Modeling Implicit Computational Thinking in *Zoombinis* Mudball Wall Puzzle Gameplay

Authors:
Ma. Victoria Almeda, Ph.D., EdGE at TERC
Elizabeth Rowe, Ph.D., EdGE at TERC
Jodi Asbell-Clarke, Ed.D., EdGE at TERC
Richard Scruggs, M.Ed., University of Pennsylvania
Ryan Baker, Ph.D., University of Pennsylvania
Erin Bardar, Ph.D., EdGE at TERC
Santiago Gasca, M.A., EdGE at TERC

ABSTRACT

Introduction

There is increasing evidence that games can play a significant role in promoting STEM learning for children and young adults (Authors, 2017). Well-designed games provide the opportunity to support the development and measurement of implicit learning, which is argued to be foundational for explicit learning in the classroom (Polyani, 1966). Computational thinking (CT) is important to the development of 21st century skills (Voogt et al., 2015), and may also support the inclusion of learners with cognitive differences. Previous studies have shown a potential link between cognitive differences and computational thinking, with students with autism excelling at pattern recognition (Dawson et al., 2007) and learners with ADHD demonstrating superior performance on idea generativity than adolescents without ADHD (Abraham et al., 2006). As such, assessing implicit CT through gameplay is not only valuable in its own right, but can also help educators broaden and deepen STEM experiences for diverse learners.

Learning analytics methods offer unique opportunities for providing scalable, replicable measures of implicit learning in games (Authors, 2017). In this study, we used learning analytics techniques to build automated detectors able to recognize students' implicit Computational Thinking (CT) skills and strategies from gameplay log data. Success in accurately detecting these skills and strategies represents a new way of providing formative, real-time game-based assessments of CT.

Zoombinis (Authors, 2015) is an award-winning, popular educational game that elicits implicit computational thinking. We study CT within Mudball Wall, where a large gridded wall blocks the Zoombinis' progress (Figure 1). Players must use computational thinking skills and strategies to discover the correct patterns of shapes & colors of mudballs to hit dotted cells on the wall and catapult their Zoombinis onward.

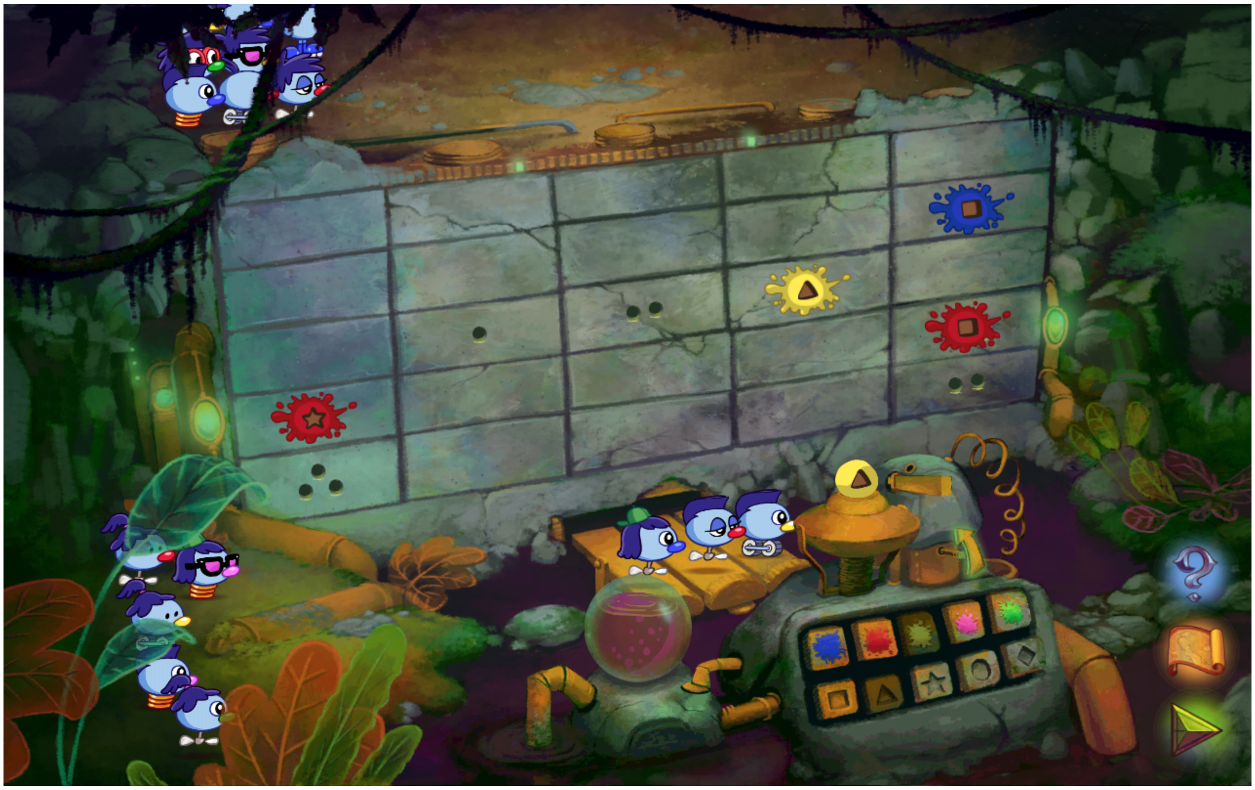
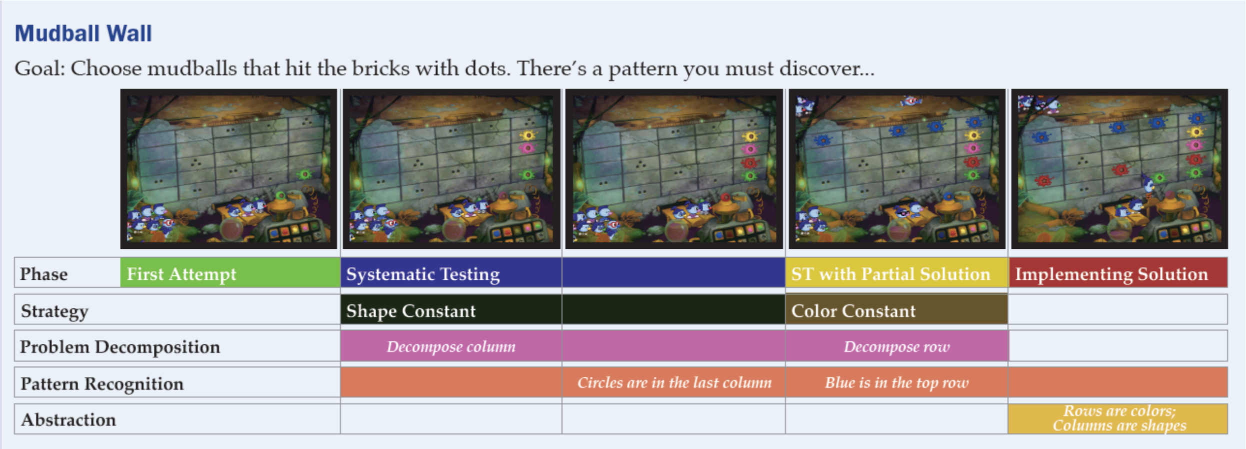


Figure 1: Zoombinis screenshot of Mudball Wall

Methods

Our process of modeling implicit computational thinking includes five steps: 1) Reliable hand-labeling of CT behaviors within *Zoombinis* gameplay, 2.) Synchronizing labels to gameplay process data, 3) Distilling gameplay process data into features useful for measuring behaviors consistent with CT, 4) Building detectors of players' CT strategies based on human labeling, 5) Validating the detectors as formative assessments of implicit CT by comparing to performance on external pre/post assessments.

In step 1, researchers watched *Zoombinis* gameplay independently and refined label definitions of implicit computational thinking, until they agreed that the labeling system was an exhaustive representation of all the CT-related gameplay behaviors seen. Next, two researchers independently labeled all rounds of Level 1 play from 74 students (37 upper elementary and 37 middle school), achieving acceptable interrater reliability (Cohen's kappa = 0.733 to 0.940 across CT skills and strategies). Figure 2 shows sample labeled data.


Mudball Wall

Goal: Choose mudballs that hit the bricks with dots. There's a pattern you must discover...

Phase	First Attempt	Systematic Testing		ST with Partial Solution	Implementing Solution
Strategy		Shape Constant		Color Constant	
Problem Decomposition		*Decompose column*		*Decompose row*	
Pattern Recognition			*Circles are in the last column*	*Blue is in the top row*	
Abstraction					*Rows are colors;* *Columns are shapes*

Figure 2: Sample annotated Mudball Wall labeling

We then built separate detectors for each human-applied label of implicit CT, trying four common classification algorithms used to predict categorical variables from a set of features: W-J48, W-JRIP, step regression, and Naive Bayes. The goal of building these detectors is to replace the hand labeling of CT labels with an automated model that can be applied directly to data. Detectors were built using 4-fold student-level batch cross-validation, a process in which models are repeatedly built on 75% of the students and tested on the remaining 25%, to estimate model generalizability to new data. These detectors were evaluated using two goodness metrics: Cohen's Kappa and AUC ROC.

Results and Discussion

Table 1 presents the best performing algorithm for each CT label in Mudball Wall. While the 2D Pattern Completer and Alternating Color and Shape strategies obtained relatively low kappa values, all detectors achieved moderately high to excellent AUC values, in the range of 0.702 to 0.976. Specifically, the Implementing a Full Solution and Abstraction detectors both achieved an AUC value of 0.976, meaning that the detectors can correctly distinguish between the absence and presence of these CT skills 97.6% of the time. This is a level of quality higher than seen in many medical applications, indicating that these detectors can be used for formative assessment of students' implicit computational thinking and for informing teachers which students need learning support.

Table 1: Models for Mudball Wall; only best model shown for each CT construct

Label	Definition	Algorithm	Kappa	AUC
Phases of Problem Solving				
1. Trial and Error	No evidence of testing hypotheses in an ordered, planned way.	Naive Bayes	0.262	0.811
2. Systematic Testing	Testing hypotheses about an underlying rule in an ordered, planned way with the end goal of finding a working solution to implement	Step Reg	0.775	0.947
3. Sys. Testing w/ Partial Solution	Testing hypotheses about a second dimension of the underlying rule when the first dimension is known.	Step Reg	0.589	0.900
4. Implementing a Full solution	Completing the pattern once a working solution for all dimensions of the puzzle has been found.	Step Reg	0.793	0.976
Computational Thinking				
1.Problem Decomposition	The reduction of the ambiguity or complexity of a problem by breaking it into smaller, more manageable parts.			
Explicit	Player chooses mudballs in color/shape combinations that visibly establishes an answer for one dimension (row OR column) at a time.	Step Reg	0.367	0.757
Implicit	Player chooses mudballs in color/shape combinations that reveals information about both dimensions simultaneously (row AND column) to more quickly establish the overall grid rules	Step Reg	0.305	0.758
2. Pattern Recognition	Player makes use of knowledge that all values of an attribute (e.g., all red shapes or all circles) are in a single column/row.	Step Reg	0.712	0.927

3. Abstraction	Player generalizes across values (colors or shapes) to identify and extract the rules of the puzzle. (e.g., player understands rows are colors, columns are shapes, and begins targeting dots to complete the puzzle).	Step Reg	0.797	0.976
Strategy				
1.Color or Shape Constant	Player holds one attribute (color or shape) constant, while systematically testing values of the other attribute to establish the rule of a row or column.	Naive Bayes	0.391	0.756
2. 2D Pattern Completer	Special case of Color or Shape Constant strategy in which a player completes an ENTIRE row AND an ENTIRE column to establish the full grid pattern before moving on to implementation.	Naive Bayes	0.105	0.702
3. Maximizing Dots	Player appears to actively target dots on the grid using information available to them from previous moves.	Step Reg	0.665	0.915
4. Try All Combinations of Color & Shape	Player tries all shapes/color pairs, changing both attributes between moves so as not to repeat a shape or color.	Step Reg	0.665	0.915
5. Alternating Color and Shape	Player systematically alternates between holding color (e.g., red) and shape (e.g., circle) constant, to establish the rule of a row or column.	Naive Bayes	0.258	0.825
Gameplay Efficiency	How well the learner appears to understand the game mechanic and applies an effective strategy across an entire round of play.	Naive Bayes	0.535	0.823
Learning Game Mechanic	Indicates a lack of understanding of the game mechanic (e.g., repeating identical mudballs)	Step Reg	0.492	0.869

Acting Inconsistent with Evidence	From the expert perspective, the move contradicts evidence available from prior moves.	Step Reg	0.539	0.894

Detecting implicit computational thinking has potential implications for providing an innovative form of assessment for diverse learners who often face challenges in demonstrating their knowledge in the context of traditional assessments. Diverse learners may be more able to demonstrate what they know when learning occurs in a well-designed game, where they are able to bypass potential barriers of traditional assessments and test solutions across puzzles that do not rely on word problems. This work, therefore, represents a step towards providing implicit STEM learning assessments using learning analytics techniques toward a more effective and inclusive STEM education. The final step of our process, not yet complete, will be to apply these detectors to a broader sample and validate them against external pre-post assessments of computational thinking collected as part of a national *Zoombinis* implementation study.

BRIEF SUMMARY

In this paper, we examine implicit computational thinking (CT) skills and strategies of 74 upper elementary and middle school students in the context of the Mudball Wall puzzle in the educational game, *Zoombinis*. By leveraging gameplay log data and building automated detectors of students' implicit CT skills and strategies, we develop measures that can be used to analyze gameplay behaviors at scale and providing actionable information for STEM instruction.

Acknowledgements

We are grateful for grant NSF/EHR/DRK12 grant #1502282. We are thankful for the study participants and the many contributions of our wonderful EdGE colleagues without whom the study could not have been conducted.

References:

Abraham, A., Windmann, S., Siefen, R., Daum, I., & Güntürkün, O. (2006). Creative thinking in adolescents with attention deficit hyperactivity disorder (ADHD). *Child Neuropsychology*, *12*(2), 111-123.

Authors. (2017). Assessing implicit science learning in digital games. *Computers in Human Behavior*, *76*, 617-630.

Authors. 2015. Zoombinis. Game [Android, IOS, MacOS, Windows, Web]. (5 March 2019). TERC, Cambridge, MA.

Dawson, M., Soulières, I., Ann Gernsbacher, M., & Mottron, L. (2007). The level and nature of autistic intelligence. *Psychological science*, *18*(8), 657-662.

Polyani, M. (1966). Chapter 2: Emergence. *The Tacit Dimension*, 29-52.

Voogt, J., Fisser, P., Good, J., Mishra, P., & Yadav, A. (2015). Computational thinking in compulsory education: Towards an agenda for research and practice. *Education and Information Technologies*, *20*(4), 715-728.