Computational Classrooms: A Constructivist, Research-Based Approach to Designing a Computer Science Course for Elementary and Middle School Teachers

Introduction

To address the complex threats to Earth's life-sustaining systems, students need to learn core concepts and practices from various disciplines, including mathematics, civics, science, and, increasingly, computer science (NRC, 2012; United Nations, 2021). Schools must therefore equip students to navigate and integrate these disciplines to tackle real-world problems. Over the past two decades, STEM educators have advocated for an interdisciplinary approach, challenging traditional barriers between subjects and emphasizing contextualized real-world issues (Hoachlander & Yanofsky, 2011; Vasquez et al., 2013; Ortiz-Revilla et al., 2020; Honey et al., 2014; Takeuchi et al., 2020).

Despite extensive evidence supporting integrated approaches to STEM education, subject boundaries remain, with disciplines often taught separately and computer science and computational thinking (CS & CT) not consistently included in elementary and middle school curricula. In today's digital age, CS and CT are crucial for a well-rounded education and for addressing sustainability challenges (ESSA, 2015; NGSS Lead States, 2013; NRC, 2012). While there's consensus on the importance of introducing computational concepts and practices to elementary and middle school students, integrating them into existing curricula poses significant challenges, including how to effectively support teachers to deliver inquiry instruction confidently and competently (Ryoo, 2019).

Existing frameworks and tools for teaching CS and CT often focus on maintaining fidelity to canonical concepts and formalized taxonomies rather than on practical applications (Grover & Pea, 2013; Kafai et al., 2020; Wilkerson et al., 2020). This focus can lead teachers to learn terminology without fully understanding its relevance or application in different contexts. In response, some researchers suggest using a learning sciences perspective to consider "how the complexity of everyday spaces of learning shapes what counts, and what should be counted, as 'computational thinking'" (Wilkerson et al., 2020, p. 265). These scholars emphasize the importance of drawing on learners' everyday experiences and problems to make computational practices more meaningful and contextually relevant for both teachers and their students. Thus, this paper aims to address the following question: How can we design learning experiences for in-service teachers that support (1) their authentic engagement with computational concepts, practices, and tools and (2) more effective integration within classroom contexts? In the limited space of this proposal, we primarily address part 1.

Study and Course Design

Project Background

In an NSF-funded project, we convened a Research Practitioner Partnership (RPP; Penuel & Gallagher, 2017; Coburn, Penuel, & Geil, 2013) composed of 10 university researchers, content specialists, and current and former elementary and middle school teachers to address the problems of practice outlined above. The RPP focused on developing principles for modifying existing curricula to include more sustainability and computational education, and on developing a professional learning model to support teachers in that work. The RPP met once monthly to develop shared visions and goals, reflect on our professional practice, dive deeper into issues of sustainability and CT (with a focus on purposeful integration), analyze various institutional documents (e.g., UN sustainability goals; State CS standards), modify county curricula, and engage in contextualized computational inquiry. During this time, we tested out several activities for a graduate-level course for in-service teacher leaders entitled, "Teaching and Learning

Computer Science and Computational Thinking for Elementary and Middle School," which will be co-taught by the authors of this paper (who are also members of the RPP), in the Fall of 2024.

Course Design: The Inquiry Cycle

Our course design was informed by research suggesting that "top down" instructivist approaches to computational education that focus on definitions and taxonomies of computational practices disconnected from everyday, disciplinary, and curricular contexts fail to adequately support teachers to integrate CS and CT into their STEM lessons. We conjecture that taking a "bottom-up" constructivist approach to computational learning will help with teacher buy-in, with making the content more accessible, and with teachers translating what they've learned into the classroom. To do this, we have devised a pedagogical approach that we call *the inquiry cycle*.

Each inquiry cycle takes place over the course of a single 3-hour lesson, and includes three phases: *challenge*, *reflect*, and *connect*. During the *challenge* phase, teachers (students in the course) will be presented with a STEM inquiry problem, which is rooted in real-world contexts, applicable to teachers' and students' daily lives, and is designed to naturally elicit engagement in computational practices. During the *reflect* phase, teachers will "go meta" on the practices they used while completing the challenge. Rather than introduce computational concepts and terminology (like "abstraction" or "algorithmic thinking") ahead of time, instructors will listen for these practices in teachers' collaborative reflections and label them with the canonical terminology after the fact. The class will keep a post-it wall of practices so that over the course of four inquiry cycles, teachers will see the utility of a multitude of computational practices and tools for their STEM inquiry and learning. Finally, in the *connect* phase, teachers will be asked to make explicit connections to what they have seen and want to see their students doing.

After four inquiry cycles addressing core disciplinary issues (unplugged CT, data analysis and representation, algorithms and programming, and machine learning and artificial intelligence), teachers will have the opportunity to design an inquiry cycle for their peers. Finally, they will be tasked with modifying their own elementary and middle school STEM lessons to include more computational tools and practices. They will test these modified lessons with their students, and refine based on classroom data they collect.

Pilot Testing and Data Collection of Inquiry Activities

In preparation for the course, we piloted three inquiry challenges with the RPP team, many of whom are current or former classroom teachers and who have had little to no formal training in CS or CT education. The first inquiry challenge we piloted, for the Data Analysis and Representation lesson, was a *Track Your Trash* activity, where participants recorded what they threw away over the course of a few days. They worked in groups to clean a messy data set, and then analyze and represent the data in order to convince a stakeholder of their choice to make a meaningful change. For instance, one group calculated the ratio of the RPP's organic to non-organic waste to convince the county to institute a composting program, and another group used estimates of single-use plastic consumption at a household, neighborhood, and county level to increase consumer awareness of the impact that individuals' plastic disposal has on a larger scale. In the second inquiry challenge, for the Unplugged CT week, RPP members played common household games, including a *Guess Who?* game with animals and over/under number guessing games. In the third inquiry challenge, for Machine Learning and Artificial Intelligence week, RPP members started off by playing Google's Quick, Draw! (a game where the computer guesses a drawn object) in pairs. After discussing what they thought the computer was doing in order to guess what was being drawn so quickly and effectively, they were instructed to explore the millions of drawings in the training data and revise their conjectures. Finally, they used another Google-operated platform called Teachable Machine to build a waste sorter, effectively training the machine to recognize various classes of objects. They similarly reasoned about what they thought the program was doing, and why/how it operated the way it did (sometimes in expected ways and other times, in very unexpected ways). Following each inquiry *challenge*, members engaged in a group *reflection* about the strategies they used, and afterwards categorized and labeled these strategies using more formal computational practices. They then made explicit *connections* to the classroom.

The inquiry challenges have been deliberately selected to ensure their relevance to the educators participating in the course. Although the connections may not be immediately apparent to the teachers, each activity is intricately linked to a core computational principle or topic. The Unplugged activity, for instance, enables participants to apply both abstraction and algorithmic thinking as they develop strategies to consistently win the games. The *Track Your Trash* activity involves pattern recognition and algorithmic thinking as participants analyze weekly trash disposal data to identify trends and make estimations at larger scales. Additionally, the AI and machine learning module serves as an introduction to systems like large language models or recommendation algorithms and requires the use of practices like abstraction, tinkering, decomposition, debugging, and pattern recognition.

While we plan to collect data during the course we will offer in the Fall, data for this proposal consist of recorded RPP meetings of these three piloted inquiry cycles, “exit tickets” filled out by RPP members after each meeting, and a series of RPP member interviews conducted by an external evaluator at the end of each project year. In what follows, we provide illustrative evidence of the benefits of our constructivist approach for RPP members’ engagement in, understanding of, and developing comfort with CT and CS concepts and practices.

Findings and Analysis

Reflecting on the *Track Your Trash* activity, “Eleanor” (a retired 4th grade teacher) said,

> “I think we definitely had to **break the problem down into very small pieces** and try to **identify what kind of data we were gonna need**. You know, we had to **figure out what the question was first**, and we picked ours. But then we really had to think, okay, so what kind of information would we need? And **what kind of data would answer that question**?”

Notice here that Eleanor highlighted some computational practices her group engaged in, but in colloquial language. She highlighted multiple practices, involving both how to approach and simplify a problem and also how to select and utilize data to answer a specific question. “Kerri,” a content specialist and retired computer scientist, picked up on the practices Eleanor shared, and labeled them with more precise terminology. She said,

> “So I can't resist putting in a comment. From what I heard you say. I heard **decomposition**. And I also heard **identifying a problem** which are 2 things that are very important in computer science. If this were me and this were my class, I would stand in my chair and say. I'm so excited. You came to that conclusion.”

By allowing RPP members (and eventually students) to organically construct insights into their learning using their own language and labeling them with the “correct” terminology afterwards, it empowers them to define computational thinking in ways that they can recognize as naturally integrated with their science reasoning. We can see some of this self-efficacy in how Eleanor reflected on an interview she recently did with a project evaluator. In response to Kerri, Eleanor shared that when asked about whether she felt like she understood what computational thinking was, she proudly told the interviewer that "well, I know I know it a lot better than I did a year ago."

During the debrief following the pilot of the Unplugged CT activity, "Christina," a highschool mathematics teacher (and former elementary teacher) stated,

> "I think as adults, like we, **we all came in with like a strategy**, some sort of strategy when we were looking at either the animals or the number… like when you're like doing a word problem. I think a lot of students…just get tripped up looking at the problem itself and they get overwhelmed. But it's like...what can you start doing? Like **it's a concept of like narrowing something down into a smaller and smaller, more manageable chunk.** Um, I think it can be applied in math class, and it can be a really good strategy to help them manage problems that might require more rigor than they're comfortable with."

By emphasizing the fact that all of the members approached the games with different strategies, Christina highlights one of the strengths of this bottom-up design. Members are able to engage with the concepts at multiple levels of complexity and still achieve success in the task at hand. She then goes on to describe the process of decomposition using informal language. Lastly, she reflects on how this activity can be used to potentially prevent the performance anxiety that might arise when students are faced with problems that require them to work outside their comfort zone. Eleanor expanded on this idea by remarking on ways that this method can be utilized in other subjects. She said,

> "I mean, I even see how some of it…more like the elimination [in the Guess Who? game] could also work with…vocabulary words too. Like you could have a set of vocabulary words. And students could then ask those questions like, I'm thinking of a word and…use like certain things like vowels, consonants, like they have, they can use other strategies. But so…it's the concept of the computational thinking, but in not just math and science too. Yeah, certainly doesn't have to be limited to math and science for sure."

Of note here, is the way the inquiry activity led to a robust and meaningful connection to the teachers' own practice. By participating in these activities that bring out computational practices organically, participants were able to be more engaged in the disciplinary substance while also providing opportunities to reflect on their own pedagogy. Teachers are able to make connections between active reasoning and computational thinking in such a way that the successful application of this translational practice becomes readily apparent. Eleanor summarizes her thoughts by saying,

> "I also noticed how these games relate to computational thinking in a relatable way. It certainly makes computational thinking easier to grasp and understand!"

During the discussions, members observed that in both games, they subconsciously engaged in algorithmic thinking and problem decomposition in their efforts to win. Other emerging themes from the discussion included potential modifications to the games to further enhance the implementation of computational thinking principles and how engaging students with games might increase their inclination to conduct further research to improve their gameplay, while simultaneously improving their knowledge of the subject matter. Participants also explored ways to modify the games to suit multiple disciplines, such as mathematics, environmental studies, and vocabulary.

Exit tickets were collected at the end of each meeting. One of the questions on the exit ticket following the Machine Learning and Artificial Intelligence activities was “Did engaging in the AI/Machine Learning activity support your understanding of Computational Thinking? If so, how?” Eleanor replied,

> “The AI/Machine learning activities support my understanding of CT because it proves that there are necessary steps, skills, procedures, and information needed to carry out certain tasks. With the quick draw, it helped me to see **how the computer used abstraction**, when it first focused on the basic shapes, then using **decomposition** as the drawing continued in order to better identify the picture being drawn. After, the was able to use its **algorithm and pattern recognition** to determine the object being drawn. I also found the data and prior samples fascinating because it led to more “ah-has” and questions about what the computer is able to recognize and utilize in its process. These activities the past couple weeks have overall given me more confidence with my own understanding and definition of CT.”

Eleanor was able to connect the activities to her understanding of CT by stressing the importance of steps, skills, and procedures in task execution. This demonstrates an understanding of CT as a structured problem-solving process. “Amelia,” (a current elementary school educator) responded to the question with,

> “It did for sure. Especially when I realized that the computer was making **several iterations** of each image in order to more **accurately define an object/action**. It made me think about humans and language acquisition. Like when you're learning a new language the more that **you see the same word** the better you get at recalling it. Also, programming a computer to do this by **breaking down steps** is now making more sense. For example, there must be a code that tells the computer to take several images. A step I never would've thought of but completely understand its necessity.”

This response shows how Amelia increased her understanding of the process computers go through to “recognize” objects or actions. She was then able to connect it to her understanding of human learning, specifically in language acquisition. This was further amplified by another participant, a university researcher named “Arthur” who echoed this new way of thinking, saying that “understanding THAT there's **different kinds of algorithms** is an important CT insight for me.” By engaging with the machine learning model the participant began the process of demystifying AI while also recognizing that AI algorithms are doing something fundamentally different from algorithms used in block-based programming, for instance.

Contribution to the Teaching and Learning of Science and Impact on NARST Members

As our data have shown, a “bottom-up” approach to teaching CS and CT that is rooted in meaningful and contextualized problems can support learners’ self-efficacy and understanding of core concepts and practices. Data collected from this course will help us determine the efficacy of this professional learning model for practicing in-service teachers. If the model proves to be effective, we will share the course curriculum with other teacher educators. It is our hope that this model will support teachers in successfully integrating computational practices and tools into their STEM classrooms, since most are not provided time for a designated CS block. Our findings will also contribute to research on transdisciplinary learning in teacher education.

References

Coburn, C.E., Penuel, W.R., & Geil, K.E. (2013). *Research-Practice Partnerships: A Strategy for Leveraging Research for Educational Improvement in School Districts*. William T. Grant Foundation, New York, NY.

Every Student Succeeds Act, 20 U.S.C. § 6301 (2015). https://www.congress.gov/bill/114th-congress/senate-bill/1177

Grover, S., & Pea, R. (2013). Computational thinking in K–12: A review of the state of the field. *Educational researcher*, *42*(1), 38-43.

Hoachlander, G., & Yanofsky, D. (2011). Making STEM Real. *Educational Leadership*, *68*(6), 60-65.

Honey, M., Pearson, G., & Schweingruber, H. (2014). *STEM integration in K-12 education: Status, prospects, and an agenda for research*. Washington D.C.: National Academies Press.

Kafai, Y., Proctor, C., & Lui, D. (2020). From theory bias to theory dialogue: embracing cognitive, situated, and critical framings of computational thinking in K-12 CS education. *ACM Inroads*, *11*(1), 44-53.

National Research Council. (2012). *A framework for k-12 science education: Practices, crosscutting concepts, and core ideas.* Washington, DC: The National Academies Press.

NGSS Lead States. 2013. *Next Generation Science Standards: For States, By States.* Washington, DC: The National Academies Press.

Ortiz-Revilla, J., Adúriz-Bravo, A., & Greca, I. M. (2020). A framework for epistemological discussion on integrated STEM education. *Science & Education*, *29*(4), 857-880.

Penuel, W. R., & Gallagher, D. J. (2017). *Creating Research-Practice Partnerships in Education*. Cambridge, MA: Harvard Education Press.

Ryoo, J. J. (2019). Pedagogy that supports computer science for all. *ACM Transactions on Computing Education (TOCE)*, 19(4), 1-23.

Takeuchi, M. A., Sengupta, P., Shanahan, M. C., Adams, J. D., & Hachem, M. (2020). Transdisciplinarity in STEM education: A critical review. *Studies in Science Education, 56*(2), 213-253.

United Nations Department of Economic and Social Affairs. (2021). *The Sustainable Development Goals Report*. https://doi.org/10.18356/9789210056083

Vasquez, J. A., Sneider, C., & Comer, M. (2013). *STEM lesson essentials, Grades 3–8: Integrating science, technology, engineering, and mathematics.* New York: Heinemann.

Wilkerson, M. H., D’Angelo, C. M., & Litts, B. K. (2020). Stories from the field: locating and cultivating computational thinking in spaces of learning. *Interactive Learning Environments*, *28*(3), 264-271.

Abstract. This paper presents the design, testing, and implementation of a professional learning model to help in-service educators integrate computer science (CS) and computational thinking (CT) into their pedagogy. The central question addressed is: How can we create learning experiences that support teachers' authentic engagement with computational concepts, practices, and tools? The authors are part of a Research Practitioner Partnership (RPP), which meet monthly to develop principles for modifying curricula to include sustainability education and CT. Stressing a “bottom-up” constructivist approach, teachers are able to engage with real-world problems that naturally lead to computational understandings. The approach centers on inquiry cycles, where teachers engage with CS principles, reflect, and connect experiences to classroom practices. These cycles, prototyped during RPP meetings, aim to build confidence in delivering transdisciplinary lessons incorporating CS and CT. Data from the meetings indicates that the inquiry activities demystified computational principles, making them more accessible and relevant to educators, suggesting that a “bottom-up” approach to teaching CS and CT that is rooted in meaningful and contextualized problems can support learners’ self-efficacy and understanding of core concepts and practices. If successful, this model could help teachers effectively integrate CS and CT into their STEM classrooms.

Data

So I can't resist putting in a comment. From what I heard you say. I heard **decomposition**.

Kathy Benson
And I also heard **identifying a problem** which are 2 things that are very important in computer science. If this were me and this were my class, I would stand in my chair and say. I'm so excited. You came to that conclusion.
Ann Johnson
Well, you know, when we did that interview I don't know if everybody's done it or not. And she asked, well, you know, do you think you understand what computational thinking is? I was like, well, **I know I know it a lot better than I did a year ag**o. That's all I can say I could attest to that. So

Amra Nansimbi
01:57:19
I think, when Jonathan said that people were the computers or humans were the computers that made me think of the fact that, like when we were looking at the data in our team, we were trying to **look for patterns** and notice what exactly is happening

here in order to create some kind of conclusion with it to formulate. So I don't know that human element really just clicked with me.
Thanks, thanks for that.

Amy Green
And
so I think I think that's a really cool. **This is a really cool kind of activity to demonstrate, like just what good work computing and computational thinking do for us?**
And and it and I for our grouping, communicating out to other people could like, persuade, or or if so. so, I thought, that was.

01:38:03.660 --> 01:38:17.080
Amy Green: this so this whole thing of **empathy, and like including humans and making personal connections as being integral to the sustainability integration piece is a we start contrast to like traditional purchase, to science, learning, and school, which is typically emphasizes, like impersonal, impartial, objective, not having to do it like an idea.** I'm. Taking one step. Obviously not having like you. You have to keep the person out of it for it to be truly scientific. And I think that slowly, like we're gonna like. I think that's a good space for us to they'd be contributing to like we' this is another argument for why that's not always the most productive way to a present science in schools, but also to engage students in learning, science, ideas, and schools. So we've got this one part of an argument for why, you know, humanifying the content and the processes of the science is impactful. For, as I was saying, like engaging with being able to engage with the content, and like happy with being able to recognize it's the importance in their own lives which will have to do with Yeah, that has to do with like learning theory stuff. and then also looking forward. It's also I think, going to be an a helpful way to connect when we start to look at how to better support language. Learners like not only English language learners to be able to but make it make it more human. I think we're kind of See, there's a lot of There's there's a lot of good strategies connected to that being an intentional. So I don't know. I just. I feel like all this. What what we're doing is making pretty strong arguments for further disrupting that that status flow perception that science is supposed to be kind of that again, and impersonal. And you know that we prick, and writers have worked hard for a long time to keep you out of it, you know. like humans are studying something from afar. They're not part of the system that being studied. Anyway. The I thought that was really interesting

864
01:40:11.880 --> 01:40:36.630
Jonathan R: Yeah. One of the things that we discussed was the because of the fact that everyone came with disparate types of data and different variables that in order for us to come up with that other number that we would have to estimate to a certain extent, and that is part of, you know, that's part of research that sometimes. **So we did have to come up with an estimation for that. But it was based off of the rationale of how much we had used per the week, and we would just kind of scaled up from there. And of course, my brain, hearing this and hearing the conversion of how many cars it is. I'm thinking of. The students utilizing scratch and they click a button on scratch, and when they click the button, it'll start populating cars on the**

screen, and you can have it so that the limit is 1,000 blah blah blah cars and the the screen will just. It'll just continuously fill up with cars until it reaches that number, and then they could have a big psa. That says this is how much plastic or something along those lines. But that's just another way to think, how can you utilize technology to really convey this information? And that's like, sure we can do a Psa. We can do a poster. We can do a social media post lots of different things, but it's all about who is who are we trying to reach? And how are we going to appeal to them whether we're appealing to their emotions, whether we're appealing to their sense of you know, empathy like, what is it that we're trying to do? What is the argument that we're trying to make? And how are we gonna get that across using data.

Amy Green: and 1 one last takeaway that I wanted to share. Dennis, I think, like this: this whole humanizing piece is is super compelling to me, and I feel like It's going to be interesting when we start talking more about computer science, computational thinking, because I think there's often a general assumption that those things are in person impersonal non-human, too, like we're talking about stuff that machines do or sometimes people say thinking like a machine. But I know that a big part of that conversation is to to humanize the computational thinker, the human behind the computing. So I think this is a really cool space that we're okay.

01:57:19.250 --> 01:57:22.420
Amra Nansimbi: I think, when Jonathan said that people were the computers or humans were the computers that made me think of the fact that, like when we were looking at the data in our team, we were trying to look for patterns and notice what exactly is happening here in order to create some kind of conclusion with it to formulate. So I don't know that human element really just clicked with me.

1076
02:01:24.690 --> 02:01:27.560
Kathy Benson: I love that, Jonathan, I think to automate complex problems is takeaway that once you're doing division of a million. it's time to put a tool in and Our group was talking about skill with spreadsheets, and how there's some variability in teachers with that. So part of the.

00:54:29,680 --> 00:54:36,580 (Laurel)
We were back to the animal game, as we said, like, if you don't know certain characteristics or certain qualities about the animals, **it encourages you to research them and to find out information. Um, so it could go with almost any topic that students are learning.** If they don't have that information, it becomes a learning opportunity to find that information, that background information further. So it can help them learn about animals is what you're thinking. Yeah. Learn more about animals. You know. Or it could be whatever the topic is. I mean, it could be they could be saying something else to it. I mean, but in this specific example, it's animals, but it could be something else. Um, could be habitats, could be, um, types of rocks. Like it can just be depending on what that topic is on that subject is.

00:56:10,160 --> 00:56:17,149 (Claire)
I think when it comes to I guess it's like for both of these, I think as adults, like we, **we all came in with like a strategy**, some sort of strategy when we were looking at either the animals or the

number line, something like, definitely in math. Um. And in science. He was just like having a strategy in general, like when you're like doing a word problem. I think a lot of students like they just get tripped up looking at the problem itself and they get overwhelmed. But it's like, hey, like just like, what can you do? Like what can you start doing? Like **it's a concept of like narrowing something down into a smaller and smaller, more manageable chunk.** Um, I think can be applied in math class, and it can be a really good strategy to help them manage problems that might require more rigor than they're comfortable with.

00:56:57 (Laurel) I mean, I even see how some of it like more like the elimination could also work with like an L.A. could really easily work on like, vocabulary words too. Like you could have a set of vocabulary words. And students could then ask those questions like, I'm thinking of a word and. But they have to use like certain things like vowels, consonants, like they have, they can use other strategies. But so it just **it it's the concept of the computational thinking, but in not math and science to. Yeah, certainly doesn't have to be limited to math and science for sure.**

00:58:21,450 --> 00:58:28,680 (Amy)
There's so many examples. And that's what I really like about this is **it's a really authentic way to simultaneously strengthen, like disciplinary understandings and while also strengthening, um, computational thinking skills and strategy is like it, it does good work for both of those at the same time** without one necessarily being like like they're both in service of each other. One isn't being like, um, you know, sacrificed for the other, you know?

Andy (Exit Ticket) (AI)
Hmmm, now you've got me thinking about what kind of thinking I was using when trying to figure out how the AI works. The guts of the AI is definitely a different kind of algorithm than the kind we usually think about in CT terms... So I guess understanding THAT there's different kinds of algorithms is an important CT insight for me.

Laurel (Exit Ticket) (AI)
The AI/Machine learning activities supports my understanding of CT because it proves that there are necessary steps, skills, procedures, and information needed to carry out certain tasks. With the quick draw, it helped me to see the how the computer used abstraction, when it first focused on the basic shapes, then using decomposition as the drawing continued in order to better identify the picture being drawn. After, the was able to use it’s algorithm and pattern recognition to determine the object being drawn. I also found the data and prior samples fascinating because it led to more “ah-has” and questions about what the computer is able to recognize and utilize in its process. These activities the past couple weeks have overall given me more confidence with my own understanding and definition of CT.

Amra (Exit Ticket) (AI)
It did for sure. Especially when I realized that the computer was making several iterations of each image in order to more accurately define an object/action. It made me think about humans and language acquisition. Like when you're learning a new language the more that you see the same word the better you get at recalling it. Also, programming a computer to do this by breaking down steps is now making more sense. For example, there must be a code that tells the

computer to take several images. A step I never would've thought of but completely understand it's necessity.

Laurel (Exit Ticket) (Unplugged)
The games were lots of fun and engaging! It was insightful to hear how others interpreted the games and the strategies they chose to use. I also noticed how these games relate to computational thinking in a relatable way. It certainly makes computational thinking easier to grasp and understand!

Kathy (Exit Ticket) (Unplugged)
I thought the connect was effective in abstracting that these kind of activities are fun for any thing that has categories or attributes. In terms of pedagogy, it can be done in a number of formats (e.g., guess who, make your own sorts or groups, odd one out, mystery venn diagrams, or concept attainment). By concept attainment, I mean you don't tell others what the concept (category or abstraction you are illustrating), you just revel on by one examples and non-examples until someone guesses the concept.

A few minutes later, Jack, a former elementary school teacher, shared a recent insight about how in the movie *Hidden Figures*, about NASA scientists in the 1960's, they referred to people as 'computers' because they did all of the computing by hand. In response, Angela said,

> "I think, when Jack said that people were the computers or humans were the computers that made me think of the fact that, like when we were looking at the data in our team, we were trying to **look for patterns** and notice what exactly is happening here in order to create some kind of conclusion with it to formulate. So I don't know, that human element really just clicked with me. Thanks, thanks for that."

Angela had an "aha" moment, that computational thinking is a process by which *humans* think critically *about* computational processes whether or not a computer is actually involved.

"Arthur," (a university researcher) responded:

> Hmmm, now you've got me thinking about what kind of thinking I was using when trying to figure out how the AI works. **The guts of the AI is definitely a different kind of algorithm than the kind we usually think about in CT terms... So I guess understanding THAT there's different kinds of algorithms is an important CT insight for me.**

In the process of attempting to train a machine to recognize different types of recyclable materials, this participant's curiosity as to the inner machinations led to some higher order thinking.