

# Labeling Debugging in May's Journey Gameplay

Ma. Victoria Almeda

TERC

2067 Massachusetts Ave, MA 02140,  
mia\_almeda@terc.edu

Erica Kleinman

Northeastern University,

360 Huntington Ave, MA 02115,  
kleinman.e@husky.neu.edu

Chaima Jemmali

Northeastern University,

360 Huntington Ave, MA 02115,  
jemmali.c@husky.neu.edu

Carter Ithier

Northeastern University,

360 Huntington Ave, MA 02115,  
ithier.c@husky.neu.edu

Elizabeth Rowe

TERC

2067 Massachusetts Ave, MA 02140,  
elizabeth\_rowe@terc.edu

Magy Seif El-Nasr

Northeastern University,

360 Huntington Ave, MA 02115,  
m.seifel-nasr@northeastern.edu

## ABSTRACT

Debugging is an integral part of programming and is difficult to master among novice programmers. We examine how middle-school learners approach debugging within a puzzle game, *May's Journey*, using a Playback Tool that provides a replay of student gameplay to support teacher practice and research. Teachers and researchers can use our labeling system and Playback Tool to observe students' debugging process and identify those who are struggling to learn how to debug their code.

## CCS CONCEPTS

• Human-centered computing ~ Visualization systems and tools

## KEYWORDS

Debugging; labeling; programming; digital games; playback analysis

## 1 INTRODUCTION AND RELATED WORK

In this poster, we focus on gameplay behaviors of middle school students in order to understand how young learners practice debugging in *May's Journey*, a 3D puzzle game that teaches the basics of programming by having players type simple instructions in the game's custom programming language to interact with objects and navigate through an environmental maze [1].

Murphy et al. [2] conducted quantitative analyses of debugging logs and final debugging solutions from novice programmers to identify unproductive and productive debugging strategies. Building on this research, this paper reports on the human-labeling of debugging in *May's Journey* using a Playback Tool that shows both programming and maze views, with a detailed recording of a player's programming process. This tool has been used to study implicit learning in physics and computational thinking games [3]. This work asks: What behavioral indicators of debugging in gameplay can humans reliably label?

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the Owner/Author.  
SIGCSE '20, March 11–14, 2020, Portland, OR, USA  
© 2020 Copyright is held by the owner/author(s).  
ACM ISBN 978-1-4503-6793-6/20/03.  
<https://doi.org/10.1145/3328778.3372624>

## 2 METHODS AND RESULTS

Using the Playback Tool, six researchers developed a human labeling system of debugging by independently watching 28 levels of gameplay from nine 5th to 9th grade students. These students finished between the 1st and 11th levels of the game. We discussed labeling as a team and iteratively revised the labeling system to incorporate emergent *May's Journey* gameplay behaviors. Examples of our debugging labels include:

**Phases of Debugging Puzzle** – How the player is approaching the solution to the puzzle

**Debugging Puzzle Efficiency** – The efficiency with which the player is able to discover and implement the solution to the puzzle

**Phases of Debugging Code** – How the player is writing and debugging code as he or she progresses through a level

**Debugging Code Efficiency** – The efficiency with which the player is able to correct mistakes in their code

**Overarching Strategies to Write Code** – How the player is approaching the process to write code

**Code Writing Strategies** – Moment to moment behaviors regarding how the player goes about developing their code

## 4 CONTRIBUTIONS AND FUTURE WORK

Because this Playback Tool is able to show players' progression in solving puzzles and writing code, it offers researchers and teachers deep insight into emergent patterns of debugging from novice programmers. With the labeling system proposed, our next step is to establish inter-rater reliability with a small sample of videos. In future work, we will build automated detectors of debugging grounded on human-applied labels that predict players' debugging efficiency at a large scale and in real time.

## ACKNOWLEDGMENTS

We are grateful NSF/AISL grant #1810972.

## REFERENCES

- [1] Jemmali, C., Bunian, S., Mambretti S., and Seif El-Nasr, M. 2018. Educational Game Design: An Empirical Study of the Effects of Narrative. In Proceedings of FDG, ACM Press, New York, NY, 1-34.
- [2] Murphy, L., Lewandowski, G., McCauley, R., Simon, B., Thomas, L., and Zander, C. 2008. Debugging: the good, the bad, and the quirky – a qualitative analysis of novices' strategies. In SIGCSE, ACM Press, New York, NY, 163-167.
- [3] Rowe, E., Asbell-Clarke, J., Almeda, M., Bardar, E., Baker, R. S., and Scruggs, R., 2019. Advancing Research in Game-Based Learning Assessment: Tools and Methods for Measuring Implicit Learning. In E. Kennedy & J. Qian (eds.) *Advancing Educational Research with Emerging Technology*. IGI Global, Pennsylvania, In press.