARIS,” and that “there were too many things happening at once” (Interview, February 5, 2018). A key challenge with the debugging activity was that the bugs we integrated into the code were exclusively logic bugs.

4 Discussion

These findings provide insights for thinking about debugging as a productive tool not only for learning and assessment but also for instruction. By allowing participants to search and seek out to diagnose and resolve bugs, they were authentically immersed in the platform and developed proficiency in computational design. Though some participants thought the debugging activity was helpful in starting in an unfamiliar visual programming platform, other participants were overwhelmed it. Thus, additional research is need to explore whether varying the type of bug might reduce the overwhelming response some participants had to the activity. Building on recent work exploring debugging as an assessment tool [5], our study provides evidence for using debugging activities as a pedagogical approach. Further research is needed to empirically examine

DOI: https://doi.org/10.3217/978-3-85125-609-3-23

how these debugging approaches impact specific learning outcomes in computational design activities.

References

1. Kidwell, P. (1998). Stalking the Elusive Computer Bug. *IEEE Annals of the History of Computing, 20*(4), 5-9. doi:10.1109/85.728224
2. Griffin, J., Kaplan, E., & Burke, Q. (2012). Debug'ems and other Deconstruction Kits for STEM learning. *IEEE 2nd Integrated STEM Education Conference*. doi:10.1109/isecon.2012.6204168
3. Holden, C., Gagnon, D., Litts, B. K., & Smith, G. (2014). ARIS: An open-source platform for widespread mobile augmented reality experimentation. In M. Mendes (Ed), *Technology platform innovations and forthcoming trends in ubiquitous learning.* IGI Global.
4. Saldana, J. (2009). *The Coding Manual for Qualitative Researchers*. Thousand Oaks, CA: Sage.
5. Fields, D. A., Searle, K. A., & Kafai, Y. B. (2016). Deconstruction Kits for Learning. *Proceedings of the 6th Annual Conference on Creativity and Fabrication in Education - FabLearn 16*. doi:10.1145/3003397.3003410