

---

# Assessing Computational Thinking in Students' Game Designs

**Amy K. Hoover**

Northeastern University  
Boston, MA 02115, USA  
amy.hoover@gmail.com

**Jackie Barnes**

Northeastern University  
Boston, MA 02115, USA  
jacqbarn@gmail.com

**Borna Fatehi**

Northeastern University  
Boston, MA 02115, USA  
borna.f@gmail.com

**Jesús Moreno-León**

Universidad Rey Juan Carlos &  
Programamos  
Madrid, Spain  
jesus.moreno@programamos.es

**Gillian Puttick**

TERC  
Cambridge, MA 02140  
g.puttick@terc.edu

**Eli Tucker-Raymond**

TERC  
Cambridge, MA 02140  
eli\_tucker-raymond@terc.edu

**Casper Hartevelde**

Northeastern University  
Boston, MA 02115, USA  
c.hartevelde@neu.edu

**Abstract**

Designing games requires a complex sequence of planning and executing actions. This paper suggests that game design requires computational thinking, and discusses two methods for analyzing computational thinking in games designed by students in the visual programming language Scratch. We present how these two analyses produce different narratives of computational thinking for our case studies, and reflect on how we plan to move forward with our larger analysis.

**Author Keywords**

Climate change; game design; computational thinking; student agency.

**ACM Classification Keywords**

H.5.m [Information interfaces and presentation (Algorithms, Design, HCI)]: Miscellaneous; I.6.8. [Simulation and Modeling]: Types of Simulation—*Gaming*; J.2. [Computer Applications]: Physical Sciences and Engineering—*Earth and atmospheric sciences*

**Introduction**

While developing computational thinking (CT) skills is essential for achieving computational literacy [11, 2], choosing what and how to assess it is a significant challenge. Perhaps partly because designing games requires designers to

---

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the Owner/Author. Copyright is held by the owner/author(s). *CHI PLAY'16 Extended Abstracts*, October 16-19, 2016, Austin, TX, USA ACM 978-1-4503-4458-6/16/10. <http://dx.doi.org/10.1145/2968120.2987750>

## CT Metrics in Dr. Scratch

### A. Abstraction and problem decomposition:

1. More than one script and more than one sprite
2. Definition of blocks
3. Use of clones

### B. Parallelism:

1. Two scripts on green flag
2. Two scripts on key pressed and a sprite
3. Two scripts on receive message, create clone, on backdrop change

### C. Logical Thinking:

1. If
2. If else
3. Logic operations

### D. Synchronization:

1. Wait
2. Broadcast, when I receive message, stop all, stop program, stop programs sprite
3. Wait until, when backdrop change to, broadcast and wait

model complex situations and to algorithmically plan, represent, and iterate ideas over time, recent studies suggest a link between designing games and increased scores on CT tests [12, 8, 9, 1]. In addition to designing games, we believe that automated assessments where both students and their instructors may quickly access feedback can potentially encourage CT development. One such assessment tool investigated in this paper is called Dr. Scratch (<http://drscratch.org>) [6]. Based on the seven metrics described in the first and second sidebars, it aims to quantify the CT expressed in programs developed in a visual programming language called Scratch (<http://scratch.mit.edu>) [10]. However, one question is whether those metrics are indeed appropriate for assessing CT. The purpose of this work-in-progress is twofold. One aim is to generate a conversation about how different analytic methods evaluate CT. The other is to think about how measures often coded through a qualitative rubric might be automated for more immediate analysis, and how such an analysis can generate instant feedback to designers while designing.

To explore this topic of comparing different analytic methods to evaluate CT in the context of game design, and with the aim of automating assessment, this paper analyzes games designed by five middle-school-aged girls who took part in a four-day workshop focused on designing games about climate change with Scratch. As CT is about applying computation to solve problems [11], we aimed to explore what kind of CT skills these girls expressed in representing climate change through a game, and assumed that a more complex representation meant higher CT because higher code complexity has been shown to correlate with higher levels of CT assessment [7]. We used two methods for assessing students' game designs: the aforementioned Dr. Scratch and a qualitative analysis. Specifically, the questions we asked in assessing the student games are:

- How do student designers make game design choices to represent climate change, given the tools available in Scratch and their previous Scratch experience?
- How can we measure CT from students' design choices, and how is this measure of CT reflected in the end product?

## Methods

To explore computational thinking in games designed by students, a pilot study was conducted during a four-day intensive game design and climate change workshop for five middle school girls, henceforth referred to as designers and by their pseudonyms: Tina, Rianna, Linda, Elina, and Cyn-dia. Researchers conducting the workshop facilitated student learning by leading discussions on climate change and providing support for game design and the design tools. The designers were introduced to the visual programming language Scratch (<http://scratch.mit.edu>) [10] and discussed climate change on the first day. For the 2nd, 3rd, and 4th days, each designer independently designed her game, but user tested in pairs. Designers concluded the workshop by presenting their final games and design processes.

Each designer completed a survey of her environmental attitudes and demographic information, and her level of Scratch experience. Audio of the designers' final presentation and interview data were also collected. Game files were saved at the end of each day.

Each game is analyzed through two modes: 1) CT scores computed by Dr. Scratch, and 2) a qualitative analysis of design choices. For the first mode of analysis with Dr. Scratch, each game file is scored based on seven different metrics designed to score and reflect a spectrum of CT shown in the first and second sidebars. For each metric,

### CT Metrics in Dr. Scratch

#### E. Flow Control:

1. Sequence of blocks
2. Repeat, forever
3. Repeat until

#### F. User Interactivity:

1. Green flag clicked, ask and wait, mouse blocks
3. Video, audio input

#### G. Data Representation:

1. Modify sprite properties
2. Operate on variables
3. Operate on lists

games can receive scores between 0 and 3, where 0 indicates a poor performance on a particular metric and 3 an excellent performance. The maximum CT score is 21.

The second mode of analysis is qualitative and elaborates on how designers' final choices relate to their planning processes, explanations of their games, and CT scores from Dr. Scratch. Based on the serious game design framework proposed by Harteveld [3], the design choices made by the workshop participants are examined with respect to three considerations: player interactions, mitigation topic, and the metagame (e.g., narrative, instructions).

### In-Progress Findings

Preliminary results explore assessing the CT expressed in game designs of the five case study designers. Scores from Dr. Scratch are first discussed, followed by a qualitative assessment of game design. We discuss the discrepancies between these two measures, and how they represent differing evaluations of computational thinking.

#### *Dr. Scratch Metrics for Assessing CT*

The CT score calculated by Dr. Scratch for the final game presented by each designer is shown in Table 1. Dr. Scratch currently calculates these scores such that when the proficiency factor is achieved in the code, the CT that would be necessary to include the basic and developing factors of a particular metric is assumed. For example, the "if" and "if else" represent basic and developing factors in the "logical thinking" metric while "logic operations" represents the proficiency factor. Including a single logic operation in a game would result in a score of 3 out of 3 regardless of whether "if" and "if else" blocks were implemented. A separate *cumulative* CT score calculated by assuming equal importance of each basic, developing, and proficiency factor is shown in parentheses in Table 1. A plugin for calculating

this new cumulative CT score is available at <https://github.com/jemole/hairball/blob/master/hairball/plugins/masteryNEU.py>

From both sets of CT scores it is evident that many of the designers faced similar programming challenges. For example, almost all designers delineated game characters and sprites by their default names, even though renaming them improves code accessibility and readability. Similarly, sprite attributes were most often uninitialized, meaning the value for these attributes was also decided by Scratch. Furthermore, rather than implementing user defined blocks, most simply copied repeated blocks, a process which often leads to programming bugs [5, 4]. From these scores it is also apparent that designers omitted "repeat until" blocks, list data structures, user defined blocks and cloning, and webcam or audio input. While these blocks may have been omitted for game design reasons, it is most likely due to lack of familiarity with these functions.

While the online version of Dr. Scratch that computes the CT score in this paper has problems recognizing lists of logic operations and sometimes increased CT score based on unused blocks, these issues have been fixed in the newest version of Dr. Scratch. Future work aims to test which type of scoring more accurately reflects the level of CT analyzed qualitatively.

#### *Qualitative Investigation of Designers' Games*

Although Dr. Scratch reports similar scores in design complexity for each of the five games, the qualitative analysis indicates more variety in CT, where more complex game designs and realistic portrayal of climate change problems are assumed to represent higher levels of CT [7]. Interestingly, the qualitative analysis performed in terms of interactive design choices indicates that Cyndia's game is perceived as exhibiting more CT than the others, a result supported by the original CT score computed by Dr. Scratch.

Participants	Computational Thinking Scores				
	Tina	Rianna	Linda	Elina *	Cyndia
Games	Ballerina and Cow	Factory Game	CO <sub>2</sub> in the Park	Planting Tree	Methane Cow
Final Score (out of 20)	16 (13)	16 (15)	16 (14)	17 (17)	18 (14)
Flow Control	2 (2)	2 (2)	2 (2)	2 (2)	2 (2)
Data representation	2 (2)	2 (2)	2 (2)	2 (2)	2 (2)
Abstraction	1 (1)	1 (1)	1 (1)	2 (2)	3 (2)
User Interactivity	2 (2)	2 (2)	2 (2)	2 (2)	2 (2)
Synchronization	3 (2)	3 (3)	3 (3)	3 (3)	3 (3)
Parallelism	3 (2)	3 (3)	3 (2)	3 (3)	3 (2)
Logic	3 (2)	3 (2)	3 (2)	3 (3)	3 (1)
Sprite attributes	*	*	*	*	*
Duplicated Scripts	*	*	*	*	*
Sprite naming	*		*	*	*
Dead code					*

**Table 1:** CT Scores of Games Calculated by Dr. Scratch and By Hand (in Parentheses)

### Games by Tina & Rianna



**Figure 1:** Tina's Game: Cow Game



**Figure 2:** Rianna's Game: Factory

She programmed multiple scenes (stages) and accounted for several different forms of user interaction. The following sections include qualitative analysis of each of the games.

*Tina & Rianna: Point and Click, Greenhouse Gases Games*  
Both Tina and Rianna made similar game design choices with content and player interface. In Tina's game about "Cows" (Figure 1), players press the arrow keys to control a ballerina avatar that jumps through methane bubbles emanating from a cow with the goal to get "all of the orange bubbles out of the atmosphere." In Rianna's "Factory Game" (Figure 2), players click on carbon clouds emanating from the factory to make them disappear. Both include basic instructions and are about mitigating harmful environmental factors somewhat unrealistically as neither jumping through nor clicking on methane bubbles would sequester them.

At the end of the workshop, rather than explaining how methane contributes to systems of climate change, Tina articulated a simple causal relationship between cows, methane, and the atmosphere: "It's related to global warming because the methane is bad and it goes into the atmosphere and it never really leaves and it's harmful." On the other hand Rianna not only wants players to understand how much pollution factories produce, but also explains a feedback-loop-type concept in her game, "[These clouds] keep appearing and appearing and they sort of never stop because factories, they don't really stop unless you actually stop them. As long as we buy stuff and as long as they're making stuff, they're not going to stop." Despite the intended feedback loop, her design decisions lack that complexity. Both games reduce the complexity of climate change issues to a linear relationship between clicking or jumping through clouds to make them disappear, a simple point and

### Games by Linda & Elina



**Figure 3:** Linda's Game: CO<sub>2</sub> in the Park



**Figure 4:** Elina's Game: Planting Trees

click dynamic that may not facilitate meaningful engagement with the inherent complexity of climate change.

#### *Linda & Elina: Two Scenic, Green Mitigation Games*

Linda & Elina also made similar design decision with dynamics and metagame content. In Linda's game called "CO<sub>2</sub> in the park"(Figure 3), players are prompted to first identify the CO<sub>2</sub> sinks in the picture (plants) and then click them as much as possible before time runs out to mitigate CO<sub>2</sub> in the atmosphere. In Elina's game "Planting trees to eliminate CO<sub>2</sub>" (Figure 4), players control a bee who drops seeds at desired locations. In both systems, players must identify and object or location and then perform an action.

Both of these games represent climate change in a more realistic way than those designed by Tina and Rianna. While in reality bees do not fly around holding seed packets, they do spread plant growth through pollination, which does in fact eliminate CO<sub>2</sub> from the atmosphere. Similarly, like real climate change, there is a limited amount of time for humans to react.

While Tina and Rianna's games represent visual content on the screen as methane bubbles or factory pollution, Linda and Elina's games show green grass, people, or plants within a park. Further, these two designers discuss human unawareness of the problem and player perceptions of the game. For example, when explaining her game in the end-of-workshop-presentation, Linda describes how humans' unawareness of the problem factors into her visual design choices, saying: "Well you see these people sitting on the fence they're kind of unaware of this big cloud of smoke right there so which represents the CO<sub>2</sub>." Elina points out in her presentation that there's a time limit and why that is important: "There's a tradeoff as the person playing the game is that you have the choice to use time to plant trees and help the air or you can not plant trees and it wouldn't help

the environment." Both designers represent aspects of mitigation on screen, like plants and are capable of perspective switching (e.g. imagined human perspective of others, the projected players' perspective). For these reasons, these games may represent more CT in planning and execution than in those of Tina and Rianna.

#### *Cyndia: A Multi-Step Game with Diverse Player Interaction*

Cyndia's "Save the Planet!" game(s) include multiple player dynamics and informational screens. On her survey, Cyndia reports that she taught herself Scratch in 3rd or 4th grade, so her elaborate design may be in part due to her previous design experience. The game appears programmed to change experiences upon repeated play. The first experience opens with an information screen and a game to collect methane bubbles from cows (similar to Tina's game). Next there is a section phrased as a quiz where the player clicks the carbon sink on the screen. Third, the player drags objects to either a trash or a recycling bin. Informational screens transition between each step.

Interestingly, Cyndia appears to have a complex perception of the many factors involved in climate change. She clearly articulates her perception of climate change, and the multifaceted design of her game manifests the type of meaningful play she wanted players to experience. She also articulates that other players may not know about climate change, and that she wants to teach them about recycling procedures and the urgency of mitigation before the situation worsens. "A lot of people don't know if things are recycling or trash and they end up putting things that are trash in the recycling or you put things that are recycling in the trash and if they incinerate them then that creates extra fossil fuels..." Visually, Cyndia includes methane gases within one game, similar to Tina, but the other screens include recycling bins and colorful quiz screens. While she

## Game by Cyndia

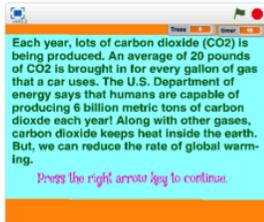


Figure 5: Scene 1



Figure 6: Scene 2



Figure 7: Scene 3



Figure 8: Scene 4

represents a "problem" visually in the first game, she also follows this representation with mitigation practices.

## Discussion and Future Work

While the results indicate differences between the quantitative and qualitative analyses, the goal of this work is to both explore the constraints of such analyses and improve the computational scoring to provide more meaningful feedback to students and instructors.

Qualitatively, it is possible that the simplicity of the Scratch environment does not facilitate complex game designs. For example, Tina and Rianna both created games portraying a polluting effect of climate change with a simple point and click dynamic, which is an overly simplistic interpretation of mitigation. Linda and Elina designed slightly more complex games, where players "identify" and then click, which were slightly more mechanically and visually realistic (both portrayed green outdoor scenes). Because higher complexity in the games and the designers intentions was assumed to represent an expression of higher levels of CT, Linda and Elina are thought to have expressed slightly more CT.

Some of the simplicity in the designs could be because of the designer's varying levels of experience with Scratch. Tina reported one year of experience, while Rianna and Linda reported no previous experience. Elina reported about eight months of coding experience in Scratch while Cyndia had been coding in Scratch for years. However, regardless of the intention of the designers, looking at only the games still evaluates the level of expressed CT. Interestingly, Linda and Elina discussed both human and player perspectives, and may also be examples of more agentic engagement in terms of design of an experience that has a goal of motivating its player for change.

We propose that the combined qualitative and quantitative

analysis can help improve the tool and its assessment of CT. The Dr. Scratch team has already made a plug-in to assess programs with cumulative scoring, and we plan on exploring which summation strategy more closely aligns with the CT qualitatively observed in games.

While Dr. scratch provides insight into computational thinking in games and feedback, we plan on developing more metrics to comprehensively assess CT. With only the Dr. Scratch results, adequately assessing and comparing project complexity in terms of computational thinking was difficult. With the Dr. Scratch team we are developing metrics that are included in a qualitative coding rubric that are currently not assessed with Dr. Scratch. Some of the important factors not captured here might be length of the games, number of code blocks, number of variables, and designer motivations that we would like to include in our own quantitative and qualitative rubrics in the future to measure CT.

Following this pilot study, our research team has been developing a curriculum for supporting students designing climate change games in Scratch. Two local teachers were observed during their weather and climate unit, and have implemented our curriculum in their classrooms. We currently have many student designed games and extensive data on case-study student pairs. Our future work will focus on exploring how our different data sources measure computational thinking differently, and how we can automate any of these evaluations in our future analyses.

## Acknowledgments

We want to thank the workshop participants. Furthermore, this material is based upon work supported by the National Science Foundation under Grant No. 1542954.

## REFERENCES

1. Ashok R Basawapatna, Kyu Han Koh, and Alexander Repenning. 2010. Using Scalable Game Design to Teach Computer Science from Middle School to Graduate School. In *Proceedings of the Fifteenth Annual Conference on Innovation and Technology in Computer Science Education*. ACM, 224–228.
2. Mark Guzdial. 2008. Education Paving the way for computational thinking. *Commun. ACM* 51, 8 (2008), 25–27.
3. Casper Hartevelde. 2011. *Triadic Game Design: Balancing Reality, Meaning and Play*. Springer.
4. Miryung Kim, Lawrence Bergman, Tessa Lau, and David Notkin. 2004. An ethnographic study of copy and paste programming practices in OOPL. In *Empirical Software Engineering, 2004. ISESE'04. Proceedings. 2004 International Symposium on*. IEEE, 83–92.
5. Zhenmin Li, Shan Lu, Suvda Myagmar, and Yuanyuan Zhou. 2006. CP-Miner: Finding copy-paste and related bugs in large-scale software code. *IEEE Transactions on software Engineering* 32, 3 (2006), 176–192.
6. Jesús Moreno-León, Gregorio Robles, and Marcos Román-González. 2015. Dr. Scratch: Automatic Analysis of Scratch Projects to Assess and Foster Computational Thinking. *RED. Revista de Educación a Distancia* 15, 46 (2015).
7. Jesús Moreno-León, Gregorio Robles, and Marcos Román-González. 2016. Comparing Computational Thinking Development Assessment Scores with Software Complexity Metrics. In *2016 IEEE Global Engineering Education Conference (EDUCON)*. IEEE, 1040–1045.
8. Chrystalla Mouza, Yi-Cheng Pan, Lori Pollock, James Atlas, and Terry Harvey. 2014. Partner4CS: Bringing computational thinking to middle school through game design. (2014).
9. Mark Overmars. 2004. Teaching computer science through game design. *Computer* 37, 4 (2004), 81–83.
10. Mitchel Resnick, John Maloney, Andrés Monroy-Hernández, Natalie Rusk, Evelyn Eastmond, Karen Brennan, Amon Millner, Eric Rosenbaum, Jay Silver, Brian Silverman, and Yasmin Kafai. 2009. Scratch: Programming for All. *Commun. ACM* 52, 11 (2009), 60–67.
11. Jeannette M Wing. 2006. Computational thinking. *Commun. ACM* 49, 3 (2006), 33–35.
12. Min Lun Wu and Kari Richards. 2011. Facilitating computational thinking through game design. In *International Conference on Technologies for E-Learning and Digital Entertainment*. Springer, 220–227.