

Real Time Assessment of Computational Thinking

Kyu Han Koh, Ashok Basawapatna*, Hilarie Nickerson, Alexander Repenning*

Department of Computer Science

University of Colorado Boulder

*AgentSheets Inc.

Boulder, USA

kohkh@colorado.edu, ashok@agentsheets.com, hnickerson@colorado.edu, alex@agentsheets.com

Abstract

This paper suggests a Cyberlearning tool based on a highly innovative assessment methodology that helps teachers with computer science education. Currently, there is a strong push to integrate aspects of programming and coding into the classroom environment. However, few if any tools exist that enable real-time formative assessment of in-class programming tasks. The proposed REACT (Real Time Evaluation and Assessment of Computational Thinking) system is a first step toward allowing teachers to see which high-level concepts students have mastered and which ones they are struggling with as students code in real time. REACT supports and facilitates the teaching of 21st century computing skills such as computational thinking [1] in the classroom environment.

Keywords— *Computational Thinking; Online Assessment; Real Time Assessment; Computational Thinking Pattern; Computational Thinking Pattern Analysis*

I. INTRODUCTION

The problem of making programming both accessible and exciting, which leads to a gap between supply and demand for computer scientists, has its roots in early schooling and is international in scope [2]. The Scalable Game Design (SGD) project [3] has the ambitious goal of revolutionizing computer science education in public schools through a combination of game design and science, technology, engineering, and math (STEM) simulations integrated into the middle-school curriculum. More than 10,000 students from inner-city, remote rural, and Native American schools have participated in SGD, making it one of the US's largest middle school computer science education studies. The SGD results shows that the SGD approach works and broadens participation [3]—even in some of the toughest, poorest, and most isolated schools in the nation. Seventy-four percent of male participants and sixty-four percent of female participants wanted to continue with similar courses as electives. SGD students create playable games based on sophisticated concepts that include advanced mathematics and artificial intelligence.

We designed and implemented a real time online assessment system for the SGD project teachers. In this paper, we describe a system entitled REACT (Real Time Evaluation and Assessment of Computational Thinking) as a first step toward allowing teachers to see which high-level computational thinking concepts students have mastered and which ones they are struggling with as students code in real time. To this end the REACT system displays which

Computational Thinking Patterns students are currently implementing, which patterns they have yet to implement, and the correctness of previously implemented patterns [4]. REACT supports and facilitates the teaching of 21st century computing skills such as computational thinking [1] in the classroom environment.

II. BACKGROUND

Real Time assessment systems allow teachers to gain insight into the level of understanding of individual students and their class as a whole at any given point in time, offering more rapid and comprehensive information access for both teachers and students as compared to typical assessment and feedback methods. The use of student response systems, often referred to as clickers, represents one common approach to gathering such information within the classroom [5]. Typical clicker systems allow students to individually submit their answers to multiple choice questions provided by the instructor for a variety of purposes, including formative assessment and low-stakes quizzing [6]. Additional uses include managing interaction, guiding thinking, conducting experiments, and increasing engagement. More sophisticated systems, such as InkSurvey [7], allow free-form text and graphical input. Depending on the purpose, instructors may choose to receive feedback anonymously from student response systems or to match responses with individuals. Practices for teaching computer science using such systems have not differed appreciably from those employed in other disciplines, though more attention has perhaps been paid to the potential for real-time analysis and longitudinal collection of data [8, 9]

A. Teaching and Learning Based on the Zones of Proximal Flow and Computational Thinking Pattern Analysis

The Zones of Proximal Flow framework [10] describes what we call a “Project First, just-in-time principles” pedagogy that we have found to be highly effective. The right-hand side of Figure 1 shows the Zones of Proximal Flow diagram—a combination of Csíkszentmihályi’s Flow[11] with Vygotsky’s Zone of Proximal Development [12] conceptualization. Project First leads students through the Zone of Proximal Development (ZPD), which, according to Vygotsky, is an ideal zone for learning because it pushes learners to their threshold of knowledge. With the right external support from the teacher, students can overcome this threshold and learn advanced topics such as programming efficiently. The resulting framework as applied to Scalable Game Design includes a curriculum of

increasingly advanced game design activities that range from basic 1980s arcade games such as Frogger (bottom-left in Figure 1) to more contemporary games such as The Sims. As students progress, they encounter sophisticated concepts such as artificial intelligence (top left in Figure 1).

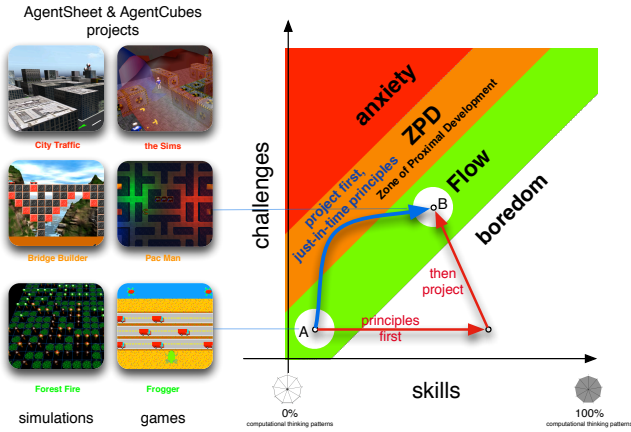


Figure 1. Zone Of Proximal Flow wherein ZPD is located in between regions of Flow and anxiety

In the Zones of Proximal Flow diagram, the vertical axis represents the level of the design challenge that would be intrinsic to a certain game or STEM simulation. The horizontal axis represents students’ computational thinking skills as measured by Computational Thinking Pattern Analysis (CTPA) [4]. CTPA is not looking for constructs such as IF and LOOP statements at the programming level. It looks instead for more semantic object interactions, called Computational Thinking Patterns (CTP) [13], such as collisions and diffusion at a phenomenalistic [14] level. In the ZPF diagram CTPA captures a single aggregate value between 0% CTP coverage, i.e., a student not exposed to any of the patterns in the inventory, and 100% Computational Thinking Pattern coverage, i.e., a student exposed to all CTP—presumably through building a sequence of projects. In summary, CTPA can assess the state a student is in. Now the question becomes one of how assessment based on CTPA can be used to establish a real-time interaction between teacher and students.

III. REAL-TIME STEM PROGRAMMING FORMATIVE ASSESSMENT TOOLS

We have built the REACT system as an embedded, formative, real-time graphical assessment tool that quickly gives teachers insight into student mastery of computational thinking constructs as they are creating games and simulations. This ability for formative assessment is the first step towards objectively determining which learning state students are engaged in as well as which computational thinking and STEM topics students understand and/or find challenging.

REACT has three technical objectives as the followings:

- Build a web-based system running on desktop and tablet computers. Embedded assessment will be employed to serve as formative assessment tool helping teachers to effectively support computer science education.

- Communicate students’ progress information to teachers hierarchically, allowing teachers to quickly get a high level sense of the entire class but also enabling them to gradually explore individual student progress.
- Provide teachers the most useful representations of class/individual progress allowing them to make effective instructional decisions.

A. REACT vs. Other Online Assessment Tools

The emphasis on computational thinking makes our system unique in the realm of real-time in-class assessment tools. There are many attempts at real-time assessment that focus both on end-user programming tools and computational science in general. Table 1 lists a subsection of these attempts comparing our system to two other end-user programming assessment systems.

Table 1. Comparing the REACT system to other end-user programming assessment tools

Assessment system name	REACT	Hairball [15] & Scrape [16]
Real-Time with active alarm vs. Post performance	Real Time	Post
Programming vs. Computational Thinking Based Pattern Assessment (CTPA)	CTPA	Programming
Individualized formative assessment in real-time	Yes	No
Main target audience	Teachers	Researchers

This table summarizes different coding based assessment tools in terms of four characteristics. These include a real-time system for alerting the teacher if students are making mistakes, explicitly displaying Computational Thinking Pattern based assessment versus lower-level programming construct assessment, the ability to formatively assess any student in real time, and the main target audience of the system. In all of these aspects REACT better informs teachers as to the state of student projects.

Other coding-based assessment tools focus on summative assessment, that is, assessing the final artifact produced by the programming task. In contrast, REACT focuses on formative assessment to assess students as they are coding. Since REACT aims to enable teachers to see the state of student programs quickly, it is necessary for REACT to integrate real time embedded assessment with a quickly readable graphical display that alerts the teacher to students who may need the most help, or to tasks the class as a whole might not understand. The combination of formative assessment capability and its real-time use in the classroom makes REACT unique and innovative.

The formative assessment capabilities of REACT are important for the following reasons:

- REACT enables teachers to better understand student mastery of specific topics in real time.
- REACT gives teachers an initial way to see if their students are actively engaged in the coding endeavor (i.e. are the students in Flow or in the ZPF region, or is the task too challenging or boring?).
- REACT gives teachers the ability to perceive potential in-class problems before they occur. This lets teachers preemptively focus on adding scaffolding or challenges before students develop the notion that programming is “hard and boring.”

B. REACT Data Analysis

The REACT system is embedded into the online publicly available Scalable Game Design Arcade [4, 13] enabling real time data mining of student projects as they are programmed during class. The REACT system breaks down all collectable student project information and records it in the REACT database. REACT analyzes the student project information stored in this database through Computational Thinking Pattern Analysis in real time. This analysis extracts semantic meaning out of the code by interpreting which Computational Thinking Patterns have been implemented by students. The analyzed data are illustrated through different levels of visualization; Computational Thinking Pattern Analysis Graph (Figure 2), Computational Thinking Pattern Analysis Forensics (Figure 3), and Assessment Dashboard (Figure 4).

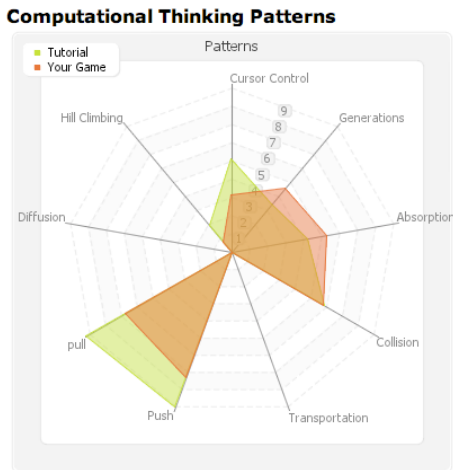


Figure 2. CTPA Graph illustrates analyzed student skills and learning at the semantic level



Figure 3. The Computational Thinking Pattern Analysis Forensics graph explains how a student has progressed his/her

computational thinking pattern implementations by programming with AgentSheets (or AgentCubes)



Figure 4. Example REACT Assessment Dashboard showing every student's performance in a given classroom.

The Assessment Dashboard indicates to teachers where students are in their programming tasks. The dashboard visualizes the programming progression for each student in the class through CTPA [4]. In Figure 4, green indicates students who are completing the program correctly, orange indicates students who may need some help with their program, and red indicates students who are in significant need of scaffolding. The Dashboard clearly shows students that might be in trouble. By selecting a specific student in the Dashboard, a teacher can see in-depth representations of that student's progression in their Computational Thinking Pattern Analysis Graph [4] and Computational Thinking Pattern Analysis Forensics.

IV. PRELIMINARY RESULTS

REACT testing took place over four weeks with four teachers, six classes, 23 hours of class time, and 134 student projects. All students were 6th graders with, according to each teacher, little to no prior programming experience.

Findings of this proposed research study indicate an overwhelmingly positive reaction from teachers using REACT, with every teacher in the study planning to independently continue using REACT in future game programming assignments.

For example, one teacher felt that REACT helped identify struggling students more effectively stating:

“As everybody was starting the game I feel they benefitted because I could see how they were starting. As they were progressing I would say it was more attributal to those students who were struggling—not beginners but I would call them challenged learners—my low end learners who are struggling, those students who are struggling and having a

hard time I feel like it's better to give those students intervention”

Furthermore, another teacher, comparing the current REACT to a non-REACT class stated the following:

“We got more done in this class and we asked them to do more. We approached it the same way. In the other (non-REACT) class I couldn't tell where they were I had to rely on the helpers. With REACT I was able to say that person is not up to speed and send someone over there and I couldn't do that this morning (non-REACT class) so that was interesting... I'm a lot more positive about it now than I was just looking at it from before we had classes. I see the power in it and I see very useful things particularly for a beginning class with a beginning project.”

More in-depth research must be done; however, these teacher testimonials begin to indicate the power of a REACT enabled classroom for helping teachers to support students in their computer programming project based learning activities.

The following table lists the general realized anticipated benefits and the un-anticipated collateral benefits indicated in this REACT study.

Anticipated Benefits	Collateral Benefits
REACT enables formative assessment of game design projects	Teacher summative assessment of student game design projects using the REACT tool
REACT can be use by the teacher for effective in-class management through intervention	REACT can lead student self-assessment and peer interaction, and teacher/student 2 way validation

V. CONCLUSION

Our work on REACT thus far begins to show its power to aid teachers in their real-time assessment goals related to computational thinking skills. However, much more field research must be completed in order to evaluate and validate the current REACT approach and strategy. For example, what presentation of the REACT data is most and least useful to teachers interpreting student progressions and learning achievement? What should be modified in subsequent incarnations of the REACT system?

In our previous research [10], we discussed our evidence for the existence of the Zones of Proximal Flow. REACT begins to implement an early proof of concept strategy to illustrate student skills and challenges that progress over time through the Scalable Game Design curriculum. Still the question of how to set the thresholds for the anxiety and boredom zones remains. As data is collected from thousands more students from numerous classrooms and districts across the United States, it is our hope that the REACT system can help make more explicit the thresholds for these ZPF regions.

VI. ACKNOWLEDGMENTS

This work is supported by the National Science Foundation under Grant Numbers DLR-0833612, IIP-1345523, and IIP-0848962. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

REFERENCES

- [1] Wing, J. M. Computational Thinking. *Communications of the ACM*, 49, 3 (2006), 33-35.
- [2] Vasagar, J. Michael Gove, to scrap 'boring' IT lessons. *the Guardian News*, <http://www.guardian.co.uk/politics/2012/jan/11/michael-gove-boring-it-lessons>, 2012.
- [3] Repenning, A. and Ioannidou, A. Broadening Participation through Scalable Game Design. In *Proceedings of the ACM Special Interest Group on Computer Science Education Conference, (SIGCSE 2008)* (Portland, Oregon USA, 2008). ACM Press, 305-309.
- [4] Koh, K. H., Basawapatna, A., Bennett, V., *et al.*, Towards the Automatic Recognition of Computational Thinking, *IEEE International Symposium on Visual Languages and Human-Centric Computing 2010*, Leganés-Madrid, Spain, September 21-25, 2010
- [5] Trees, A. R., and Jackson, M. H. The learning environment in clicker classrooms: Student processes of learning and involvement in large university level courses using student response systems. *Learning, Media and Technology*, 32, 1 (2007), 21-40.
- [6] Caldwell, J. E. Clickers in the large classroom: Current research and best-practice tips. *CBE-Life Sciences Education* 6 (2007), 9-20.
- [7] Gardner, T. Q., Kowalski, S. E., and Kowalski, F. V. Interactive simulations coupled with real-time formative assessment to enhance student learning. In *Proc. 2012 American Society for Engineering Education Conference and Exposition*, ASEE (2012).
- [8] Carter, P. An experience report: On the use of multimedia pre-instruction and just-in-time teaching in a CS1 course. In *Proc. 43rd ACM Technical Symposium on Computer Science Education*, ACM Press (2012), 361-366.
- [9] Teel, S., Schweitzer, D., and Fulton, S. Braingame: A web-based student response system. *Journal of Computing Sciences in Colleges*, 28, 2 (2012), 40-47.
- [10] Basawapatna, A., Repenning, A., Koh, K. H., Nickerson, H., The Zones of Proximal Flow: Guiding Students through a Space of Computational Thinking Skills and Challenges, *ICER '13: International Computing Education Research Conference*, August 12-14, San Diego, California, USA.
- [11] Csikszentmihalyi, M. *Flow: The Psychology of Optimal Experience*. Harper Collins Publishers, New York, 1990.
- [12] Vygotsky, L. S. and Kozulin, A. *Thought and Language*. The MIT Press, 1996.
- [13] Ioannidou, A., Bennett, V., Koh, K. H., *et al.* Computational Thinking Patterns. In *Proceedings of the Annual Meeting of the American Educational Research Association (AERA 2011)* (New Orleans, LA, April 8-12, 2011)
- [14] Michotte, A. *The Perception of Causality*, Methuen & Co. Ltd., London, 1963.
- [15] Boe, B., Hill, C., Len, M., *et al.* Hairball: lint-inspired static analysis of scratch projects. In *Proceedings of the Proceeding of the 44th ACM technical symposium on Computer science education* (Denver, Colorado, USA, 2013), ACM, 215-220.
- [16] Brennan, K. and Resnick, M. *New frameworks for studying and assessing the development of computational thinking*. City, 2012.