Labeling Implicit Computational Thinking in Pizza Pass Gameplay

Elizabeth Rowe
Jodi Asbell-Clarke
Educational Gaming Environments(EdGE) group @ TERC
Cambridge, MA
elizabeth_rowe@terc.edu
jodi_asbell-clarke@terc.edu

Ryan Baker
Penn Center for Learning Analytics
University of Pennsylvania
Philadelphia, PA
rbaker@upenn.edu

Santiago Gasca
Erin Bardar
Educational Gaming Environments (EdGE) group @ TERC
Cambridge, MA
santiago_gasca@terc.edu
erin_bardar@terc.edu

Richard Scruggs
Penn Center for Learning Analytics
University of Pennsylvania
Philadelphia, PA
rscruggs@upenn.edu

Abstract

Players can build implicit understanding of challenging scientific concepts when playing digital science learning games [7]. In this study, we examine implicit computational thinking (CT) skills of 72 upper elementary and middle school students and 10 computer scientists playing a game called *Pizza Pass*. We report on the process of creating automated detectors to identify four CT skills from gameplay: problem decomposition, pattern recognition, algorithmic thinking, and abstraction. This paper reports on hand-labeled playback data obtaining acceptable inter-rater reliability and 100 gameplay features distilled from digital log data. In future work, we will mine these features to automatically identify the CT skills previously labeled by humans. These automated detectors of CT will be used to analyze gameplay data at scale and provide actionable feedback to teachers in real-time.

Author Keywords

Implicit learning; computational thinking; learning games; data mining.

ACM Classification Keywords

H.5.1 Information Interfaces and Presentation: Multimedia Information Systems—*Evaluation/methodology*; K.3.2 Computers and Education: Computer and Information Science Education—*Computer science education*; K.8.0 Personal Computing: General—*Games*

Figure 1. Map of the Zoombinis' journey from Zoombini Isle to Zoombiniville.

Figure 2. Zoombinis must present pizzas and sundaes with certain toppings to appease trolls at Pizza Pass. (Source: [8])

Introduction

Zoombinis [11] is an award-winning, popular learning game that elicits computational thinking (CT). Players guide Zoombini characters on a journey through a series of challenging logic puzzles, leading them to safety in Zoombiniville (see Figure 1). The game includes a suite of 12 puzzles, each with four levels, and provides scaffolded problem-solving for learners ages 8 and above. *Zoombinis* puzzles were designed to develop the mathematics concepts necessary for computer programming and data analysis such as sets, logical relationships, dimensions, mappings, sorting, comparing, and algorithms [5]. Currently, we are studying gameplay among students in grades 3-8 in order to understand how students implicitly learn CT during *Zoombinis* gameplay and how their teachers can build upon that knowledge.

This paper reports on human-labeling of gameplay observations from 72 upper elementary and middle school students (CT novices) and 10 computer scientists (CT experts) playing the first level of one of the puzzles, *Pizza Pass* (see Figure 2). We then calculate over 100 features from the data. In future work, we will use data mining to determine which combination of these features can be used to replicate the human labels. The detectors developed by our work represent implicit *game-based learning assessments (GBLA)*, which may reveal knowledge that often goes unexpressed in typical assessments [7].

Implicit Computational Thinking

Learners may demonstrate knowledge through behaviors that they are not yet able to express formally [6, 11]. This is referred to as *implicit knowledge*. Game-based learning assessments (GBLA) show promise as a new method of assessing implicit knowledge by avoiding jargon, construct-irrelevant material, and test anxiety which can make traditional assessments challenging [10].

As described in [8], we defined a learning progression of CT and problem-solving skills in *Zoombinis* based on several definitions of CT emerging in the field [1-4, 12]. We used this learning progression to guide our labeling of strategies and behaviors in gameplay consistent with facets of the progression (Figure 3). Facets of this iterative learning progression of CT that we hypothesize will be evident in *Zoombinis* gameplay include:

1. **Problem Decomposition**: The reduction of the ambiguity or complexity of a problem by breaking it into smaller, more manageable parts. This is comparable to isolating variables or systems to test.

2. **Pattern Recognition**: The recognition that objects are arranged following a rule or rules. The identification of groups of solutions or characteristics of solutions that can be categorized.

3. **Abstraction:** The removal of details to identify and extract relevant information to define main idea(s) or solutions.

4. **Algorithm Design**: The creation of an ordered list of instructions for solving a problem or doing a task. The creation or explication of general solutions to a problem or family of problems.

In [8], we identified six specific, iterative phases of problem solving that are intertwined with expressions of CT in *Zoombinis* gameplay, four of which are evident in Pizza Pass (Figure 3):

Computational Thinking Learning Progression
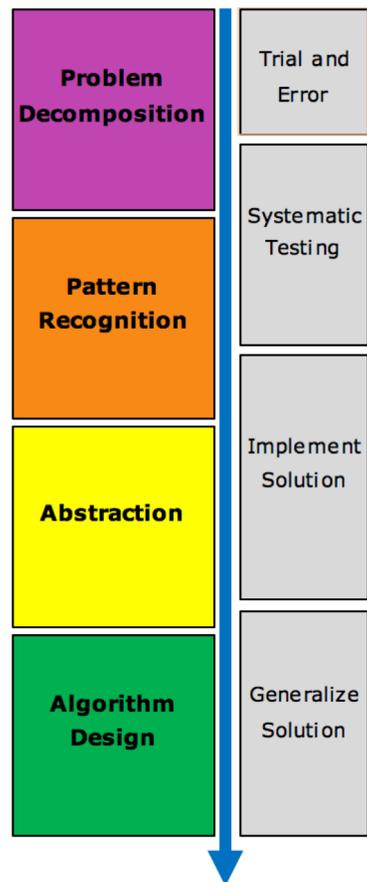
Figure 3. A learning progression of computational thinking operationalized in *Zoombinis* gameplay. (Source: [8])

1. **Trial & Error:** No evidence of testing hypotheses in an ordered, planned way. Actions are independent of prior actions or do not build off prior actions in a productive way.
2.   **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.
3. **Systematic Testing with a Partial Solution:** Testing hypotheses about a second dimension of the underlying rule when the first dimension is known.
4. **Implementing a Full Solution:** Completing the pattern once a working solution for all dimensions of the puzzle has been found.

These facets of CT are demonstrated in their progression from **Trial and Error**, where there is no productive systematic pattern to the behaviors, towards **Systematic Testing**, typically involving problem decomposition. When players recognize patterns in solutions to the smaller problems, they abstract to general rules so that they can then move to **Implementing a Solution** to the larger puzzle. When learners encounter new puzzles that require similar solutions, they may begin to **Generalize Solutions** leading toward demonstrating algorithm design.

Pizza Pass

As shown in Figure 2, 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. The player selects a combination of toppings via buttons on a machine, and a Zoombini delivers the meal to the troll(s). However, the troll(s) only say whether (a) they want more toppings, (b) don't like at least one of the toppings, or (c) the meal is perfect. The troll throws incomplete meals to the side of the path, while meals that all trolls reject are thrown into a pit. Once all trolls are satisfied, they (noisily) eat their pizzas and let the remaining Zoombinis through. The central question this research addresses is: ***How can we validly and reliably measure implicit computational thinking in* Pizza Pass *gameplay?***

Methods

Similar to our previous modeling of learner understanding of Newtonian mechanics in the physics game, *Impulse* [7], and scientific inquiry skill in simulations [9], we are building automated tools that can use gameplay data to provide information about players' implicit CT learning in *Pizza Pass* play:

1. Hand-label Zoombinis gameplay to capture the variety of strategies used to solve the puzzles.
2. Merge labels with gamelog data
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 on progress in the first three steps of this process, including reliability data from the labeling completed to date and the 100+ features engineered to model that human labeling.

Phase	First Attempt	Systematic Testing		Implementing a Full Solution
Strategy		One at a Time : Tries one topping at a time		
Problem Decomposition		Find solution more efficiently by isolating toppings		
Pattern Recognition				Keeps only liked toppings
Abstraction				Combines liked toppings

Figure 4: Labeling of implicit computational thinking in Pizza Pass

Sample and Context

Pizza Pass gameplay data have been collected from 36 upper-elementary children in grades 3-5 (17 females, 19 males), 36 middle school children in grades 6-8 (19 females, 17 males), and 10 computer scientists (4 females, 6 males). Playtesting sessions last approximately 1 hour, and involve gameplay across multiple *Zoombinis* puzzles. Child participants were recruited from local schools and clubs, as well as after-school programs. Adult participants were recruited through personal social networks.

Human Labeling

When developing the labeling system, we noticed that many players are focused solely on learning the game mechanic (i.e., how the game works) the first time they play the game making it difficult to discern any CT or problem-solving skills. For this reason, labeling begins at Round 2 in Level 1 for all players. All subsequent Level 1 play is labeled independently by the same two researchers with an inter-rater reliability check-in after every tenth player.

Results

Pizza Pass Labeling

Figure 4 is an idealized, annotated labeling of one round of Pizza Pass play in Level 1. In this round, the player is demonstrating what we label as a *One-at-a-time strategy*—the player tries one topping at a time and, after trying all toppings, combines only those the troll likes. This strategy provides evidence of systematic thinking, problem decomposition, pattern recognition, and abstraction. Evidence of algorithmic thinking in this case occurs when players repeat the same strategy across multiple rounds.

We also identified two other common strategies: additive and winnowing. In the *additive* strategy, players try each topping at a time and on subsequent deliveries retain only those toppings the troll likes at the

end. For the *winnowing* strategy, players try all toppings at once then remove one at a time. Human labelers also rate each round on the overall efficiency of the gameplay—*low* (a lot of trial and error), *moderate* (defined strategy with a few mistakes), *high* (defined strategy with no more than one mistake).

Table 1 presents inter-rater reliability results from these 82 players. A total of 288 rounds of gameplay with 2010 delivery events were labelled.

Label	Reliability
Phases of Problem Solving	
1. Trial and Error	0.87
2. Systematic Testing	0.82
3. Sys. Testing w/Partial Solution	0.90
4. Implementing a Full Solution	0.78
Computational Thinking	
1. Problem Decomposition	0.85
2. Pattern Recognition	0.90
3. Abstraction	0.76
Strategy	
1. One-at-a-Time	0.78
2. Additive	0.83
3. Winnowing	0.62
Gameplay Efficiency	0.95

Table 1: Reliability of Pizza Pass Labeling (Cohen's Kappas for Phases, CT, and Strategies; Cronbach's Alphas for Gameplay Efficiency)

For all labels except gameplay efficiency, Cohen's Kappas are reported to account for chance agreement. Gameplay efficiency is an ordinal rating, so a Cronbach's alpha is reported. These reliabilities provide an upper bound on the quality of the detectors. Therefore, only labels with reliabilities higher than 0.60 will be retained for the creation of automated detectors.

Feature Engineering
Informed by the experience developed through creating and implementing the hand labeling scheme, we calculated 100+ features from the raw data logs that represent potentially meaningful evidence of implicit computational thinking. These features fall into six main categories. Sample features in each category include:

1. **Overall Gameplay**: Duration of play; Number of rounds played; Percentage of Zoombinis exiting level at the end of each round;
2. **Topping Duplicates**: Number of repeated topping combinations delivered; number of consecutive repeated topping combinations;
3. **Topping Futzes**: Number of topping changes before delivery; Relative time between changes (speeding up/slowing down);
4. **Topping Selection**: Number of deliveries in this round; Number of deliveries with one topping change since last delivery; Number of unique topping combinations tried;
5. **Timing**: Number of topping selections before any feedback has been given; Number of topping selections before all feedback has been given; Average time between deliveries;
6. **Troll Satisfaction**: Number of deliveries rejected by all trolls; Number of deliveries rejected by one troll; Number of consecutive rejections by one troll;

The next step will be to use these features to model the human labeling through the creation of automated detectors.

CONCLUSION

In this paper, we outlined a process for creating automated assessments of implicit computational thinking from gameplay behaviors. Reliability of the human labeling and sample distilled features were also discussed. This work sets a model for implicit GBLA that can be used to reveal knowledge and skills through activity rather than relying on what learners can express on typical assessments. This work may ultimately inform how researchers and educators can assess learning from a more cognitively diverse set of learners, revealing and unleashing otherwise untapped everyday knowledge.

ACKNOWLEDGEMENTS

References

1. Valerie Barr and Chris Stephenson. 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 (February 2011), 48-54.
2. CSTA. 2017. CSTA K-12 Computer Science Standards. Retrieved from http://www.csteachers.org/?page=CSTA_Standards.
3. Google. 2016. CT Overview. Retrieved from https://edu.google.com/resources/programs/exploring-computational-thinking/#!ct-overview.
4. Shuchi Grover and Roy Pea. 2013. Computational Thinking in K–12: A Review of the State of the Field. *Educational Researcher* 42, 1, 38-43.
5. Chris Hancock and Scot Osterweil. 1996. Zoombinis and the Art of Mathematical Play. *Hands On!* 19, 1 (Spring 1996), 1,17-19.
6. Michael Polanyi. 1966. *The Tacit Dimension*. University of Chicago Press, Chicago, IL.
7. Elizabeth Rowe, Jodi Asbell-Clarke, Ryan Baker, Michael Eagle, Andrew Hicks, Tiffany Barnes, Rebecca Brown & Teon Edwards (2017). Assessing implicit science learning in digital games. *Computers in Human Behavior, 76*, 617-630. DOI: 10.1016/j.chb.2017.03.043
8. Elizabeth Rowe, Jodi Asbell-Clarke Kathryn Cunningham & Santiago Gasca (2017, October). Assessing implicit computational thinking in Zoombinis gameplay: Pizza Pass, Fleens, and Bubblewonder Abyss. Proccedings of the ACM SIGCHI Annual Symposium on Computer-Human Interaction in Play, Amsterdam
9. Michael Sao Pedro, Ryan Baker, Janice Gobert, Orlando Montalvo, & Adam Nakama (2013). Leveraging Machine-Learned Detectors of Systematic Inquiry Behavior to Estimate and Predict Transfer of Inquiry Skill. User Modeling and User-Adapted Interaction, 23 (1), 1-39.
10. Valerie J. Shute, Iskandaria Masduki, Oktay Donmez, Vanessa P. Dennen, Yoon-Jeon Kim, Allan C. Jeong, and Chen-Yen Wang. 2010. Modeling, Assessing, and Supporting Key Competencies Within Game Environments. In *Computer-Based Diagnostics and Systematic Analysis of Knowledge*. D. Ifenthaler, P. Pirnay-Dummer, & N. M. Seel (Eds.). Springer US, Boston, MA, 281-309.
11. Sternberg, R. J. (1996). *Successful Intelligence: How Practical and Creative Intelligence Determine Success in Life.*
12. TERC. 2015. *Zoombinis*. Game [Android, iOS, MacOS, Windows, Web]. (7 August 2015). TERC, Cambridge, MA.
13. Jeannette M. Wing. 2006. Computational thinking. Commun. ACM 49, 3 (March 2006), 33-35.