Mazzy: A STEM Learning Game

Dominic Kao, D. Fox Harrell
Computer Science and Artificial Intelligence Laboratory
Massachusetts Institute of Technology
Cambridge, Massachusetts 02139, USA
{dkao,fox.harrell}@mit.edu

ABSTRACT

In this paper we present *Mazzy*[1][2], a game developed to foster computational literacy [1] and as an experimental testbed for evaluating the impacts of avatar type on performance and engagement. The latter aspect has been reported on elsewhere [3,4], this paper focuses on the design of *Mazzy* itself. In *Mazzy*, players write programs to guide their character through a maze. In doing so, players must create short computer programs, use procedural thinking, and debug issues in their algorithmic structures. *Mazzy* is based on constructionism, a pedagogical approach in which building objects is central to the process of learning [7]. We describe the state of the current game, and report on the most recent progress.

Categories and Subject Descriptors

K.8.0 [**Personal Computing**]: General – Games

Keywords

Educational games, computer science learning

1. INTRODUCTION

Mazzy is a Science, Technology, Engineering and Mathematics (STEM) learning game, designed to be an engaging game, but also fostering computational thinking. *Mazzy* has been used as a framework for studying virtual identities in a STEM setting [3,4]. Players use the keyboard to write procedures to guide a character in a maze (see Figure 1). Levels become increasingly difficult, unlocking new modes, giving players access to more complex mechanisms (e.g., loops, if blocks). Using these mechanisms in an effective and creative manner is key to solving harder mazes.

2. THEORETICAL FRAMEWORK

Mazzy is based on constructionism, a pedagogical approach in which building objects is central to the process of learning [8]. In *Mazzy*, players play the game by building their own computer programs. Constructionism is based on the principle that learning a new concept or idea is easier if it can be assimilated into existing models [9]. In *Mazzy*, the character is "body syntonic" [7]; this means that players are drawing on well-established knowledge of motion. Players are learning computing by creating programs via a real concrete object that can be manipulated.

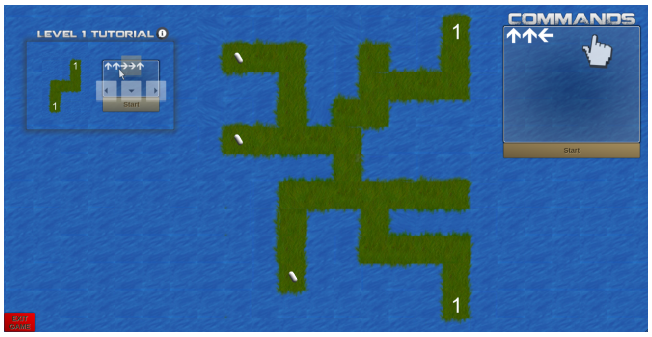

Figure 1: In *Mazzy*, players write "code" to navigate a maze.

Mazzy is inspired by the Logo programming language [6]. Logo (modeled on the language LISP) was especially popular for its turtle graphics. In turtle graphics, an on-screen turtle is designed to carry out drawing functions. *Mazzy* has its similarities (e.g., they are both designed to teach the same type of thinking [1,5]), but *Mazzy* is designed explicitly as a video game to provide more opportunities for engagement. This means *Mazzy* has scoring, specific goals, game progression, etc.

3. CURRENT GAME

Mazzy uses symbolic representation for code instead of natural language. This has several advantages: 1) syntactical simplicity; syntax errors are not possible by design, and 2) learnability; the symbols are meant to explain themselves. When players run a program, each symbol is highlighted as it is processed; this "always-on debugging" stems from the philosophy that building systems is an iterative process, that things almost never work on the first try.

Mazzy has three levels. Each of the levels features animated tutorials to guide the player. Bonus items challenge players to solve levels in a more complex manner. Levels become harder; for instance level two requires the player to program multiple characters in parallel, and level three requires the

[1]Current game: http://groups.csail.mit.edu/icelab/mazzy/
[2]Newest work: http://groups.csail.mit.edu/icelab/mazzy/p/

Proceedings of the 10th International Conference on the Foundations of Digital Games (FDG 2015), June 22-25, 2015, Pacific Grove, CA, USA.
ISBN 978-0-9913982-4-9.

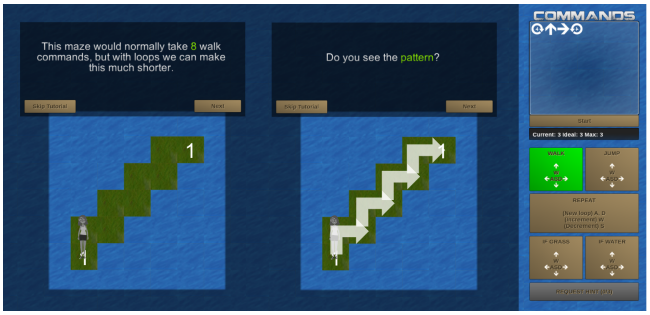

Figure 2: This level introduces loops. The program reads "repeat four times, move up, move right, end repeat".

player to program boolean logic into the level map. The game has been designed to be challenging for experimental purposes. In our experiments in which participants are given the option of quitting the game at any time, about 1/3 of the participants complete the entire game.

In our studies using Mechanical Turk [4], players described the game as a "skill based programming game." Players made analogies to the games *Chip's Challenge*, *RoboRally*, *Snake*, *Pacman* and *Dig Dug 2*. We often received positive feedback, e.g., "I love this game! I want to play more. Is this a real thing? Can I download it somewhere?" This is gratifying, as engagement is one of our goals.

4. RECENT DEVELOPMENTS

In our newest work, we have focused on integrating a more incremental progression. Levels are "mind-size bites" [7]; levels progress into requiring the player to apply more complex mechanisms to solve a given maze. The twelve levels total guide the player in learning the modes (e.g., walk mode for basic movement, repeat mode for creating loops, etc.). Interspersed are hands-on tutorials introducing each new concept. See Figure 2.

Players that complete the game have learned how to use repeating structures (e.g., "for loops") and conditionals (e.g., "if blocks"). See Figure 3. Researchers have found that programming bugs associated with loops and conditionals are among the most common [10], making understanding these constructs even more crucial. *Mazzy* is designed to keep programs syntactically correct at all times. For instance, deleting the start or end symbol for a loop deletes both symbols. Each level challenges players to use an ideal (optimal) number of symbols. Moreover, we have developed a hint system that allows users who are stuck to seek help.

5. CONCLUSION

We have presented *Mazzy*, a STEM learning game designed to foster computational thinking. Players in *Mazzy* create programs to solve increasingly complex mazes. The solutions to said mazes' often require using loops and conditionals; core concepts in the computing sciences. *Mazzy* is a novel task that requires successful application of these concepts, and as such players that complete *Mazzy* have learned these crucial constructs.

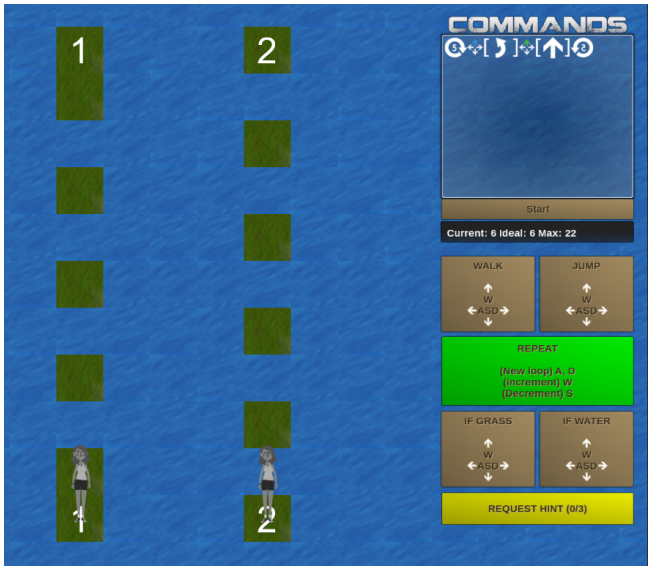

Figure 3: This level uses conditionals nested in a loop to move two characters using a single program. The program reads "repeat five times, if water above, jump up, end if, if grass above, move up, end if, end repeat".

6. FUTURE WORK

In future work, we plan to allow users to create their own maps in *Mazzy*; drawing on notions of critical pedagogy [2] in which a co-construction of reality occurs between the game and the player. This will allow players to explore their own generative themes such as mapping their neighborhoods and communities to the game assets they create.

7. ACKNOWLEDGMENTS

We thank the anonymous reviewers for their feedback. This work is supported by National Science Foundation (NSF) Award IIS-1064495 and a Natural Sciences and Engineering Research Council of Canada (NSERC) fellowship.

8. REFERENCES

[1] A. A. DiSessa. *Changing Minds: Computers, Learning, and Literacy*. MIT Press, 2001.

[2] P. Freire. Education for Critical Consciousness. In *The Paulo Freire Reader*, pages 80–110. 1998.

[3] D. Kao and D. F. Harrell. Exploring construction, play, use of virtual identities in STEM learning. *Jean Piaget Society Annual Conference*, 2015.

[4] D. Kao and D. F. Harrell. Toward Evaluating the Impacts of Virtual Identities on STEM Learning. *Foundations of Digital Games*, 2015.

[5] D. Knuth. Literate programming. *The Computer Journal*, 1984.

[6] Logo: http://el.media.mit.edu/logo-foundation/.

[7] S. Papert. Mindstorms: Children, Computers, And Powerful Ideas, 1993.

[8] S. Papert and I. Harel. Situating Constructionism. *Constructionism*, 1991.

[9] J. Piaget. *Piaget and His School*. Springer Berlin Heidelberg, Berlin, Heidelberg, 1976.

[10] J. C. Spohrer and E. Soloway. Novice mistakes: are the folk wisdoms correct? *Communications of the ACM*, 29(7):624–632, 1986.