Kashihara, A. et al. (Eds.) (2024). Proceedings of the 32nd International Conference on Computers in Education. Asia-Pacific Society for Computers in Education

Pyzzles: Towards the Design of a Zugzwang-Inspired Learning Tool for Novice Programmers and its Effect on Debugging Skills and Self-Perceived Debugging Confidence

Elijah Justin CALLANTA
Department of Information Systems and Computer Science,
Ateneo de Manila University, Philippines
elijah.callanta@student.ateneo.edu

Abstract: This paper presents the design and preliminary results of Pyzzles (Python Puzzles), a web-based application inspired by Zugzwang principles from chess to enhance novice programmers' debugging skills. In Zugzwang puzzles, a player must decide on the move that most benefits their position. Pyzzles replicates this concept by providing learners with code snippets containing debugging errors and defining the "best debugging move" as the minimal number of lines edited to correct the code. Partial points are awarded for suboptimal corrections. Pyzzles was tested on a small cohort of 16 participants from introductory programming classes over 14 days. While Pyzzles significantly increased users' self-perceived debugging confidence, there was no significant improvement in actual debugging skills, defined as the ability to identify erroneous lines of code. The study's small sample size limits the generalizability of these findings, and the paper concludes with suggestions for future research to better integrate Pyzzles into the broader literature on debugging tools and to explore enhancements that could support both self-perceived confidence and actual debugging proficiency.

Keywords: Computers in Education, Human-centered computing, Human Computer Interaction, Software Engineering

1. Introduction

This study contributes to research aimed at improving students' debugging skills (Beaubouef, 2005). Enhancing debugging skills can reduce attrition rates among Computer Science students in secondary and tertiary education (Hoda & Andrae, 2014). The paper focuses on Debugging Confidence (O'Dell, 2017), which is linked to high debugging skill (Marzieh et al., 2005). However, there is a lack of research on investigating Debugging Confidence. This paper introduces a novel approach by incorporating Zugzwang principles into learning software. These features measure self-perceived debugging confidence and actual debugging skill. The study uses Language Specification Errors and Program Specification Errors in debugging puzzles. It aims to offer alternative methods for enhancing debugging learning and reducing attrition rates. This approach can benefit educators and researchers in teaching debugging to novice tertiary-level students.

Formally, the study set out to answer the following:

1. (RQ1) How did the Zugzwang Debugging Puzzles affect students' Debugging Skills?
2. (RQ2) What effects did the Zugzwang Debugging Puzzles have on students' Self-Perceived Debugging Confidence
3. (RQ3) Is there a relationship between the students' Self-Perceived Debugging Confidence and Debugging Skills?

2. Methodology

The study involved 16 students from two Introduction to Computer Programming classes. Over two weeks, they completed debugging tasks and pre/post-tests to measure their debugging skills and self-perceived confidence. Debugging skill was defined as the ability to edit predefined faulty code. Puzzles were based on course concepts, with harder puzzles introducing higher-level algorithms and more language specification errors. The user interface featured a minimalist design, sequential puzzle unlocking, and gamification elements like levels and walling to ensure progression and engagement.

The puzzle interface used design principles salient to Zugzwang puzzles, including windows for puzzle identification, objectives, skill and puzzle ratings, hints, expected output, and an immersive IDE for coding and testing. The scoring rubric, adapted from a CS101 course, focused on "Correct Outputs" and "Intended Functionality," tracking user edits on predefined faulty lines and assigning scores based on test cases and accurate bug identification.

Self-perceived debugging confidence was assessed via surveys before and after tests, with experimental group data comparing the percentage of accepted random puzzles to confidence ratings, introducing a novel method for evaluating confidence and skill improvement in debugging education.

3. Preliminary Results

(RQ1) Initial results indicate that users in both groups were able to better identify bugs (as defined in the methodology) in the pre-test than in the post-test. There were more instances of users identifying more than 80% of faulty lines in the pre-test puzzles than in the post-test puzzles. Overall, the experimental group outperformed the control group in almost every puzzle in the pre-test, but this result was reversed during the post-test, where the control group outperformed the experimental group in the majority of the puzzles. This would indicate that having access to the Main Puzzles in the system did not have a significant effect on enhancing the debugging skills of the specified users. However, this result may be inconclusive given the sample size of the study.

(RQ1) Additionally, initial testing resulted in a pre-test average score of 5.32 (out of 10) for the control group, while the experimental group had a score of 5.45. The post-test average score for the control group was 5.15, while the experimental group scored 4.66. Despite the experimental group performing better initially, the control group scored higher in the post-test. This aligns with findings for RQ1, which addressed bug identification performance. This supports Miljanovic and Bradbury's (2017) finding that identifying bugs first can improve subsequent bug-fixing ability. A Mann-Whitney U test revealed p-values of p=1.00 for pre-test scores (U=35.5) and p=0.694 for post-test scores (U=30.5), indicating no statistically significant impact of Pyzzles' "Main Puzzles" on assessment scores or performance.

(RQ2) Regarding RQ2, initial testing resulted in a pre-test average confidence of 66 (out of 100) for the control group, while the experimental group had an average score of 56. The post-test average confidence for the control group was 72, while the experimental group had a score of 69. A Spearman rank correlation test found no significant relationship between confidence and the ratio of random puzzles accepted (p=.736). However, the Wilcoxon signed-rank test (W=18.5) revealed a significant increase in self-perceived debugging confidence (p=.05), indicating a 16.27% boost in confidence after using Pyzzles. Despite the small sample size (n = 16), these preliminary results suggest that Pyzzles significantly improves novice programmers' confidence. Further research with a larger sample is recommended.

(RQ3) Spearman rank correlation between post-test self-perceived debugging confidence and post-test average score, gave me a correlation coefficient of 0.24 and a p-value of 0.35. This suggests that at a 95% confidence level, there was no statistically significant relationship between a user's self-perceived confidence in the post-test survey and their average score for the post-test assessment. I'd like to note that even if the correlation were significant, the correlation coefficient would indicate a weak relationship, which does not present any conclusive evidence that the two variables were correlated.

4. Conclusion

The study introduces Pyzzles (Python Puzzles), a web-based application designed using Zugzwang principles from chess to improve debugging skills. The “best debugging move” was defined as the minimal number of lines edited to fix code. Results showed a 16.2% increase in self-perceived debugging confidence (SPDC) among participants, df=16 p≤.05. (RQ2), with higher SPDC correlating with more exposure to Pyzzles puzzles. However, there was no conclusive evidence that increased SPDC improved actual debugging skills (RQ1), and there was no statistically significant relationship between a user's self-perceived confidence in the post-test survey and their average score for the post-test assessment (RQ3). The control group outperformed the experimental group, possibly due to the small sample size. The study highlights the potential of Zugzwang-based methods in teaching debugging and suggests further research with larger samples to validate these findings.

Acknowledgements

I would like to thank his thesis advisers, Dr. Jenilyn Agapito-Casano and Mr. Jonathan Casano, whose help was very valuable to the study. In addition to this, the author would like to thank Mr. Arjo Mejilla for his interest in the study as well as for his help in gathering participants for this study.

References

Agapito, J., & Rodrigo, M. M. (2018). Identifying meaningful gamification-based elements beneficial to novice programmers. Proceedings of the 26th International Conference on Computers i.

Beaubouef, T. (2005). Why the high attrition rate for computer science students: Some thoughts and observations. Association for Computing Machinery Special Interest Group on Computer Science Education Bulletin.

Hoda, R., & Andreae, P. (2014). It’s not them, it’s us! Why computer science fails to impress many first years. Proceedings of the Sixteenth Australasian Computing Education Conference.

Marzieh, A., Elliman, D. G., & Higgins, C. A. (2005). An analysis of patterns of debugging among novice computer science students. Association for Computing Machinery Special Interest Group on Computer Science Education Bulletin.

Miljanovic, M. A., & Bradbury, J. (2017). RoboBUG: A serious game for learning debugging techniques. The 2017 Association for Computing Machinery Conference.

O’Dell, D. H. (2017). The debugging mindset: Understanding the psychology of learning strategies leads to effective problem-solving skills. Association for Computing Machinery Queue.

Papastergiou, M. (2009). Digital game-based learning in high school computer science education: Impact on educational effectiveness and student motivation. Computers and Education.

Şar, A. H., Avcu, R., & Işiklar, A. (2010). Analyzing undergraduate students’ self-confidence levels in terms of some variables. Procedia-Social and Behavioral Sciences.

Schalich, M. E. (2015). Analysis of pre-test and post-test performance of students in a learning center model at the elementary school level. Dominican University of California.

Vartanian, E. (2021). 10 common mistakes Python programmers make (and how to fix them). Educative.