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.seif-el-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.

CSCS 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?

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