Assessing Implicit Computational Thinking in Zoombinis Gameplay

Elizabeth Rowe
Jodi Asbell-Clarke
Santiago Gasca
EdGE @ TERC
2067 Massachusetts Avenue
Cambridge, MA 02140
elizabeth_rowe@terc.edu
jodi_asbell-clarke@terc.edu
santiago_gasca@terc.edu

Kathyrn Cunningham
School of Interactive Computing
Georgia Institute of Technology
Atlanta, GA 30332
kcunningham@gatech.edu

ABSTRACT

In this study we examine how playing *Zoombinis* can help upper elementary and middle school learners build implicit computational thinking (CT) skills. Building on prior methods used with the digital science learning games, *Impulse* and *Quantum Spectre*, we are combining video analysis and educational data mining to identify implicit computational thinking that emerges through gameplay [1]. This paper reports on the first phase of this process: developing a human labeling system for evidence of specific CT skills (e.g., problem decomposition, pattern recognition, algorithmic thinking, abstraction) in three *Zoombinis* puzzle by analyzing video data from a sample of elementary learners, middle school learners, game experts, and computer scientists. Future work will combine these human-labeled video data with game log data from these 70+ learners and computer scientists to create automated assessments of implicit computational thinking skills from gameplay behaviors in large player audiences. This poster with video examples will share results of this work-in-progress.

CCS CONCEPTS

•Social and professional topics → Computational thinking; *Student assessment; K-12 education;* •Applied computing → Computer games

KEYWORDS

Implicit learning; Computational thinking; Video analysis; Learning games;

ACM Reference format:

FDG'17, August 14-17, 2017, Hyannis, MA, USA

ACM ISBN 978-1-4503-5319-9/17/08.
http://dx.doi.org/10.1145/3102071.3106352

1 INTRODUCTION

Zoombinis is an award-winning, popular learning game that focuses on computational thinking. Originally designed as a puzzle game situated in the problems of database design, the dynamic suite of 12 puzzles, each with four levels, allows ample scaffolded problem-solving for young learners (ages 8 and above).

With the re-release of *Zoombinis* for tablets, desktops, and Chomebooks, the authors are conducting a national implementation study with classes in grades 3-8 to understand how students implicitly learn computational thinking in *Zoombinis* gameplay. This study has two major components:

a. To observe gameplay and build data detectors to identify implicit learning of computational thinking from player behaviors in the game, and

b. To examine how teachers can bridge that implicit learning to explicit learning through classroom activity.

This paper reports on the first phase of research towards part A - human-labeling gameplay observations to lay the groundwork for building data mining models and detectors. This work is grounded in the notion of implicit game-based learning assessments (GBLA), where game behaviors are thought to reveal knowledge that may go unexpressed in typical assessments used in school and educational research [1, 2].

2. IMPLICIT COMPUTATIONAL THINKING

"We know more than we can tell" [Polanyi, 3]

Implicit knowledge may not yet be articulated by the learner but is demonstrable through behaviors. Today's STEM learning assessments are typically laden with terminology that may present barriers to learners' expression of their underlying knowledge, and self-contained or decontextualized tests do not call upon previous knowledge or experience of learners to support new learning [4]. Game-based learning assessments show promise to provide a stealth method of assessing content and skills outside of school-like tests [5]. For research on *Zoombinis*, we defined a learning progression of computational thinking and problem solving skills to guide our labeling of strategies and behaviors in gameplay consistent with facets of the progression (Figure 1).

Drawing from various definitions of computational thinking from research and practice [6-9], we have defined a learning progression that we hypothesize can be operationalized within *Zoombinis* gameplay.

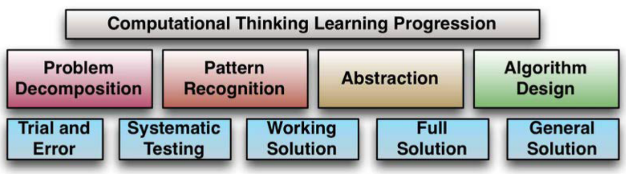

Figure 1: An Iterative Learning Progression of CT that is operationalized in *Zoombinis* gameplay

While this graphic is linear, the progression is iterative in practice with novices and experts moving back and forth through the steps, though possibly at different rates. The key elements of computational thinking that we hypothesize will be evident in *Zoombinis* gameplay are:

- **Problem decomposition**: the reduction of ambiguity or complexity of a problem by breaking it into smaller, more manageable parts. This is comparable to isolating variables or systems to test.
- **Pattern Recognition**: the recognition that objects are arranged following a rule or rules and the identification of groups of solutions or characteristics of solutions that can be categorized.
- **Abstraction:** the removal of details to identify and extract relevant information to define main idea(s) or solutions.
- **Algorithm Design**: the creation of an ordered list of instructions for solving a problem or doing a task. Also, the creation or explication of general solutions to a problem or family of problems. Often in problem-solving and design, these CT skills are exhibited in a progression of behaviors across problems.

In applying CT understanding to coding and design projects, learners may exercise additional facets of CT not in our current progression, such as efficiency and performance constraints, and debugging and systematic error detection [10]. These are being explored through research on coding environments such as Scratch [11] And Alice [12].

We have similarly defined specific, iterative Phases of Problem Solving that are intertwined with expressions of CT:

1. **Planning and preparation:** Using techniques that precede puzzle play to make more efficient and effective problem-solving.
2. **First attempt:** First move in the puzzle with no planning or preparation.
3. **Trial & Error:** No evidence of testing hypotheses in an ordered, planned way. Actions are independent of prior actions.
4. **Systematic Testing:** Testing hypotheses about underlying rule in an ordered, planned way. Next action depends on previous action. Goal of this phase is finding a working solution to implement.
5. **Systematic Testing with a Partial Solution:** Testing hypotheses about a second dimension of the underlying rule when the first dimension is known.
6. **Implementing a Full Solution:** Completing the pattern once the FULL working solution has been found. There should be no errors in this phase.

These facets of CT are demonstrated in their progression from **Trial and Error**, where there is no systematic pattern to the behaviors towards **Systematic Testing**, typically involving problem decomposition. When players recognize patterns in solutions to the smaller problems, they abstract that towards general rules so that they can then move towards **Implementing a Solution** to the larger puzzle. When learners encounter new puzzles that require similar solutions, they may begin to **Generalize Solutions** leading toward algorithm design. The central question this research addresses is: ***What are the behavioral indicators of implicit computational thinking in Zoombini gameplay that humans can reliably label?***

3. METHODS

We are building automated tools that can use *Zoombinis* gameplay data to provide information about players' implicit learning in the game using the same process we did for the physics game, *Impulse* [1]. To do this we:

1. Video record and then human label *Zoombinis* gameplay from beginners and expert players, children and adults, to capture the variety of strategies used to solve the puzzles.
2. Merge human labels with log data generated by the gameplay
3. Distill log data into features useful for measuring strategies that appear in the videos, focusing on the strategies that are consistent with CT.
4. Build detectors of players' CT strategies in the gameplay log grounded in human labeling.
5. Validate the detectors as formative assessments of implicit CT by comparing the performance of learners on external pre/post assessments of similar content.

This paper reports details about human labeling system developed from analysis of 77 videos as the first step in the development of GBLA of implicit computational thinking.

3.1 Sample & Procedures

Forty-two elementary students in grades 3-5, 28 students in grades 6-8, and seven computer scientists from the northeastern U.S. have thus far participated in our playtesting. Of these 77 participants, there were 33 females and 44 males. Participants were recruited from local schools and clubs, as well as after-school programs. Playtesting sessions last approximately 1 hour. ScreenFlow [13] is used to screen capture players' activity and resulting game states; video of facial expressions and gestures; and audios of their discussions while playing. When learners play individually, they are asked to think aloud as they play. Groups of players are asked to talk with each other as they play.

3.2 The Game: Zoombinis

The goal of *Zoombinis* is for players to rescue 400 Zoombini characters from the evil Blouts on Zoombini Isle by safely moving them through 12 increasingly complex puzzles and arrive safely in Zoombiniville (Figure 2). Three *Zoombinis* puzzles are the focus of this paper: Allergic Cliffs, Pizza Pass, and Mudball Wall.

Figure 2: 4 *Zoombinis* Screenshots. (1) Puzzle Map with labels (top left); (2) Allergic Cliffs (top right); (3) Pizza Pass (bottom left); and (4) Mudball wall (bottom right)

3.2.1 Allergic Cliffs. The Zoombinis must cross two bridges spanning a chasm. Each bridge is accompanied by a cliff face that is allergic to one or more Zoombini traits. Players choose which of the two bridges each Zoombini should cross. Each Zoombini that causes a cliff face to sneeze is knocked back along the bridge to the starting side, and one of the six pegs holding both bridges up is dislodged. When all six pegs are gone, both bridges collapse, stranding the remaining Zoombinis.

3.2.2 Pizza Pass. The Zoombinis' path is blocked by one or more trolls that demand a meal (pizza, or pizza and sundae) with a specific set of toppings. However, the trolls only say whether (a) they want more, (b) don't like at least one of the toppings, or (c) the meal is perfect. If there is more than one troll, each troll must receive his or her particular meal preference.

3.2.3 Mudball Wall. A large wall split into grid-squares blocks the Zoombinis' progress. Three Zoombinis line up on planks at the bottom of the screen, waiting to be launched over the wall. Each grid-square of the wall contains 0-3 dots, indicating how many Zoombinis will be launched over the wall. The launch is triggered when the player fires a mud-ball onto a grid-square containing dots. A machine allows players to choose the shape and color of the next mud-ball to fire. The shape and color determine the landing position of the mud-ball on the wall. There is a limited amount of mud, and only those Zoombinis who make it over the wall by the time the mud runs out are safe.

4. HUMAN LABELS OF IMPLICIT COMPUTATIONAL THINKING

The four authors have varying levels of computer science and *Zoombinis* play experience. We started with the definitions described in the 'Implicit Computational Thinking' section and an initial set of behavioral indicators. We iteratively watch 2-3 videos independently, discuss our labeling as a team, and revise the labeling system to incorporate emergent gameplay behaviors.

Table 1: Gameplay behavioral indicators of implicit computational thinking skill in Allergic Cliffs

Gameplay Behaviors	Labels
Selecting Zoombinis with the same common attributes (e.g., nose color & eyes) 3+ times in a row and sending them over the same bridge.	**Systematic Testing**—testing one attribute at a time **Problem Decomposition**—isolating attributes **Pattern Recognition—**placing all Zoombinis with blue noses over the top bridge
Placing Zoombinis on the appropriate bridges once they have tested all values of the attribute they believe the bridges are using.	**Implementing a Full Solution** **Abstraction**—they generalize to the attribute level (nose color) rather than values (blue vs. yellow noses).

Table 2: Gameplay behavioral indicators of implicit computational thinking skill in Pizza Pass

Gameplay Behaviors	Labels
Selecting one pizza or ice cream topping at a time. After all topics have been tried, placing all those the troll likes on one pizza.	**Systematic Testing**—testing one topping at a time **Problem Decomposition**—isolating toppings **Pattern Recognition**—selecting toppings the troll accepts
Selecting one pizza or ice cream topping at a time until they find one a troll likes. They retain the desired topping on all future pizzas and add new toppings one at a time	**Systematic Testing**—testing one topping at a time **Problem Decomposition**—isolating toppings **Pattern Recognition**—selecting toppings the troll hasn't rejected

Table 3: Gameplay behavioral indicators of implicit computational thinking skill in Mudball Wall

Gameplay Behaviors	Labels
Creating mudballs by holding one attribute (shape or color) constant	**Systematic Testing**—testing one mudball attribute at a time **Problem Decomposition**—isolating mudball attributes to columns or rows **Pattern Recognition**—recognizing whether common attribute is in the same column or row
Varying both mudball attributes so that all values have been tested within the first 5 moves	**Systematic Testing**—systematically varying mudball attributes **Pattern Recognition**——recognizing whether common attribute is in the same column or row **Abstraction—**generalizing mudball values to attributes (e.g., rows are color, columns are shapes)

While the labels are consistent across puzzles, the behavioral indicators used to identify each vary by puzzle. Tables 1-3 describes sample gameplay indicators of computational thinking skills for the Allergic Cliffs, Pizza Pass, and Mudball Wall puzzles, respectively. In all cases, if a sequence of behaviors is repeated across multiple rounds (attempts to solve the puzzle), it is considered evidence of Algorithm Design.

5. IMPLICATIONS & NEXT STEPS

We have found evidence of implicit computational thinking in 3 of the 12 *Zoombinis* puzzles. Our next steps are to identify behavioral indicators for other *Zoombinis* puzzles and to establish inter-rater reliability for the labeling system with a small sample of videos. The ultimate objective of this work is to build automated detectors of implicit computational thinking grounded in human labels applied to all video data.

The measurement of implicit learning may enable the assessment of a broad array of diverse learners, even those who are unable to express their knowledge on a traditional exam. This work shows and example of using data mining methods to measure implicit learning through behaviors exhibited in digital environments, which may be particularly important for learners with cognitive differences [14]. The ability to measure computational thinking through behavior-based data generated by digital environments might provide novel forms of assessments leading to an inclusive STEM education and workforce opportunities.

ACKNOWLEDGMENTS

We are grateful for NSF/EHR/DRK12 grant#1502282 and the study participants. We are thankful for the many contributions of our research group, the Educational Gaming Environments group at TERC, without whom the study could not have been conducted.

REFERENCES

[1] Rowe, E., Asbell-Clarke, J. & Baker, R. (2015). Serious game analytics to measure implicit science learning. In C.S. Loh, Y. Sheng, & D. Ifenthaler *Serious Game Analytics: Methodologies for Performance Measurement, Assessment, and Improvement.* Springer Science+Business..

[2] Rowe, E., Asbell-Clarke, J., Baker, R., Eagle, M., Hicks, A., Barnes, T., Brown, R., & Edwards, T., (2017). Assessing implicit science learning in digital games. *Computers in Human Behavior*. DOI: 10.1016/j.chb.2017.03.043

[3] Polanyi, M. (1966). *The Tacit Dimension*. London: Routledge. (University of Chicago Press. ISBN 978-0-226-67298-4. 2009 reprint).

[4] Arena, D. A., & Schwartz, D. L. (2013). Experience and explanation: Using videogames to prepare students for formal instruction in statistics. *Journal of Science Education and Technology*.

[5] Shute, V. J., Masduki, I., Donmez, O., Dennen, V., Kim, Y., Jeong, A. C., & Wang, C. (2010). Modeling, Assessing, and Supporting Key Competencies Within Game Environments In D. Ifenthaler, P. Pirnay-Dummer, & N. M. Seel (Eds.), *Computer-Based Diagnostics and Systematic Analysis of Knowledge* (pp. 281-309). New York: Springer-Verlag.

[6] Barr, V., & Stephenson, C. (2011). Bringing computational thinking to K-12: What is involved and what is the role of the computer science education community? *ACM Inroads, 2*(1), 48-54.

[7] Wing, J. M. (2011). Computational thinking. Paper presented at the VL/HCC.

[8] Google (2016). CT Overview. Retrieved from https://edu.google.com/resources/programs/exploring-computational-thinking/#!ct-overview.

[9] CSTA (Producer). (2017). CSTA K-12 Computer Science Standards. Retrieved from http://www.csteachers.org/?page=CSTA_Standards.

[10] Grover, S., & Pea, R. (2013). Computational Thinking in K–12 A Review of the State of the Field. *Educational Researcher, 42*(1), 38-43.

[11] Hoover, A., Barnes, J., Fatehi, B., Moreno-Leon, J., Puttick, G., Tucker-Raymond, E., & Harteveld, C., (2016, October). Assessing Computational Thinking in Students' Game Designs. Poster presented at CHI Play, Austin, TX.

[12] Grover, S., Bienkowski, M., Basu, S., Eagle, M., Diana, N., and Stamper, J. (2017, March). A framework for hypothesis-driven approaches to support data-driven learning analytics in measuring computational thinking in block-based programming. In *Proceedings of the Seventh International Learning Analytics & Knowledge Conference* (LAK '17). ACM, New York, NY, USA, 530-531. DOI: https://doi.org/10.1145/3027385.3029440

[13] Screenflow for Mac 6.0 [Computer software]. (2016). Nevada City, CA: Telestream. https://www.telestream.net/screenflow/overview.htm

[14] Haladyna, T. M., & Downing, S. M. (2004). Construct-irrelevant variance in high-stakes testing. *Educational Measurement: Issues and Practice, 23*(1), 17-27.