The Use of Makerspaces for the Development of Computational Thinking Skills and Dispositions

BY

ROXANA HADAD

B.A., University of Illinois at Urbana-Champaign, 1998

M.P.S., New York University, 2002

THESIS

Submitted as partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Educational Psychology
in the Graduate College of the
University of Illinois at Chicago, 2020

Chicago, Illinois

Defense Committee:

Kimberly A. Lawless, Chair and Advisor
Michael K. Thomas, Educational Psychology
Theresa A. Thorkildsen, Educational Psychology
Yue Yin, Educational Psychology
Rachel F. Adler, Northeastern Illinois University

ACKNOWLEDGEMENTS

This dissertation would not have been possible without the cooperation of the study's participants, their families, and the facilitators. I am so thankful for the hard work of the ACTMA team: Yue Yin, Steven Bonomo, Yvette Garcia, Mila Kachovska, Qiao Lin, Dominica McBride, Ephran Ramirez, Germania Solórzano, Jeff Sweeton, Kate Thomas, Xiaodan Tang, and especially C. Meghan Jacobsen, who helped coordinate the logistics and the research for an unwieldly project. A special thanks to the National Science Foundation, which supported the work in this grant under award number 1543124.

I'd like to thank my committee for their insight and guidance: my advisor, Kim Lawless (who actually believed a student could be a PI), as well as Rachel Adler, Michael Thomas, Terri Thorkildsen, and Yue Yin. Irene Lee was instrumental in involving me in the national discussion and community around computational thinking early on. My colleagues at CCAS@NEIU and the Computer Science Equity Project at UCLA were continual sources of inspiration and encouragement. I am also eternally grateful to my Los Angeles writing group, who kept me working on this through a difficult pregnancy.

To my parents, Marcia and Jorge Hadad, in addition to lifelong love and support, you instilled a love of art, science, and learning. To my daughters, Phoebe and Miriam, I am so grateful you were so understanding of a mother that was not always present and for giving me the motivation to continue in my work. Finally, the biggest thank you goes to my husband, Alex Austin, for his continued support, patience, and belief in me. I love you.

TABLE OF CONTENTS

LIST OF TABLES

TABLE PAGE

LIST OF FIGURES

FIGURE PAGE

LIST OF ABBREVIATIONS

ACTMA	Assessing Computational Thinking for Maker Activities
CCSS	Common Core State Standards
CT	Computational Thinking
CTP	Computational Thinking Patterns
CS	Computer Science
CSTA	Computer Science Teachers Association
GEAR UP	Gaining Early Awareness and Readiness for Undergraduate Programs
K-12	Kindergarten through 12^{th} grade
LED	Light Emitting Diode
ISTE	International Society for Technology in Education
NGSS	Next Generation Science Standards
STEM	Science, Technology, Engineering, and Math

SUMMARY

A case study of a day in a summer makerspace program was carried out. The interactions of four facilitators with seventeen high school-aged participants were examined through video, audio, screen recordings, participant notebooks, and observer field notes. Open coding was conducted, and interactions were also coded for computational thinking skills and dispositions, cognitive apprenticeship methods, and level of impact.

Facilitators utilized cognitive apprenticeship methods with tinkering, embodiment, walkthroughs, drawing, and debugging, resulting in participants' tinkering becoming more sophisticated, as well as their demonstrations of their computational thinking skills and dispositions.

I. INTRODUCTION TO THE PROBLEM

A foundation in computer science is integral to a student's understanding of the world today, yet, few K-12 schools are introducing students to it. Focusing on computational thinking as the process by which to prioritize computer science integration across subjects, and making as the activity used to engage in this process, are promising solutions. However, there is little understanding or guidance on what practices instructors should employ in maker activities to support student development of computational thinking skills and dispositions. This study aims to examine how instructors can operate within makerspace environments and activities to best contribute to their students' computational thinking.

Computer science in K-12

Computing is becoming the backbone to much of our societal infrastructure. It effects how we work, commute, shop, bank, consume entertainment, and communicate with one another. For this reason, computing jobs make up 67% of all projected new jobs in science, technology, engineering, and math (STEM) fields (U.S. Bureau of Labor Statistics, 2019) and are the leading source of new wages in the United States (Code.org, 2020).

But the power computing holds is more than just economic. Because of computing's pervasiveness, leaders in the digital arena are defining how our American and global cultures operate. Computing has far reaching impacts in our communities and our identities; it has woven itself into our systems of security, education, and democracy. To only be a consumer and not an active producer of technology is to be powerless in today's society (Bobb & Brown, 2017).

In order to be an active producer of technology, we need at least a basic understanding of computer science. Computer science is the study of computers and algorithms; this includes the design of hardware and software, how they are implemented, and that implementation's impact on society (Tucker, 2003).

Unfortunately, only 45% of high schools in the United States have computer science classes. Girls and underrepresented minorities only comprise 28% and 21% of Advanced Placement computer science exams taken nationwide, respectively. Only 40% of schools in rural communities, 35% of schools with higher (75-100%) percentages of underrepresented minority students, and 33% of schools with higher percentages (75-100%) of students receiving free and reduced lunch teach computer science (Code.org, CSTA, & ECEP, 2019).

In a quick remedy to address these disparities, many districts are implementing what Harel refers to as "pop computing", superficial coding tutorials that let students play with coding apps instead of learning to design an app (Harel, 2016). When this "computer science light" is implemented as a band-aid strategy in mostly poor and underrepresented minority schools, it creates a hierarchical system where students from those groups are being tracked as customer support or service technicians. Not only do these positions receive the least amount of pay in the tech industry, they also have the least impact on decision making, design, and development of technology that has an impact on various communities. When White and Asian middle- and upper-class students are instead provided opportunities for an expansive computer science education, the technology industry is at risk for what Kamau Bobb refers to as "technical ghettos", where poor, Black and, Hispanic students are not afforded the opportunity to use computing as a tool to advance economically or socially (as cited in Anderson, 2016).

A focus on computational thinking

Therefore, instead of simply coding, the focus for educators should be a focus on computational thinking, or CT. CT highlights the *practice*, the problem-solving skills and knowledge, needed for effective use of computing. Focusing on CT allows for the alignment of concepts from other subjects, whether that be math, science, art, language arts, or social studies, with abstractions and algorithms with and utilizing computing to provide a novel form of engagement with, and sense-making of those concepts (Grover, 2018).

Emphasizing CT bypasses the focus on coding as the primary path to having students understand computer science, when in fact it is only a small part of computer science. CT is about solving problems using computers, while coding is using a language in order to carry out these solutions. Focusing on coding ignores the deeper problem-solving techniques computer scientists use before software is developed. CT also builds upon problem solving skills that most students already have or can relate to.

In addition, an integral component of CT is challenging assumptions through the evaluation of an artifact, encouraging critical thinking. In other words, CT is inherently culturally responsive, because questioning why software is designed a certain way that might not meet the needs, or could even be a detriment, to a community of people, develops a person's sociopolitical and critical consciousness (Ladson-Billings, 1995). It is difficult for the computing industry to continue to be discriminative if everyone is invited to be critical of it.

Making as a vehicle for computational thinking

So, if CT is the thinking process needed to fully engage in our digital culture, what is the activity that helps carry out this process? Making, as it pertains to the maker movement, involves developing an idea into a product, whether that be a scarf, a lamp, or a mobile phone app. Humans have always had the desire to make, but access to the computers and the internet have removed the barriers to developing certain skills. Tools and design software have become more efficient and powerful, allowing an idea to go from concept to prototype much more easily and quickly. Sharing and collaborating on work using online resources allows anyone to build using designs they can download, with mentors and peers from around the world guiding work and provide feedback along the way (Good, 2013).

Makerspaces are informal community workspaces with access to various materials and tools where people can congregate to make and innovate. These materials and tools can include anything from yarn, sewing machines, and cardboard tubes to computers, 3D printers, and laser cutters. Though they vary in size and capacity, what makes a room a makerspace is that it houses a community established to provide the time and space for people to work, and generates a culture in which participants have permission to be creative and possibly fail. Importantly, makerspaces attract people with varying levels and areas of expertise, making them an excellent environment for understanding the interplay among various disciplines, which also makes them effective at demonstrating the versatility and power of CT. Because of their communal nature, makerspaces have the potential to engender the CT dispositions of communication and collaboration, as well as providing the support needed to be persistent and comfortable in dealing with complexity (ISTE & CSTA, 2011). Makerspaces were also born out of the hackerspace movement (Cavalcanti, 2013), so technology is a large part of many makerspaces, and it is often used to solve problems and create products. Indeed, the hacking roots of makerspaces encourage the CT ideas of remixing and iteration.

A focus on process and interest-driven work, in addition to exposure to novel STEM experiences, has helped spawn school makerspaces across the country. Because making is an iterative process that generally requires communication, it aligns with the goals of the Common Core State Standards and Next Generation Science Standards of emphasizing students being able to articulate their thinking, more than getting the right answer. Making also encourages student agency in regards to their learning, shaping how they understand a domain by utilizing new materials, tools, and methods (Herold, 2016).

That said, students are not going to develop CT skills just by entering a makerspace. Scaffolds and resources need to be put into place for specific learning objectives to be met. Instructors need to know what their students already know, as well as what they need to know, and how to get there. But these pedagogical structures need to fit within the interest-driven culture of a makerspace in order to retain its ability to engage and educate students. Research has explored how making has engaged students and encouraged the use of computing, but there is still uncertainty as to how to best structure these learning environments in a way that utilizes students' prior knowledge and interest, while promoting the development of CT skills.

Statement of the Problem

The development of CT skills and dispositions is necessary for all learners -- not only to take part in the burgeoning technology industry, but to be active contributors to how computing is shaping society. Definitional confusion and thin use of instructional design and learning frameworks has impeded the implementation of CT in the classroom. Making, as part of the maker movement, has shown some promise in engaging students and developing CT skills outside of purely programming environments. Structuring this engagement with the appropriate conceptual frameworks and their corresponding instructional tools is important in the meeting of learning objectives in making environments. What practices can teachers use to capitalize on the nature of maker learning environments in order to develop CT skills and dispositions?

Purpose of the Study

The purpose of this study was to examine the structure of makerspace environments and activities that contribute to computational thinking skills. Specifically, this study addresses promising practices that instructors can employ to support student development of CT skills and dispositions within makerspace activities. Tinkering was found to be a practice that was embedded in the learning of all of the activities, but required the structure of practices like embodiment, walkthroughs, drawing, and debugging in order to further the development of CT skills and dispositions.

II. LITERATURE REVIEW

Introduction

Being able to think computationally should be an integral part of the school day, as we prepare students to think critically about current problems, and provide them with the available tools to solve them. Many instructors are turning to the practice of making for communities of students to think through modern problems and their solutions. The maker ethos encourages participants to externalize their thoughts, making thinking visible and tangible so they can observe their processes and better communicate their ideas with others. Because making is an activity that inspires an externalization of thinking, and embraces design and digital tools, it has the potential to work as a practice for developing computational thinking skills. However, there is much work to be done to understand how we can best structure making in a learning context for developing computational thinking skills in students.

In this literature review, I explain what we know and what we do not know about computational thinking and making in K-12 education. This review should offer an understanding of context for the question "Within makerspace activities what evidence exists regarding promising practices that support youth development of CT skills and dispositions?" Before an exploration of how making can be used to teach computational thinking (CT), I first explain why scholars believe CT is important, why the educational community is currently focusing on it, and examine the definitional uncertainty that surrounds it. Once there is a better understanding of CT, I discuss the various ways CT is being explored within the K-12 classroom context. This discussion identifies the gaps in research of CT in terms of definition, instructional design, methodology, and conceptual framework. After these examples of forays into CT and making, I explain the maker movement to the reader, referencing literature that helps us define making, makers, and makerspaces, analyze prior research that demonstrates why making is important, and then look at the current research on making and learning. Once both CT and making are better understood, I explore research that highlights how making can be good for developing CT skills. Finally, I explain why using an established conceptual framework for instruction can provide the structure needed to better develop making environments to teach CT, clarifying where the research can go from here.

What is CT?

Cuny, Snyder, and Wing (2010) defined CT as "the thought processes involved in formulating problems and their solutions so that the solutions are represented in a form that can effectively be carried out by an information-processing agent," (e.g. a computer, tablet, phone, etc.). But what exactly are the "thought processes" referred to? As Snow (2014) noted, there is emerging a consensus on the characteristics of CT, but the emphasis and particulars vary depending on the context and where the definition is situated. CT has been equated with programming, algorithmic thinking, as well as with computer science itself (Alpert, 2015; Lu and Fletcher, 2009; Sengupta, Kinnebrew, Basu, Biswas, and Clark, 2013). Definitions abound, with the International Society for Technology and Education (ISTE) (2011), Computer Science Teachers Association (CSTA) (Seehorn, Carey, Fuschetto, Lee, Moix, O'Grady-Cunniff, Owens, Stephenson, and Verno, 2011), Google (n.d.), The College Board (2104), SRI (Bienkowski, Snow, Rutstein, and Grover, 2015), the British Computing Society (Csizmadia, Curzon, Dorling, Humphreys, Ng, Selby, Woollard, 2015), the K-12 Computer Science Framework (2016) and several more all weighing in. Tables 1 and 2 compare and contrast the various definitions. Table 1 displays the authors by column, with each row demonstrating where the definitions align. Table 2 displays the authors by row, demonstrating where the definitions align by column. The authors chosen were those that were most often cited.

Though the tables demonstrate how the definitions vary, there are several alignments. All CT definitions mention the ability to develop *abstractions*, which Wing saw to as the essence of CT (Wing, 2008). Abstractions are developed by identifying patterns and finding commonalities in order to create generalizations (K-12 Computer Science Framework, 2016). In order to solve a problem, we use abstraction to determine which information is necessary and which can be ignored (Lee, et al., 2013; Wing, 2008). We abstract in order to identify principles within patterns and whether elements from one instance are applicable elsewhere (Google, n.d). Wing noted that although mathematics is also is based in abstractions, computing abstractions are much richer and more complex; they do not have the easily definable algebraic properties of mathematical abstractions (e.g. you would not just add data stacks like you would add numbers) (Wing, 2008). In addition, computing abstractions are not entirely operating outside the realm of time or space -- they must work within the physical world of sufficient hard drive space or servers that are down. The other remarkable aspect for Wing regarding the power of abstraction in computing is that abstraction allows for the ability to move between different layers of information and, therefore, the ability to create complex systems (e.g. understanding what information is needed at the user layer, and how it interacts with what is needed at the developer level) (Wing, 2008).

Algorithmic thinking is also repeated in various definitions of CT. An algorithm is in itself an abstraction: step-by-step procedures for inputs to produce outputs (Wing, 2008; Buitrago Flórez, et al., 2017). Algorithms are the basis for automation, as they provide the computer with the instructions for what to do. Both Dijkstra (1974) and Knuth (1974, 1985) commented on how algorithms are the backbone of computing.

Several definitions mention the need for analysis, though some conflate analysis with evaluation, a different CT skill. Evaluation is the process of ensuring that a solution is the right fit for purpose. Analysis is an all-encompassing tool by which CT is operationalized in the classroom. Analysis involves breaking down a problem into parts (decomposition), reducing unnecessary information (abstraction), identifying processes (algorithms) and finding patterns (generalization) (Csizmadia, et al., 2015). In other words, a problem is analyzed, while a solution is evaluated. Csizmadia, et al. also differentiated evaluation and analysis as computational *thinking* vs computational *doing* (2015). Evaluation is CT in that it concentrates on performing a thought process to support learning and understanding. Meanwhile, analysis is computational doing in that it is how CT is operationalized (Csizmadia, et al., 2015).

Table 1

Computational Thinking Definitions – Authors by column

ISTE/CSTA (2011)	Google (n.d.)	College Board (2014)	SRI (2015)	Brennan & Resnick (2012)	British Computing Society (2015)	K-12 CS Framework (2016)
Formulating problems in a way that enables us to use a computer and other tools to help solve them.						Recognizing and defining computational problems
	Decomposition				Decomposition	
		Connecting computing (impacts, people and computing)	Analyze the effects of developments in computing.			
		Creating computational artifacts	Design and implement creative solutions and artifacts.			Creating computational artifacts
Representing data through abstractions such as models and simulations	Abstraction	Abstracting	Design and apply abstractions and models	Practices: Abstracting and modularizing	Abstraction	Developing and using abstractions
	Pattern recognition				Generalization (patterns)	

ISTE/CSTA (2011)	Google (n.d.)	College Board (2014)	SRI (2015)	Brennan & Resnick (2012)	British Computing Society (2015)	K-12 CS Framework (2016)
Automating solutions through algorithmic thinking (a series of ordered steps)	Algorithm design			Concepts: sequence, loops, parallelism, events, conditionals, operators	Algorithmic thinking	
					Coding	
				Data		
				Practices: Experimenting & iterating Testing & debugging		Testing and refining computational artifacts
				Practices: Reusing & remixing		Use-Modify-Create
					Evaluation	
Identifying, analyzing, and implementing possible solutions with the goal of achieving the most efficient and effective combination of steps and resources		Analyzing problems and artifacts	Analyze their computational work and the work of others.		Designing, Analyzing	

ISTE/CSTA (2011)	Google (n.d.)	College Board (2014)	SRI (2015)	Brennan & Resnick (2012)	British Computing Society (2015)	K-12 CS Framework (2016)
Generalizing and transferring this problem-solving process to a wide variety of problems					Applying	
Confidence in dealing with complexity						
Persistence in working with difficult problems						
Tolerance for ambiguity						
The ability to deal with open ended problems						
The ability to communicate & work with others to achieve a common goal or solution		Communicating Collaborating	Communicate thought processes and results Collaborate with peers on computing activities	Perspectives: Connecting		
				Perspectives: Expressing, Questioning		
					Reflecting	

Table 2

Computational Thinking Definitions – Authors by row

Author	Defining Problem	Decomposition	Impacts	Artifacts	Abstraction	Patterns	Algorithms
ISTE/CSTA (2011)	Formulating problems in a way that enables us to use a computer and other tools to help solve them.				Representing data through abstractions such as models and simulations		Automating solutions through algorithmic thinking (a series of ordered steps)
Google (n.d.)		Decomposition			Abstraction	Pattern recognition	Algorithm design
College Board (2014)			Connecting computing (impacts, people and computing)	Creating computational artifacts	Abstracting		
SRI (2015)			Analyze the effects of developments in computing.	Design and implement creative solutions and artifacts.	Design and apply abstractions and models		
Brennan & Resnick (2012)					Practices: Abstracting and modularizing		Concepts: sequence, loops, parallelism, events, conditionals, operators
British Computing Society (2015)		Decomposition			Abstraction	Generalization (patterns)	Algorithmic thinking
K-12 CS Framework (2016)	Recognizing and defining computational problems			Creating computational artifacts	Developing and using abstractions		

Author	Coding	Data	Iteration	Remixing	Evaluation	Analysis	Application
ISTE/CSTA (2011)						Identifying, analyzing, and implementing possible solutions with the goal of achieving the most efficient and effective combination of steps and resources	Generalizing and transferring this problem-solving process to a wide variety of problems
Google (n.d.)							
College Board (2014)						Analyzing problems and artifacts	
SRI (2015)						Analyze their computational work and the work of others.	
Brennan & Resnick (2012)		Data	Practices: Experimenting & iterating Testing & debugging	Practices: Reusing & remixing			
British Computing Society (2015)	Coding				Evaluation	Designing, Analyzing	Applying
K-12 CS Framework (2016)			Testing and refining computational artifacts	Use-Modify-Create			

Author	Confidence in Complexity	Persistence	Ambiguity	Open Ended Problems	Communication/ Collaboration	Expression/ Questioning	Reflection
ISTE/CSTA (2011)	Confidence in dealing with complexity	Persistence in working with difficult problems	Tolerance for ambiguity	The ability to deal with open ended problems	The ability to communicate and work with others to achieve a common goal or solution		
Google (n.d.)							
College Board (2014)					Communicating Collaborating		
SRI (2015)					Communicate thought processes and results Collaborate with peers on computing activities		
Brennan & Resnick (2012)					Perspectives: Connecting	Perspectives: Expressing, Questioning	
British Computing Society (2015)							Reflecting
K-12 CS Framework (2016)							

Hu also pointed out in the tension between CT and computational doing (2011) but stated that *doing* computation is necessary to improve one's *thinking* computationally (Inhelder and Piaget, 1958; Papert, 1980; Brown, Collins, and Duguid, 1989). Additionally, CT is difficult to see, and therefore verify, but easier to see and assess computational doing. Researchers responded to these tensions by naming CT "patterns" (Basawapatna, Koh, Repenning, Webb., and Marshall, 2011), CT "concepts" and "perspectives" (Brennan and Resnick, 2012), and CT "practices" (Bienkowski, 2015; The College Board, 2016; Brennan and Resnick, 2012) (see Tables 1 and 2). The K-12 CS Framework identifies CT as a thread that runs through computer science practices (K-12 Computer Science Framework, 2016).

Interestingly, Denning challenged the computer sciene community with the notion that "thinking" or "doing" are not the appropriate goals. Instead, Denning argues the educational community should push for students to better understand how to *design* computations. He frames this argument through the transformation of roles in the scientific community. Previous to the introduction of computing, scientists would experiment, gathering data to determine the validity of a hypothesis; or theorize, developing models using mathematics to explain what is known and using those models to predict what would happen with what is not known. Computers introduced new ways to advance science through design, using tools like simulation to explore beyond the limits of experimentation and theory. For instance, experimentation would develop test flights for aircraft, and theorization would develop mathematical models based on how those experiments worked. But *computational design* could develop a simulation of airflows around the yet-to-be engineered wing of a simulated aircraft. Denning emphasizes that design is not just being able to program or develop mathematicval models, but a skill set focused on the needs of the end user (Denning, 2017).

Computational design corresponds with the definitions put forth by Brennan and Resnick (2012) and the K-12 CS Framework (2016). They both value iterating, testing, and refining, key components of design. They also value the practice of reusing and remixing, or as Lee, et al., formalized it “Use-Modify-Create” (2011). Designers constantly iterate on an object or idea, within a set of constraints, in order to accomplish a goal (Ralph and Wand, 2009). For example, to create a video game that simulates tennis, one would try out algorithms until landing on those that best simulate gravity and spin on a ball, or design different digital rackets to until arriving at one that is similar to what the pros use.

Expressing, questioning (Brennan and Resnick, 2012) and reflecting (Csizmadia, et al., 2015) are also of value to design, as they encourage the designer to better understand goals and the requirements to achieve those goals. ISTE/CSTA, College Board, SRI, and Brennan and Resnick also reference communication and/or collaboration, elements that are vital to effective design. Computational thinkers need to be able to explain the design of an artifact and appropriately document its functionality and features to colleagues and users (Bienkowski, et al., 2015).

Expressing, reflecting, and questioning encourage learners to harness the power of computing in order to become full participants in our developing digital culture. CT could provide learners with what McWilliams refers to as “critical computational literacy” (2010), hearkening back to DiSessa’s ideas of the infrastructural nature of computing. In addition, the “critical” element emphasizes the computational thinker’s ability to maintain their own identity and their own communities’ needs, without having to exist within constraints designed by corporate or private interests that do not reflect their concerns (Jenkins, 2006). It is a "technological fluency" that Papert notes that can allow students to be aware of their ability to solve problems generated by themselves or others (Papert, 2000). Barba (2016) has argued that this is where Wing and the CS community have gone astray from Papert's initial coining of CT. In the article where Papert first used CT, he discussed what he calls the "Power Principle" in that, as humans, we develop our knowledge and understanding through use, i.e., a kind of tinkering with ideas. He criticized the present-day school system for inverting that process by assuming students need to acquire deep understanding before they can use their knowledge (Papert, 1996). Papert's vision for CT, Barba argues, was not in being able to develop abstractions or algorithms, but rather in the understanding that ideas give us the power to act (Barba, 1996).

Keeping these diverse notions of CT in mind, a definition that is operational (and is the definition that will be used in this study) combines the definitions of Selby and Woollard (2014) and ISTE and CSTA (2011). Selby and Woollard define CT as a mostly product-oriented problem-solving thought process that demonstrates the ability to think (1) in abstractions, (2) in terms of decomposition, (3) algorithmically, (4) in terms of evaluations, and (5) in generalizations, or patterns. The strength of this definition lies in that it only incorporates terms for which there is a consensus in the literature, making it widely accepted (Tables 1 and 2), as well as being composed of terms that are well defined across disciplines, making it flexible for classrooms. Csizmadia, et al. (2015) used this definition as the basis for the British Computing Society's teachers' guide.

An emphasis on agency in problem solving (Papert, 2000; Brennan & Resnick, 2012) makes the inclusion of dispositions into a definition of CT crucial (Stephenson & Malyn-Smith, 2016). The idea of dispositions comes from research on vocational education and career development which focuses on the personal qualities needed for employment (Kenworthy & Kielstra, 2015), which usually include competencies like being self-motivated and having the ability to work with people of different ages and cultures. In addition, these “habits of mind” could encourage students to use their understanding of computing to develop their sociopolitical consciousness and more fully participate as members of their communities (Bobb & Brown, 2017).

While the set of CT skills ISTE and CSTA (2011) put forward can be difficult for teachers to integrate into the classroom, the set of dispositions they propose students practice and internalize while learning about CT align with occupational analyses of CT-enabled STEM professionals (Malyn-Smith & Lee, 2012). These dispositions include (1) confidence in dealing with complexity, (2) persistence in working with difficult problems, (3) the ability to handle ambiguity, (4) the ability to deal with open-ended problems, and (5) being able to communicate and collaborate with others to achieve a common goal or solution. While these are the type of “soft” or “non-cognitive” skills that are certainly necessary in any field (Dweck, Walton, and Cohen, 2011; Farrington, et al., 2012), they are especially necessary when dealing with the messy layers of abstraction that Wing described (2008), as computational thinkers deal with both the concrete and indefinite, immense swaths of data and little bandwidth, unreliable users and hardware that is both unbending and able to break down.

Why is CT important?

It was just over fifty years ago that the first electronic computers were developed (Margolis, Holme, Estrella, Goode, & Nao, 2017), but their impact in this short amount of time is indisputable. Computer science and information technologies are reframing how nearly every field of study, organization, and individual works (Hogan and Roberts, 2015). The reason for this growth is clear: computing allows us to expand our thinking at a scale that would be impossible with our brains alone. It is humanly impossible to organize, prioritize, or calculate at the rate that a computer can. But as amazing as computers are, we can only get out of them what we put in them. We need to fully understand what inputs will give us the outputs we need.

In 2006, Jeannette Wing published a short, but seminal, article in the Association for Computing Machinery's journal *Communications of the ACM* that challenged the education community to go beyond reading, writing, and arithmetic, and make CT another integral part of every child's education. She argued that CT was vital because it provides methods for addressing our current problems across all fields, building on the power of computing, as well as understanding the limits of processes to solve problems (Wing, 2006).

Similarly, Perlis contended that all undergraduates should learn to program, because it teaches how to construct and analyze processes (Perlis, 1962). Papert, who first used the term "computational thinking" (1996), took it a step further, stating all computers should be accessible to all children as a way to provide agency to their learning and expression (Papert, 1980). He saw the way that we organized information that we input into computers could be a way to help us better operationalize our own thinking. To Papert, the computer bridges the gap between Piaget's conceptualization of concrete thinking (able to reason logically only when reasoning can be applied to concrete and specific examples) and formal thinking (advancing from logical reasoning with concrete examples to abstract examples) (Papert, 1980; Inhelder and Piaget, 1958).

Sheil used the related term "procedural reasoning" and described it as a process one uses to determine the effect of a certain set of instructions, being able to understand when these rules are meant only for specific instances, and when they can be abstracted to be used generally (Sheil, 1983). Sheil saw procedural reasoning as a new way of thinking, revolutionizing human thought in the way Newtonian mechanics did.

DiSessa's use of "computational literacy" (2000) focused on deconstructing problems in a way that a computer can further understanding and help deliver a solution. DiSessa described how literacy (as the term is commonly used today) is *infrastructural*, meaning that literacy is a driving force within the educational process, not just a learning objective. Literacy and education are so enmeshed, that it is impossible to separate out one without the other. To be clear, *computational literacy* should not be confused with *computer literacy* or *digital literacy*, which in today's education-speak generally indicates one is comfortable casually using technological devices and applications. DiSessa referred to a much deeper form of knowledge than simply being able to operate a mouse or understand how to download an app to an iPhone. For this reason, he interchanged *literacy* for *material intelligence*. Material intelligence is a type of distributed cognition, extending the mind to other tools and resources like the computer, as an addition to the individual's internal intelligence.

Material intelligence is the ability to use tools (computers) to increase our intelligence and skills. Like Sheil, DiSessa described computational literacy as a transformational effect on human understanding, using algebra's effect on the understanding of the physics of motion as an example. Computational literacy doesn't just make learning easier, but such literacy offers an entirely new dimensions of what we can learn. Fundamental principles become obvious, so the learner can easily build upon them to develop new discoveries. The learner is making sense of the world around them using a computer, and this interaction between the learner and the computer further informs their understanding of the world.

In the last decade the concept of CT has begun taken hold. The International Society for Technology in Education (ISTE) and the Computer Science Teachers Association (CSTA) have developed several resources for K12 teachers to integrate CT into their lessons (2011). The National Research Council convened two workshops with leading academics in computer science and education around "the scope and nature of computational thinking" and "the pedagogical aspects of computational thinking" (2011, 2015). Most notably, the recently released Next Generation Science Standards lists "using mathematics and computational thinking" as one of the Science and Engineering Practices that cut across the disciplinary core ideas (NGSS, 2013).

CT has begun to influence a wide breadth of areas of study, as evidenced by the growth of fields like algorithmic medicine, computational anthropology, computational economics, computational finance, data journalism, and computational law (Wing, 2010). For example, extracting game day data and being able to use it with identified patterns of speech allows some sports journalism to be automated, delivering stories about minor team sports to their fans (without them realizing the story was written by a computer (Allen, Templon, McNally, Birnbaum, and Hammond, 2010). By collecting data through mobile phones and then analyzing patterns in these datasets, anthropologists are able to identify local inhabitants of a city versus tourists and how they move differently within a city (Yang, Lian, Yuan, Xie, Rui, and Zhou, 2017).

The scientific community that has also noticeably embraced computational thinking. Alongside experimentation and theory, computing is now widely considered the "third pillar" of scientific inquiry, allowing scientists to develop, test, and refine models that would be impossible in the laboratory, while efficiently managing large amounts volumes of data (President's Information Technology Advisory Committee, 2005).

Buitrago Flórez, Casallas, Hernández, Reyes, Restrepo, and Danies (2017) list the various scientific areas where computing is transforming how work is done. Computing is used for metabolic pathway reconstruction (biology), molecular dynamics simulations, chemical pathways modeling (chemistry), physical interaction in biomolecules, optical performance simulations (physics), and physiology performance simulations (medicine), to name only a few.

Karp (2008) details how much of the information studied in the sciences, and specifically in biology, can be seen through a "computational lens" where many natural processes can be represented and described through digital data and algorithms. Processes such as the regulation of protein production, metabolism and embryonic development, and the phase transitions of physical systems can all be described, and altered, through algorithms.

CT in learning spaces

Because of its modern-day usefulness, CT is a problem-solving process that should be integrated and applied across subjects (Barr and Stephenson, 2011), however there is not a very clear path for educators to accomplish that integration. This is in part due to the definitional confusion (Grover and Pea, 2013, p.38) that has plagued CT, but also because of the lack of national direction. The Next Generation Science Standards (NGSS) includes "using mathematics and computational thinking" as one of its eight scientific practices (NGSS Lead States, 2013) and the math standards in the Common Core State Standards (CCSS) include the need for using technology "to explore and deepen their understanding of the concepts" (National Governors Association Center for Best Practices, Council of Chief State School Officers, 2010, p. 7), but the direction in these guides for teacher implementation is vague. Up until October of 2016, there has not been a national computer science framework that included CT for K-12 to begin developing standards or curriculum (K–12 Computer Science Framework, 2016).

CT integration, specifically in the STEM fields, offer three important potential outcomes (Weintrop, et al., 2015): First, bringing CT into STEM classrooms gives learners a clearer picture of what the profession actually entails and better prepares them for careers in these disciplines. Second, the use of computation can deepen understanding of content, and on the flip side, this content can be useful to explore computing. Finally, by embedding CT in classrooms, its concepts are exposed to a wider audience, including females and underrepresented minorities.

In order to address the gap in best instructional practices for CT integration, researchers have implemented and studied several interventions to cultivate CT skills. While these studies vary in their scope and structure, the research community still struggles to develop interventions and studies that can best prepare students with CT skills.

CT as programming

The fuzziness surrounding the definition of CT has been problematic for researchers and educators in determining best practices for preparing students. For instance, in a mixed method study on integrating CT into middle school science lessons, Alpert (2015) interchanged CT with "programming" and measured CT by examining science fair project presentations done in Scratch - a free visual programming environment for youth to create stories, games, and simulations and share what they made in an online community (Brennan and Resnick, 2012). To analyze what CT the students employed, Alpert used "Scrape", a tool that allows a quantitative analysis of the programming blocks used in Scratch projects to determine which programming tools are used and how often. Most of the projects were built using basic programming and only about a third of all of the projects used of advanced concepts such as conditional expressions, loops, and forever statements. Case studies were made of four of these projects, and analysis included the "scientific inquiry" that each project referenced, in conjunction with the blocks that were used with the purpose of demonstrating transfer (Bransford, Brown, and Cocking, 2000). Nevertheless, it remained unclear how the transfer could be demonstrated by connecting the use of programming blocks to scientific inquiry. Additionally, by just examining the coding blocks, there is no analysis of the thinking process that went into developing the projects.

Denner, Werner, and Ortiz (2011) have a more faceted approach to analyzing CT by what students have programmed. In the study, 59 sixth grade girls participated in after-school game programming sessions using Stagecast Creator, developing several games over the school year. Once the students had completed their games, each game was coded for three key competencies for CT: programming, documenting and understanding software, and designing for usability. For each competency, there were 24 subcategories were coded.

While a more nuanced approach to looking for CT in a final programmed project, the authors themselves point out the limitation in their study in that the games did not reflect the students' capacity, but only the results of what they did. The researchers pointed out how the more complex aspects of computing (e.g. using global variables or decomposing actions into series of rules), as well as documentation and design, seem to evade the students. Students' motivation for creating a game was tempered by what they felt they could achieve. By not examining the student's thinking process, including the starts and stops of their ideation, we do not know how the students approached the problems. CT cannot be determined only by looking at final products, but by the struggle that brought the student to that final product.

As Buitrago Flórez et al. (2017) note, programming is a vehicle by which students can develop the CT skills of algorithmic thinking, problem solving, logic, and debugging. A number of studies describe introducing students to CT, and/or measuring CT mostly through programming skills (Wolz, Stone, Pulimood, and Pearson, 2011; Sengupta, et al., 2013; Bers, Flannery, Kazakoff, and Sullivan, 2014; Grover, Pea, and Cooper, 2015; Tarkan, S., Sazawal, Druin, Golub, Bonsignore, Walsh, and Atrash, 2010; Touretzky, Marghitu, Ludi, Bernstein, and Ni, 2013). But in order for this to be accomplished, programming education should have the development of CT skills as its objective, making it explicit throughout instruction; programming should not just be framed as a way to communicate with computers.

CT as model and simulation

Several studies align with Moursund's position that the underlying idea in CT is developing models and simulations (Moursund, n.d.; Malyn-Smith and Lee, 2012; Lee, 2011; Lee, Martin, Denner, Coulter, Allan, Erickson, Malyn-Smith, Werner, 2011; Marshall, 2011; Alpert, 2015; Sengupta, Kinnebrew, Basu, Biswas, Clark, 2013; Dwyer, Boe, Hill, Franklin, and Harlow, 2013; Lee, Martin, and Apone, 2014; Farris and Sengupta, 2014). The Scalable Game Design team at University of Colorado, Boulder, used an agent-based model, AgentSheets (Basawapatna, et al., 2011). In agent-based modeling, individuals (which could be people, animals, cells, etc.) and their interaction with one another and their environment are explicitly represented. The individuals can be performing a task and that task performance can be observed (Shank, 2010). The team looked at whether students could use what they learned from programming games to create science simulations, hoping to tap into student interest in video games. They identified Computational Thinking Patterns (e.g. Generation, Absorption, Collision, Transportation, etc.), or CTPs, which are programming patterns students learn when they create games (Basawapatna, et al., 2011). For instance, the generation pattern involves a digital object, or agent, to create another agent. Simulation examples include raindrops emanating from clouds or animals breeding to create new animals. For the collision pattern, two agents physically collide, like cars atoms crashing into one another to create new elements.

The team used these CTPs in conjunction with their visual-programming environment, so that by designing games, students would understand what is needed to create models and simulations. Students were assessed with a quiz that had 8 questions, 7 of which were video of real-life actions that are similar to in-game CTPs (e.g., a sled transporting a person that collides with another person) and another which was text about a predator/prey simulation. Students then had to identify the appropriate CTP they observe in each scenario.

While an innovative measure of CT, the focus on games, modeling, and simulations is limiting, as it does not demonstrate the breadth of what CT can accomplish not just for solving problems, but for generating questions to solve. CT has the potential to solve problems for how we interact, what we wear, or how safe we are and feel. While modeling and simulation are an important component to CT, students have to be clear about how it can empower them to address issues in their own lives, or its import is lost.

Lee's (2011) study on Project GUTS (Growing Up Thinking Scientifically) also measured CT in the context of using modeling and simulation programs for middle school, but took into consideration having students understand how these tools could be used in their own lives. For the first four weeks, students were introduced to complex and Starlogo TNG, another agent-based modeling tool. They used basic programming constructs to learn computer science concepts to use in Starlogo TNG like declaring, creating global and local variables, and handling collisions between agents. They were also introduced to a variety of complex systems concepts like emergent patterns, non-linearity, participatory simulations, and simple models. Students then did a six-week topical unit on epidemics and ecosystems that included two hands-on participatory simulations, where they learned how to create and experiment with a base model. Students then had an opportunity to customize the example model or create a new model from scratch to reflect a local phenomenon of their choice, whether it be in their school or community. Students used their model as an adaptive simulation to run different scenarios and then they collected and analyzed the generated data.

After their experience in the program, Lee (2011) interviewed the twenty participants, asking them to think-aloud in response to a scenario where they had to redesign a mall after a fire there resulted in several injuries. Of the 16 individuals who described abstractions, 10 described automation; of the 10 that described both abstractions and automation, 6 performed analytic reasoning. Based on this evolution, Lee suggested an indication of a type of progression, or learning trajectory, that moves students from abstraction to automation to analysis.

It would be interesting to know more about these students' experience with the six-week units and how they related to them personally. While Project GUTS had the creative approach of including the students' own environment (e.g., schools and malls) to create models, it appears the project did so only at the end, after several weeks of learning about programming and modeling, separate from their own experiences and knowledge. Would more students have a better grasp on abstractions, automations, and analysis if they were able to incorporate their own understandings of their environment into the work they were doing, thereby making it more relevant for them?

Instructional design and CT

This highlights another issue with several studies, namely that the instructional design of the intervention does not always align with the goal of developing CT. Touretzky, Marghitu, Ludi, Bernstein, and Ni (2013) developed a three-stage model of computing instruction. They began the students on a simple icon-based programming tool in which users construct programs using a sequence of cards (Kodu). They then moved them onto to a drag-and-drop object-oriented educational programming environment (Alice) and finally, a visually based language for robotics (Lego NXT-G). The researchers believed that mastery of CT requires a deep as well as abstract understanding, where a deep understanding recognizes when a task requires the use of a programing tool like conditionals, or looping, or parallelism, while an abstract understanding recognizes the tool used regardless of the computing environment.

While being ambitious to accomplish this understanding in a five-day program, a larger problem was the lack of context for the work. Learning these computing environments without a larger objective or without a thread that ran throughout could make it difficult for a student to clearly understand why these skills might be useful to a person outside of a programming environment. Learning activities and environments need to be connected to students' lives to not seem arbitrary and incidental; they need to demonstrate to students why they should be engaged. The researchers only measured whether students enjoyed the camp and whether they would continue with the programming environments on their own, but it would be challenging to achieve their goal of developing deep or abstract understanding participating in a program design devoid of connection or context.

Kazimoglu, Kiernan, Bacon and MacKinnon (2012) placed their CT learning context in a game environment. The game they have developed, "Program Your Robot" was meant to help students practice various CT skills and introductory programming constructs. In the game, players control a robot by typing in commands. Each level has a teleporter the robot has to reach, and players need to design an algorithm to help the robot to reach it. While innovative, having a pre-established goal that has no connection to a student's life is problematic. The Power Principle is ignored; the student is working in a structured environment that leads them to the "right" answer, ignoring the concept of using CT to develop ideas and do something with them (Papert, 1996).

Wilkerson-Jerde (2013), inspired by Papert's ideas, built her research on CT around the ideas of constructionism by developing a "constructionist computer-supported collaborative environment" to give students the opportunity to use computational principles when exploring mathematics. The goal of the study was to explore whether if students could better understand the underlying properties of objects within a domain by to identifying patterns within their work as a community. Specifically, the author was interested in whether students would be able to connect the computational ideas they used to create their fractals with the patterns and themes that were generated from the community of their classmates. Wilkerson-Jerde analyzed students using "the Categorizer", an interactive gallery that allows students to share digital artifacts they have created in an online space. Learners then analyzed the set of artifacts, and sorted them into user-defined categories. Students created artifacts in the "Construction Interface" of the Categorizer, which allowed them to create an artifact by exporting a visual representation of a set of rules. These artifacts were then uploaded for display to a shared "Categorization Gallery" along with what their peers created. To determine the Categorizer's ability to support students' understanding of the relationship between computational rules, like pattern recognition, and ways of classifying objects that represent a particular phenomenon, Wilkerson-Jerde studied the implementation of the Categorizer tool with a lesson on fractals in three middle school mathematics classrooms.

The results indicated that while students were able to create categorization systems that potentially connected rules and organizational themes, only a few students explicitly made these connections. The author indicated that this could be due to the students lacking motivation to push themes beyond the fractals' aesthetics (Wilkerson-Jerde, 2013). However, the inability to make connections might also be attributed to the fact that the students were asked to grasp an understanding of fractals, the interface, and be able to generate rules and establish patterns in a matter of two one-hour sessions – a high cognitive load (Sweller, 1988) that would tax working memory in a short period of time for fairly young learners. More time with less complicated content may have generated more, albeit simpler, connections. More importantly, it appears that much of this understanding happened without much scaffolding. Primarily, students were just given a series of prompts to complete over the class session. The self-similarity and non-integer dimension properties of fractals are complex concepts to grasp, much less notice patterns about them. This demonstrates some of the weakness that cognitivists have aimed at constructionists, namely that putting students suddenly into complex situations without appropriate scaffolding will result in learning that is slower and more frustrating, and not necessarily better. (Anderson, Reder, and Simon, 2000; Guzdial, 2017, para. 12).

Measuring the acquisition of CT

Wolz, Stone, Pearson, Pulimood, and Switzer (2011) take a different and innovative tack outside the usual CT confines of STEM, exploring the similarities between CT and expository writing. In their project, 7th and 8th graders and their teachers created an online newsmagazine in either a summer, classroom, or afterschool experience. In all programs, the students learned about interactive journalism as they published their first online magazine using text, video, graphics, and animations developed in Scratch. The authors defined CT as "a mode of problem solving that emphasizes the processes necessary to express a computing-intensive solution in a structured, dynamic way" and as a required skill set that "includes how to define and analyze a problem and implement and test the solution" (p. 9:2). While the definition is somewhat vague, more problematic is the vagueness of the description of how CT was exactly infused in the curriculum. There is no description of how activities were structured to assist in problem-solving or solution evaluation and CT was never defined explicitly for the study parameters or for the participants in the study. Results from the summer and classroom programs showed a significant increase in confidence in both teacher and student computing abilities, however much might have been gained from thinking through what are the specific components of CT and how using those problem-solving skills could be transformative for students studying journalism or writing.

Brady, Orton, Weintrop, Anton, Rodriguez, and Wilensky (2015) study another intervention intended to provide CT, but are similarly as vague about CT. The Computational Thinking for Girls (CT4G) initiative was a yearlong course offering 7th to 12th grade girls a wide range of activities to introduce them to CT in 100-min sessions once a week. Throughout the year, students were engaged in social computing, modeling and simulation, and various making activities that included e-textiles, Arduinos (a microcontroller), and 3D printing. The study included pre- and post-surveys to demonstrate a positive trend in the girls' perceptions of computing and an increase in their interest in careers as they relate to CT and CS. The authors also collected student interviews, field notes, and student artifacts.

The authors do not define CT, they only reference Papert and Wing's definitions. While they state that they have aligned the program's goals with other CT initiatives like the NGSS, they do not state what goals. They also mention that throughout the class, facilitators made connections between CT and CS to the work the girls were doing, but these specific connections are not made explicit.

In one activity, the students used a circuit simulator for modeling the network communications that are utilized for instant messaging. In this activity, the authors indicate that they "encounter foundational CS ideas including abstraction, data representation, and dimensions of human–computer interaction" (Brady, et al., 2015, p.4). However, there is no explanation of how these ideas were encountered, or connected to the circuit simulation to clarify the CT that was being used. From the artifacts the students created, the authors observed how situating computational ideas in personally relevant contexts can engage students with computing. This is an important observation, but they stopped short of detailing the CT skills the students demonstrate in the ideation and creation of their final product. They also described students altering the code in a model, but again failed to describe the CT skills that the students employed. While the reporting of change in interest in CT and CS subject matter and careers is clearly important in the engagement of students, these studies and others (Hambrusch, Hoffmann, Korb, Haugan, and Hosking, 2009; Basawapatna, Koh, Repenning, 2010; Wolz, Stone, Pulimood, and Pearson, 2011; Touretzky, Marghitu, Ludi, Bernstein, and Ni, 2013; Weintrop, Beheshti, Horn, Orton, Trouille, Jona, Wilensky, 2015) missed opportunities with intensive interventions to develop CT skills or to report on CT skill development.

Grover, Pea, and Cooper (2015) did capitalize on their interventions by embedding assessments of CT with "Foundations for Advancing Computational Thinking" (FACT), an introductory computer science course for middle school. The course was developed to prepare middle school students with problem solving using algorithms. The study consisted of two iterations of design-based research and measured student understanding of designing algorithms, and whether this learning would help them adapt Scratch block-based programming to text-based languages. The six- to seven-week long online course included various topics, including the pervasiveness of computing; the discipline of computer science; algorithms; problem decomposition; and abstraction. The sessions involved working through examples to computational problems using pseudo-code ("semi-English" instructions that represent what would be written in a programming language) or in Scratch, which was the programming environment used in FACT. Importantly, instead of focusing on attitudes toward CS, the study's results demonstrated that students achieved gains in algorithmic thinking skills, their ability to transfer their conceptual understanding of computer programming tools from a block-based to a text-based programming context, and their understanding of CS in general.

However, these results are tempered by methodological limitations, that the authors acknowledge, but that do not allow the study to address the large problem the CS education research community faces, which is broadening participation with underrepresented populations like African-Americans, Latinos, and females. As it was mainly done online, the surveys, quizzes and tests required students to read a lot of text. For English Language students and those struggling with fluency, this was a challenge. Additionally, the studies were conducted in an elective class, indicating a self-selection bias among learners who were already interested and motivated. The self-selection may have resulted in a large gender disparity in the sample size (41 males vs 13 females), which along with the socio-economic status of the school community, threatens the external validity of the study and its ability to be broadly generalized (Grover, Pea, and Cooper, 2015). Furthermore, the researchers did not mention the ethnic/racial makeup of the class and how that breaks down with both the data on affect, learning gains, and transfer of knowledge abilities. Because of the ethnic/racial and gender disparity in computer science, this information is vital in doing CT studies in order to better understand if the necessary supports and environment are being provided students of color and females to encourage their inclusion in CS and its related fields.

Learning frameworks in the study of CT

Many of these problems with current empirical studies on CT, whether they be an issue of definition, instructional design, or methodology are closely tied to the matter that the work is not based on learning frameworks that emphasize the various aspects of expertise development. If our goal is to engender CT skills for every child, then, as education researchers, we must treat CT as an academic domain that should be strategically scaffolded, and use learning theories on which to build and support its study. Because the work on CT is so new, and the definition still so amorphous, it is vitally important to state a proper theoretical framework provides the structure to define how the researchers philosophically, epistemologically, methodologically, and analytically approach the work (Grant and Onsaloo, 2014). This provides the rest of the research community with an understanding of where they are coming from and how to best interpret the work and results.

Brennan and Resnick's work on studying and assessing CT is thorough and thought provoking, as they furthered the conversation on what CT actually is (Brennan and Resnick, 2012). However, despite one knowing that Brennan and Resnick have in the past based much of their work on constructionism (Kafai and Resnick, 1996; Resnick and Ocko, 1991; Resnick, 1991; Bruckman and Resnick, 1995; Resnick, 1996; Rusk, Resnick, and Cooke, 2009; Brennan, 2014; Brennan, 2015; Brennan, Blum-Smith, and Yurkofsky, 2015), there is no indication in this work of what framework helped construct their definition of CT. This information would situate our understanding of not just their definition, but the methods by which they assessed student work in CT.

Their definition of CT includes concepts that are aligned with programming concepts, practices that are aligned with design practices, and perspectives that are aligned with motivational and culturally responsive practices. The elements of the framework also appear discrete, not seeing CT so much as a problem-solving *process*, but as a number of separate elements.

They assessed student work using their definitional framework, without a theoretical framework, and it is unclear why they decided on the methods they used. Their first method of analyzing student work by running it through "Scrape" which, as I noted with the Alpert study (2015), is limited by what they themselves also state: The approach is entirely focused on results, without a focus on process and the CT practices that might have been used. As a counter to this, they used a method of interviewing students on their process, which proved to be time-consuming, with students not having the recall, self-awareness, or humility to speak about their challenges. Finally, the authors constructed design scenarios in which the student explained what a selected project did, what could be added to it, and how to debug it and iterate on it. This also proved to be time-consuming, as well as detached from the student's personal motivations and interests. With a theoretical framework as a foundation, these different assessment approaches would have structure that aligned with the learning goals.

Sengupta, Kinnebrew, Basu, Biswas, and Clark (2013) developed a conceptual framework that is based on a "learning by design" approach to science learning through CT. In this model, students (1) learn the basics of a science phenomenon, (2) model the actors and processes of the phenomenon by programming in a visual agent-based environment, (3) simulate and study how model behaves, and (4) apply the model and the science concepts to solve a problem (Sengupta, et al, 2013). This framework guided the researchers in developing an agent-based modeling environment composed of three modules that mapped onto the cycle described in the conceptual framework: The Construction World, the visual programming interface for the student to program a model; the Enactment World, which plays out the phenomenon modeled by the student in the Construction World; and the Envisionment World, which allows students to set up experiments in order to analyze their models and to compare their model against an exemplar.

Diving deeper into how CT is developed through modeling and simulation, Farris and Sengupta (2014) presented a theoretical framework for analyzing and understanding the role of points of view in the development of what they refer to as "collaborative computational thinking", also in the context of using agent-based programming and modeling. Built around concepts developed by Greeno and van de Sande (2007), the authors adopt the view that in a collaborative setting, each individual's understanding develops into a new joint understanding shaped by each individual point of view.

The students in this study worked with another agent-based modeling and visual programming environment (ViMAP) that was specifically designed to support learning in kinematics (Sengupta, Farris, and Wright, 2012). The researchers focused on the interactions of two10-year-old boys as they generated agent-based computational models for one graph that showed distance versus time and another that showed speed versus time. They described the interactions between the two boys as they negotiate between their different points of view of what the graphs were showing and how to represent that in a model, explaining how the "perspectival understanding" framework they were using developed the students' CT. The authors argued that when students engaged in programming collaboratively, they simultaneously developed their CT and physics learning by discussing it from various points of view. They saw the convergence of the students' disparate perspectives as being tied to the learner's "computational doing", generated through both dialogue and the programming they did together (2014).

While the learning-by-design conceptual framework and the perspectival understanding theoretical framework are valid ways of framing the work the researchers have done, they are limiting and do not demonstrate the full potential of CT. By focusing on frameworks that only highlight how modeling and simulation develop CT skills in students, these studies do not allow for how students can explore their own agency and how computing can work as a tool for empowerment. As with other studies on CT that use frameworks, student interest and agency is sidelined with respect to the learning goals, instead of working alongside the learning goals (Wilkerson-Jerde, 2013; Grover, Pea, and Cooper, 2015).

Though the research community has much more empirical work to do around CT, the work that has been done offer the beginning of exciting possibilities. Despite their range, they all have students developing CT skills through the creation of artifacts, and doing so in community, whether in person or online. The employment of game design, simulations, interactive writing, storytelling, and robotics are all examples of students using artifacts to externalize and demonstrate their CT learning and vision, or what is referred to as "making".

What are making, makers, and makerspaces?

Simultaneous to the growth of interest in CT, the maker movement began to take hold, as well. In 2005, Dale Dougherty began Make: Magazine, as a response to the growth of the Do-It-Yourself (DIY) community (Heather, 2016) and a year later, the magazine sponsored the first Maker Faire (a demonstration of hands on and DIY projects sponsored by Make Magazine) in the San Mateo Fairgrounds (Branwyn, 2015). The term *maker movement* is in reference to the increasing popularity of groups of people coming together (in digital and physical forums) to share the process and final product of a creative endeavor. It is comprised of "*making* as a set of activities, *makerspaces* as communities of practice, and *makers* as identities" (Halverson and Sheridan, 2014, p. 496).

Defining the term "makerspace" is difficult, as makerspaces are varied in composition and culture. They are not a specific kind of physical environment with particular equipment, but the availability of space and time with a community of other makers (Martinez and Stager, 2013). Martinez and Stager break down "ways of knowing" that can occur in a makerspace into three processes: *making* refers to learning through construction; *tinkering* is mindset that solves problems through play and discovery; and e*ngineering* is both a design and science that allow us to actively interact with our environment (2013). Vossoughi and Bevan describe engineering, as defined in the Next Generation Science Standards, as a process that involves both making and tinkering (2014). An element integral to these ways of knowing is the design process (Sheridan, Halverson, Litts, Brahms, Jacobs-Priebe, and Owens, 2014; Litts, 2015). The design process means different things in different fields, as Litts (2015) demonstrated when comparing the processes in engineering, the arts, and architecture. However, Litts showed they overlapped in that their processes all involved focusing on a problem, generating possible solutions, iterating, and communicating (Litts, 2015).

Why is making important?

Proponents of the maker movement in education often use language that emphasize the maker movement's ability to upheave the status quo, referring to it as "a learning revolution" (Kurti, Kurti, and Fleming, 2014) that can "transform" education (Martinez and Stager, 2013b; West-Puckett, 2013). Much of this sense of uprising can be attributed to the learning theories the maker movement is built upon. Freire's critical pedagogy situated education as a tool to question assumptions of how educational or economic systems operate (Freire, 1974; Flores, 2016). Part of this questioning includes the "banking" method regularly employed by formal education in which the teacher makes "deposits" into the empty heads of the student. Freire considered this an oppressive structure that should be overthrown -- students and teachers should work alongside one another to make meaning and develop ideas (Freire, 1974). Freire's ideas are part of the "maker mindset" that encourage collaborative learning and facilitation, instead of direct instruction (Flores, 2016). Freire built upon Deweyan constructivism, in that students should be encouraged to think for themselves and be motivated to articulate those thoughts (Dewey, 1916), but he took it further in order to give students agency regardless of established systems.

Like Freire, Piaget believed also that children are not “empty vessels” for knowledge to be poured into. The learner actively constructs knowledge through experience; the schemas, or mental representations, of that learning are constantly being revised and reconstructed. As children interact with the world around them, these schemas change or are deconstructed and reconstructed to creates new ones (Piaget, 1951). Papert’s constructionism is based on this theory, but he takes it a step further by stating that students will be more engaged in their learning if they are constructing something that is public that others can experience in some way. The sharing of their construction motivates the student to deeply think about what they’ve built and engage in complex critical thinking (Harel and Papert, 1991). This concept of constructing knowledge by making something public is strongly aligned with the maker movement’s focus on problem-solving using physical and digital fabrication in collaborative forums (Halverson and Sheridan, 2014).

Lave and Wenger’s ideas of situated learning are also visible in a makerspace. They argued that learning is situated; that is, it takes place within the activity, context, and culture of authentic experiences (Lave, 1988; Lave and Wenger, 1991). Situated learning is related to Vygotsky’s social constructivism, where sense-making is not distinct from the social context where it happens. Vygotsky believed that learning happens within the Zone of Proximal Development (ZPD), where appropriate facilitation allows students to master concepts that would be too difficult if left to fend for themselves (Vygotsky and Cole, 1978). This takes place through scaffolding, guided participation, and intersubjectivity, a process whereby two students arrive at a shared understanding through collaboration and negotiation (Newson and Newson, 1975). This scaffolding is important in makerspaces as participants begin to “level up” their skills, because if the situation is far too complex, they will struggle to transfer their knowledge and apply it to other situations or teach it to others (Anderson, Reder, Simon, 2000).

Although the concept of making to learn and the theories it is based on are not new, this type of learning is evolving into a movement and able to be more democratized due to digital desktop tools, a collaborative online culture (that extends offline), and the use of common design standards (Anderson, 2012). Due to variations in implementation and nomenclature, the number of schools employing in making is difficult to estimate, however, their numbers are certainly growing (Bell, 2015). Making's growing popularity is seen as a response to a need to redesign STEM education in the United States (Hertz, 2012). Lackluster results in the 2012 Program for International Student Assessment (PISA), as well as a well-documented unequal lack of access to STEM education for females and students of color generated a great deal of support for reform (Organisation for Economic Co-operation and Development, 2000, 2006, 2012; Margolis, Holme, Estrella, Goode, & Nao, 2017; Museus, Palmer, Davis, Maramba, 2011; Drew, 2011; College Board, 2014; Ericson, 2014). Additionally, with the focus of the CCSS and the NGSS being more on process and deep understanding than on results and surface understanding on various topics, having time and space set aside for students to create and innovate is seen as a path to better outcomes (Bell, 2015).

Why is making good for developing CT skills?

The maker movement was grown out of the computer hacker movement of 1990s (Litts, 2015). The concept of a hackerspace began in Germany with a collection of programmers that began sharing a physical space. Soon, members of these organizations began communing over electronic circuit design, manufacturing, and physical prototyping. "Making" developed as an answer to the exclusionary and somewhat illicit connotations of "hacking", being sure to include arts and crafts in its culture, but still retaining a kind of counter-culture environment (Cavalcanti, 2013). Nevertheless, the computing origins of making are usually strongly present, with digital resources referenced in combination with hands-on materials, providing ample opportunity to develop CT.

In addition to their computing history, makerspaces' strong alliance with design skills make it an ideal environment for developing CT skills, in the way that Denning advocated. The design process assists with metacognition, guiding students to think clearly about how they think through a problem. The design process makes thinking visible by encouraging the identification of a problem and its constraints, as well as how to generate and iterate solutions (Honey and Kanter, 2013). This explicit method of addressing a problem aligns with the development of CT skills. A student can use CT to move through design's iterative sequence, *questioning* to identify a problem, *decomposing* to consider options and constraints, *evaluating* to test and iterate solutions, etc.

The iterative nature of making mirrors the "reusing and remixing" CT practice (Brennan and Resnick, 2012) and the Use-Modify-Create progression for engaging youth in CT, as proposed by Lee, et al. (2011) (see Figure 1). They developed the model of this progression to illustrate how engagement was supporting and deepening CT acquisition.

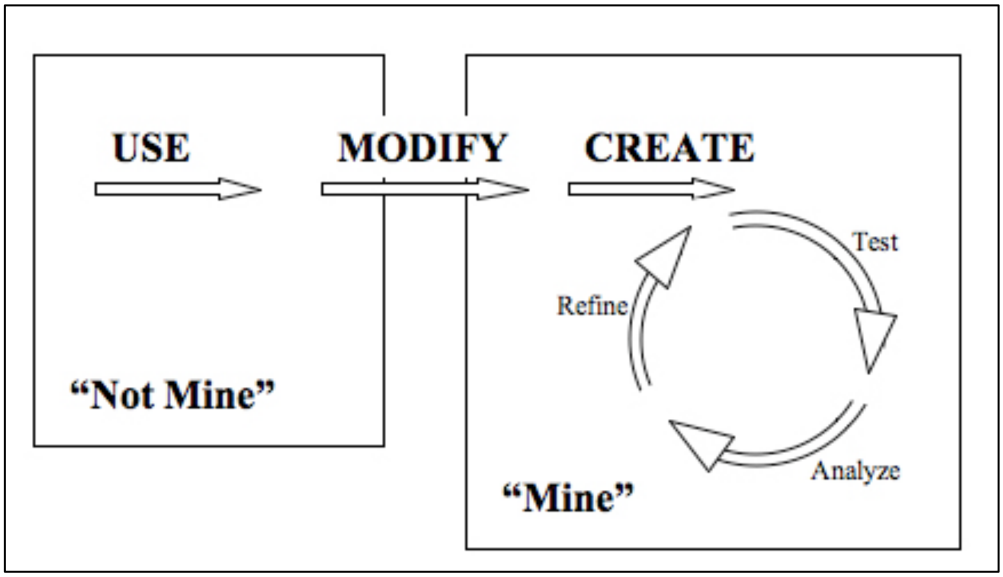


Figure 1. Use-Modify-Create Trajectory. Copyright 2016, Irene Lee. Adapted from "Computational Thinking for Youth in Practice" by I. Lee, F. Martin, J. Denner, B. Coulter, W. Allan, J. Erickson, J. Malyn-Smith, & L. Werner, 2011, ACM Inroads 2 (1): p. 35.

First, a student *uses* the work of someone else, like an existing computer model. Then, they begin to *modify* the code of the model to meet their needs and interests. Eventually, they will understand how to *create* code on their own, without using the crutch of another programmer's work. In making communities, the Use-Modify-Create progression is part of the community culture. Brahms and Wardrip referred to it as "Hack and Repurpose", as the "borrowing" culture in engineering design process' roots provides a strong backbone for how work is done (2014). More experienced makers provide novices with worked examples (Clark, Nguyen, and Swell, 2006), from which they can begin to modify and then innovate on their own.

Use-Modify-Create in making

Chu, Quek, Bhangaonkar, Ging, and Sridharamurthy (2015) used a "hack and repurpose" model for their study on encouraging the maker mindset in elementary-school students. The workshop they did with children involved a "Maker Theater kit" with which students had a template for which to build a miniature theater that would light up the characters they drew on individual cards. The theater kit was essentially the same for every student, but they could modify what characters would be featured in their story and where they would place them. The kit structure reduced the cognitive load on the students, so that they could focus on the tasks of being creative with the materials provided and getting the LED lights to work properly, and allowed them to create something much more complex than they could have created on their own. The content was scaffolded so that they were clear on what needed to get done. By the end of the Maker Theater workshops, students demonstrated signs of what the authors termed as a "maker mindset", which requires self-efficacy, motivation, and interest or "I can make," "I want to make," and "I like to make". However, the workshop was only a day long with a follow-up to the parents a week later, so it is difficult to determine whether these self-concepts developed more deeply or if the work and learning on the theaters was transferred to different contexts.

Debugging in making

Beyond just repurposing what others have done, the work of testing, analyzing and refining one's own work is a large part of making; no product is expected to be perfect right away. Makers use the CT skill of "debugging" on the fly, taking things apart in order to understand why they do or do not work, and building upon what they find (Litts, 2015). Obstacles and errors can and should be contextualized as an important part of the iteration valued in the design and problem-solving processes (Petrich, Wilkinson, and Bevan, 2013; Vossoughi, Escudé, Kong, and Hooper, 2013).

Fields, Searle, and Kafai (2016) built upon this idea by making debugging an entire learning activity. They created a series of e-textile artifacts (or "deconstruction kits") that intentionally had a fault in the coding, circuitry, or crafting, and had student pairs try to see what was wrong with them. As there are several reasons an LED might not be lighting, they found that the debugging exercise encouraged students to strategically isolate the various culprits and prioritize the order of their problem solving, as well as determine the right tools to help them solve the problem. Fields, et al, found that through debugging, students were interested in moving through the iterations of observation, hypothesis generation, and evaluation, an observation also made by Sullivan (2008) of students debugging robotics projects. The study also found that by being paired up, the students had to justify to each other what they thought the problem-solving process for each activity should be.

While this articulation between students was seen as helpful, there is no indication that there was an intentional reflection on what worked, what didn't and why, which could have helped the students develop the CT skills to understand what strategies to employ and when. If teachers want to ensure their students are building specific skills that can be transferred like those of CT, they need to discuss the skills explicitly and give students an opportunity to reflect on using those skills (Marin and Halpern, 2012), otherwise you are leaving the learning up to chance. Making's focus on "hands-on" learning as a demonstration of constructionism has resulted in research and implementation with a tendency to ignore "minds-on" learning (Perner-Wilson, Buechley, and Satomi, 2011; Moriwaki, Brucker-Cohen, Campbell, Saavedra, Stark, Taylor, 2012; Wagner, Gray, Corley, and Wolber, 2013; Kafai, Searle, Lui, Lee, Fields, and Kaplan, 2014; Qui, Buechley, Baafi, and Dubow, 2013; Chu, et al., 2015; Giannakos, Jaccheri, and Leftheriotis, 2014; Kafai, Lee, Searle, Fields, Kaplan, and Lui, 2014; Fields, Vasudevan, and Kafai, 2015; Katterfeldt, Dittert, and Schelhowe, 2015; Rode, Weibert, Marshall, Aal, von Rekowski, El Mimouni, and Booker, 2015; Schneider, Bumbacher, and Blikstein, 2015).

Wide walls in making

Because makerspaces can be open-ended and supportive of divergent ways and levels of making and knowing (Peppler, Halverson, and Kafai, 2016), they have the potential to exemplify "epistemological pluralism", or multiple ways of thinking and knowing that Turkle and Papert maintain is necessary in order to treat the computer as an expressive medium that, like paint and canvas or pen and paper, encourages differentiation in how it can be used (Turkle and Papert, 1992). Brennan and Resnick maintain that an important characteristic of a computational thinker is the ability to use the computer as not just a tool for the consumption of technology, but as a medium for design and self-expression (Brennan and Resnick, 2012, p.10). Incorporating making with CT expands computing, freeing it from being tied to the formalism and logic, to which scientific disciplines are more historically bound and have alienated those who do not approach problem solving in this way (Turkle and Papert, 1992). Both making and computing concretize the abstract (Wilensky, 1991; Turkle and Papert, 1992). Along these lines, while many in CS education support "low floor and high ceiling" entries into computing (easy for novices to become acclimated but with the flexibility to create complex artifacts), Resnick (2011) argues for the "wide walls" that making offers, encouraging the variety of projects that lead to learning.

Katterfeldt, Dittert, and Schelhowe (2015) described the wide walls they have implemented in the "constructionist learning environments" they have designed and evaluated in Germany, named *TechKreativ*. These makerspaces, which focus on digital fabrication with physical computing material for children, are built around skill development, as well as the notion of "Bildung". Bildung is a deep, sustainable learning that inculcates the learner in the practice of the maker. The wide walls allow for students to develop such diverse artifacts as a soccer shoe that measures the strength of a kick, a thief-proof handbag, and a moving robotic creature. They attempt to achieve epistemological pluralism through what they term as *begreifbarkeit* (similar to the English double meaning of 'graspable'), "imagineering", and self-efficacy. *Begreifbarkeit* is in reference to the interactions between the virtual and abstract with the physical and concrete, taking the computer out of its box and into the "real world" in the ways that Turkle and Papert described (1992). By "imagineering", the authors are referring to the ability to develop artifacts that relate to the creator's own personal life, the questioning and self-expression in CT for which Brennan and Resnick (2012) advocate. There is also an emphasis in their research on self-efficacy (the Power Principle), as they witness students feel empowered by the ability to contribute to the digital landscape, instead of merely consuming it.

Participants reported that they felt more capable in using technology and programming than after they participated in the workshop. Self-efficacy is important for epistemological pluralism to occur, for students to feel confident in transferring their knowledge and abilities to projects that are relevant to them. However, like other studies on making environments that measured self-efficacy, there seems to be no discussion as to what practices the instructors used to scaffold the learning or to engender dispositions like self-efficacy (Qui, Buechley, Baafi, and Dubow, 2013; Ornelas, Calderon, and Blikstein, 2014).

Making to learn content

Making also expands content knowledge by connecting it to other subject matter in a real way. By breaking down disciplinary boundaries, process- and product-oriented practices are highlighted, rather than a narrow focus on content, developing both engagement and skills (Sheridan, et al., 2014). For instance, Peppler notes that with e-textiles, students need to understand how both Ohm's law and stitching techniques need to work together to meet their goals (2013). Combining materials, methods, and content from various fields can encourage students to think about their knowledge in a way that can be transferred, motivating them to rethink what they use in one context and consider how it may work in other domains, whether they be digital or physical (Peppler, 2013). In other words, with its ability to be cross-curricular, making can be a demonstration for how CT can expand and connect various fields, especially (but not limited to) that of STEM.

Research demonstrates a positive effect of inquiry-based teaching on student learning of STEM; i.e. STEM is best learned by doing STEM, not just learning about STEM (Furtak, Seidel, Iverson, and Briggs, 2012; NGSS Lead States, 2013). The "doing of STEM" can be described as inquiry-based learning which engages students in "asking questions, defining problems; developing and using models; planning and carrying out investigations; analyzing and interpreting data; using mathematics and computational thinking; constructing explanations and designing solutions; engaging in argument from evidence; and obtaining, evaluating, and communicating information" (NGSS Lead States, 2013, p. 42). Through these inquiry-based STEM practices, students develop the necessary tools to think critically about and engage fully with the world around them. The key conceptual, epistemic, procedural, and social domains of inquiry-based education (Duschl, 2003; Furtak, et al., 2012) can be generated through making in makerspace environments. Makerspaces serve as a physical laboratory for inquiry-based learning, allowing learning to become real and relevant, while providing students with agency over their learning (Educause, 2013).

Khalili, Sheridan, Williams, Clark, and Stegman (2011) focused on game design as a making vehicle to learn STEM concepts, in this case immunology. Makerspace facilitators are fond of providing game design opportunities for students because they demonstrate the spirit of making in that they demonstrate understanding not just in the finished product, but throughout the design process (Mayer, 2013). Game design is relatively inexpensive compared to other making activities, as they involve few to no consumables (Pittser, 2016). A focus on games provides built-in scaffolding, with students moving up from playing and discussing games they like to making their own games based on their experience (Mayer, 2013; Pittser, 2016). Games also, as mentioned previously, can be used as conduits to developing CT skills (Denner, et al., 2012; Basawapatna, et al., 2010, Kazimoglu, et al., 2012). In Khalili, et al., the students worked on games in groups of four, each group selecting a different area of immunology to develop a game around using Game Maker. The researchers found that making a game led the students to (1) be made aware of where they did not understand some of the science concepts and investigate those gaps; (2) develop a sense of ownership that motivated them to make their game aesthetically appealing, entertaining, and accurate; and (3) be able to articulate the science concepts embedded within their games by the end of the program.

However, what was not included in this study was the consideration for student interest or identity. These elements were not included the program design, nor were they measured. The students (all African-Americans from underserved communities, 6 females and 10 males) were part of the program because they were interested in playing video games, but there was no discussion for how video games or immunology were a part of their lives, or how they intersected or were at odds with other interests or parts of their identity. The goal of the study was to "increase motivation, achievement, and exposure to STEM content for traditionally underserved students" (p. 229). While they did expose them to new concepts and programming environments, they did not include content or develop a culture that would then encourage students to develop this exposure into an invested interest, increasing motivation and achievement to bolster them through the many challenges of being a minority in the STEM fields (Eglash, Gilbert, Taylor, and Geier, 2013).

The NGSS suggests effective classroom strategies for non-dominant groups should include "connecting science education to students' sense of 'place' as physical, historical, and sociocultural dimensions", "applying students' funds of knowledge and cultural practices," and "culturally relevant pedagogy" (NGSS, Appendix D, p.7). In order to include students from non-dominant groups in STEM fields, it is vital to value their perspectives and funds of knowledge in the learning environment (Aikenhead and Jegede, 1999; Howard and Terry, 2011; Scott, Aist, and Zhang, 2014; Malaluan and Masangcay, 2015). Like Khalili, et al. (2011), there are various studies in making and makerspaces that, include the NGSS-recommended practices of "using project-based science learning as a form of connected science", as well as "multiple representation and multimodal experiences" (NGSS, Appendix D, p.7) to engage minority students, but do not engage the students in their own cultural understandings and representations of the world (Moriwaki, et al., 2012; Burge, Gannod, Doyle, and Davis, 2013; Franklin, Conrad, Boe, Nilsen, Hill, Len, Dreschler, Aldana, Almeida-Tanaka, Kiefer, Laird, Lopez, Pham, Suarez, and Waite, 2013; Giannakos, and Jaccheri, 2013; Qui, Buechley, Baafi, and Dubow, 2013; Kafai, Searle, Lui, Lee, Fields, and Kaplan, 2014; Rode, Weibert, Marshall, Aal, von Rekowski, El Mimouni, and Booker, 2015).

Makerspaces for cultural responsiveness

Makerspaces are well-positioned as environments for culturally relevant pedagogy in that they have the potential to fulfill the three dimensions of culturally relevant pedagogy, as proposed by Ladson-Billings (1995): (1) hold high expectations for what students can achieve, (2) give value to the funds of knowledge students bring (Moll, Amanti, Neff, and Gonzalez, 1992), and (3) develop students' awareness of sociopolitical issues that concern them. Calabrese Barton, Tan, and Greenberg (2016) explored these ideas as they described how and why youth engaged in making in an after-school makerspace program in Michigan and North Carolina called "Making 4 Change." The researchers frame their work from an equity standpoint for students for felt marginalized at school: they see makerspaces as places that lend legitimacy to non-school based problem-solving (Buchholz, Shively, Peppler and Wohlwend, 2014).

In "M4C", middle school students worked in groups to generate ideas around problems and associated questions. Using surveys they designed, the students explored concerns in the community. The teams used this data towards defining their problem spaces and exploring possible solutions, inviting community members to provide feedback after creating the first version of their artifact. Students were expected to design and construct helpful products that used renewable energy and to create short videos to educate others about their prototypes. Example artifacts the authors featured included an anti-rape jacket and a light-up greeting card.

By using a “mobilities of learning framework”, the authors demonstrated how makerspaces like M4C target the dimensions proposed by Ladson-Billings. Not only were the students’ interests valued, but the communities in which they lived, and the expertise of the people who lived in those communities, were valued as well. The expectations for the students were high for the gathering and analysis of the data, the development of the product, and the production of the video, and the students rose to the challenge. Finally, the researchers found that work in the makerspace engaged students around issues that were important to them and/or their communities, whether it was concern about sexual violence and bullies or economic struggles.

Schwartz, DiGiacomo, and Gutierrez (2013) were also concerned with designing environments that valued the knowledge, culture, and social connections students brought with them to a STEM learning environment. Their goal was to address issues of equity by encouraging the diverse group of participants to work together to develop the projects, regardless of age, ethnicity, or status. The authors examined El Pueblo Mágico, an after-school program that involved university students enrolled in university courses on child and adolescent development (called *amigos*, or friends), researchers, and K-8 youth from predominantly non-dominant communities working on STEM activities together. The program emphasized tinkering (instead of the usual excessive planning and frontloading that happens in classrooms), as well as the lack of hierarchy and defining problems together. Activities included building solar cars, squishy circuits (circuits where Play-Doh is used to conduct electricity), and sewn circuits.

While the blurring of lines between expert and novice and joint problem definition demonstrated a sensitivity to cultural responsiveness, the outcomes are unclear. Their methodology used cognitive ethnographies, which are concerned with how knowledge is constructed and used, not the knowledge itself (Williams, 2006). They also analyzed instances of questioning, direct assistance, modeling, side-by-side, and sharing ideas. This design can offer insight into process, which is integral to how making environments can make learning happen.

However, similar to the Calabrese Barton, et al. (2017), there is no measurement or analysis of what the students actually learned, i.e. where the process led. The learning environments established may enable equity in that they offer opportunities to learn STEM in a way that honors students from non-dominant backgrounds, but it should also be determined whether the outcomes of learning are equitable, as well. While many studies on making focus on measuring engagement or attitudes toward STEM, few analyze whether their making process resulted in learning STEM content or processes (Khalili, et al., 2011; Kolko, Hope, Sattler, MacCorkle, and Sirjani, 2012; Moriwaki, et al., 2012; Burge, et al., 2013; Giannakos and Jaccheri, 2013; Mellis, Jacoby, Buechley, Perner-Wilson, and Qi, 2013; Lee, Kafai, Vasudevan, and Davis, 2014; Katterfeldt, Dittert, and Schelhowe, 2015; Rode, Weibert, Marshall, Aal, von Rekowski, El Mimouni, and Booker, 2015).

Measuring outcomes and using making in the classroom

Because their research existed within an advanced placement computer science class, Kafai, Searle, Kaplan, Fields, Lee, and Lui (2013) did pay attention to the learning outcomes in using e-textiles to introduce computational concepts, practices, and perspectives (Brennan and Resnick, 2012). The study was on fifteen high school students who took part in a 10-week e-textiles module as part of an AP class. For computational concepts, they evaluated how students used input/output, digital and analog connections, control flow, and computing structures like sequences and conditionals by examining their circuit designs, code in both their initial designs and their final artifacts. To measure computational practices, they observed the approaches to computing students took in their designs and processes. And for computational perspectives, they analyzed pre/post interviews in which students discussed how they expressed themselves in their e-textile designs. In addition, the students were given debugging projects to test their knowledge of short circuits, polarity, and circuit closures. Students were asked a set of questions about the individual circuitry problems.

Kafai, et al., found a lot of variance in the complexity of the student designs in terms of the circuit type and how they coordinated functionality with their aesthetics. They saw students develop more efficient and sophisticated designs throughout the 10 weeks and all of the final projects included sequences, loops, conditionals, operators, and variables. The pre/post interviews revealed that students changed their perceptions on computing throughout the project, with students eventually seeing computing as relevant and a dynamic field, and themselves as computer scientists.

Though Kafai, et al., were not overt about constructionism being the guiding principle behind their work, one could assume this is the case, as Kafai has written extensively about constructionism (Kafai and Resnick, 1996; Kafai, 2006; Kafai, Desai, Peppler, Chiu, and Moya, 2008; Kafai, Peppler, and Chapman, 2009; Kafai and Burke, 2015). Useful in theorizing how learning happens, it is not clear how constructionism can be operationalized in the classroom. Similarly, the Brennan and Resnick (2012) CT definition provides an outline, but there is still no clarity for how a teacher can teach using these definitions. As Vanderlinde and van Braak (2010) point out, much education research does not get used in the classroom because it provides little practical use for educators. The researcher is looking for new knowledge, while the teacher is looking for new solutions to operational problems (Bates, 2002). Kafai, et al., do not provide the reader with actionable steps that allow the instructor to duplicate the experience they designed, but they are not alone. Much of making research is centered around one-time experiences, sometimes in a classroom, but often not, in which special materials and/or instructors/facilitators are brought into the learning space and researched, leaving the average teacher without much understanding of how this kind of learning could and should happen in her classroom (Khalili, et al., 2011; Kolko, et al., 2011; Moriwaki, et al., 2012; Giannakos and Jaccheri, 2013; Schwartz, et al., 2013; Qui, et al., 2013; Wagner, et al., 2013; Fields, Vasudevan, and Kafai, 2015; Katterfeldt, et al, 2015; Calabrese Barton, Tan, and Greenberg, 2016).

How can we use a conceptual framework of learning to better develop making environments to teach CT?

For making and CT to happen cohesively to provide opportunities for learning in a classroom or makerspace, an appropriate structure must be put in place. While we have reason to believe that making has the potential to teach CT to children, we don't know how this might actually be accomplished. The available studies involving making in order to develop CT are promising (Bers, Flannery, Kazakoff, Sullivan, 2014; Denner, et al., 2011; Basawapatna, et al., 2011; Lee, 2011; Wilkerson-Jerde, 2013; Brennan and Resnick, 2012; Wolz, et al., 2011; Grover, Pea, and Cooper, 2015; Alpert, 2015; and Farris and Sengupta, 2014). But they only go so far. A lot depends on how the content and strategic processing concepts are introduced and how students are taught to learn with them (Yadav & Cooper, 2017). Making for learning cannot be limited to just providing students with materials and a room, and hoping for the best. The learning must be supported and scaffolded in a way that develops expertise.

Collins, Brown, and Newman's (1989) cognitive apprenticeship model is similar to Lave and Wenger's examination of traditional apprenticeship practices. However, unlike traditional apprenticeship, the cognitive apprenticeship framework focuses on the higher-order metacognitive and strategic processing skills employed by experts in a domain. Facilitators (or in school settings, teachers), make the implicit processes experts utilize explicit, introducing the domain to their students through appropriate strategic processing methods. These methods are often not learned didactically, but informally through apprenticeship methods like observation and coaching, with the learning of skills and knowledge in their specific context.

Cognitive apprenticeship places an emphasis on the decontextualization of knowledge, situating learning in a variety of settings so that students can practice applying the skills they have learned in diverse contexts. Collins, Brown, and Newman argue that schools rarely provide opportunities for students to apply this kind of conceptual knowledge, thereby disregarding the development of higher order thinking skills as the organizing principles of expertise.

Cognitive apprenticeship was proposed as a way to integrate the effective scaffolded and domain-specific practices of traditional apprenticeships with more traditional learning settings that do not have the luxury of small student-teacher ratios of apprenticeship models and have a more generalized focus on knowledge and cognitive skills (Collins, Brown, & Newman, 1989). In order to make this integration possible, cognitive apprenticeship approaches learning environments holistically, composed of four dimensions: content, method, sequence and sociology (Collins & Kapur, 2014). These dimensions are divided into a set of characteristics to be examined when developing and analyzing learning environments.

Content refers to *domain knowledge*, but also *heuristic strategies* used by experts, metacognitive *control strategies* for monitoring progress, and *learning strategies* developed for other knowledge. These strategic components of content are often left as being implicit in learning environments (Margulieux, Dorn, & Searle, 2019).

Collins, Brown, and Newman state that, in regards to sequencing, learning design must keep in mind the need to understand how to support the integration and generalization of knowledge and skills. By systematically *increasing complexity* and *diversity*, while focusing on *global before local skills*, tasks are sequenced appropriately and students are encouraged to think conceptually before attending to details.

The sociology strand focuses on how structuring the social context encourages the development of understanding the importance of multiple ways of teaching and learning, and therefore, collaborative skills. With *situated learning*, students solve problems in an environment or situation that is a reflection for how their knowledge will be used in the future, and teaching them how to think and act like experts in that domain develops a *culture of expert practice. Intrinsic motivation* is generated in a learning environment in which students take part in an activity because they are interested in the goal and/or the process, not because they are interested in receiving an external reward. *Exploiting cooperation* fosters cooperative problem solving while *exploiting competition* uses the strategy of comparing what each student produces.

Cognitive apprenticeship makes novice and expert strategies explicit by employing a variety of pedagogical methods. These methods, namely *modeling*, *coaching*, *scaffolding*, *articulation*, *reflection*, and *exploration*, are intended to "help students acquire and integrate cognitive and metacognitive strategies for using, managing, and discovering knowledge" (Collins, Brown, & Newman, 1989, p. 480).

- *Modeling* demonstrates a particular task while voicing one's inner thought process for direct observation by the learners, so that they can better understand the processes necessary to accomplish the task.
- *Coaching* is when an instructor offers feedback, and reminders.
- *Scaffolding* (Wood, Bruner, & Ross, 1976) is the support or suggestions the teacher gives the student in order to assist in the completion of a task, and fading is the gradual removal of these supports. Scaffolding is employed when students have run out of reasonable ideas for how to proceed (Brown & VanLehn, 1980).
- *Articulation* is any way of encouraging students to clearly express their understanding or their problem-solving process.
- *Reflection* encourages students to consider how their approaches compare to those of the experts and other learners.

- *Exploration* pushes students to independently solve problems. Unlike the completely unstructured inquiry criticized by Kirschner, Sweller, and Clark (2006), exploration in cognitive apprenticeship utilizes the other methods to help guide the learning and be less taxing on working memory (Hmelo-Silver, Duncan, & Chinn, 2007; Margulieux, Dorn, & Searle, 2019).

These methods, done in conjunction with the other dimensions of content, sequencing, sociology, create a learning environment where metacognition is highlighted and provide a path to future learning. By integrating these dimensions in a classroom or makerspace focusing on developing CT skills, students can better understand the normally tacit processes involved in computing-based problem solving.

Conclusion

In this review, I examined the reasons for the present focus on CT. I then elaborated on the prevailing definitions of CT, and described why I use the process-based definition of Selby and Woollard (2014) in combination with the ISTE and CSTA (2011) description of dispositions and the strengths and weaknesses of current research of CT. I then looked at the maker movement, what it is, why it is generating excitement in the educational community, and the promise it shows in developing CT. Finally, I explored cognitive apprenticeship as a conceptual framework for structuring research on how CT can best be developed in a making environment.

This review provided a foundation for my study, in which I see the need to investigate the practices instructors could use to help students develop CT skills in a makerspace, structured by a conceptual framework. I clearly defined what I mean by

CT, and am intentional in examining CT with maker activities because making encourages agency and a use-modify-create ethos, integral aspects of CT, while expanding the reach of CT beyond the computer science classroom. Examining instructional practices through the lens of a cognitive apprenticeship framework allows for a clearer understanding of what best practices are for eliciting CT in a makerspace environment.

RESEARCH QUESTION

This study will explore the ways in which making environments for youth can be structured to develop CT skills and dispositions. Specifically, the research question being investigated is: **Within makerspace activities what evidence exists regarding promising practices that support youth development of CT skills and dispositions?**

III. METHODS

Introduction

The intent of this exploratory study is to learn about structuring making-based approaches for youth to develop CT skills. The results of this study of an informal environment are intended to develop a theory of change for various types of educational settings for high school-aged students. In this study, qualitative data such as field notes, transcripts from video/audio, screen recordings, and participant notebooks served to assess how CT develops and what types of activities facilitate its development.

Case Study

Yin (2003) extols the case study method for, unlike other research methods, allowing investigators to maintain the comprehensiveness and authenticity of real-life events. He states that the five important components of a research design for case studies are (1) a study's questions, (2) its propositions, (3) its units of analysis, (4) the logic linking the data to the propositions, and (5) the criteria for interpreting the findings.

The question I am examining is "Within makerspace activities what evidence exists regarding promising practices that support youth development of CT skills and dispositions?" My proposition revolves around the cognitive apprenticeship framework (Collins, Brown, & Newman, 1989): putting into place instructional structures that call attention to the strategic processing tools of CT increases student understanding of CT and how to employ it to better understand the problem or subject matter. The "bounded system" (Smith, 1978) by which I analyzed the data is constraining my analyses to four activities being conducted in one four-hour day in a summer program. In order to link the data to my proposition, I focused on the resources and scaffolds instructors employ CT using the cognitive apprenticeship framework. To interpret the data, I examined interactions between facilitators and participants in regards to the development of CT over the span of the four activities.

The summer program that was analyzed has many of the characteristics of informal and formal learning spaces like makerspaces and classrooms, providing insight into how these may operate more generally. In other words, this is an instrumental case study, or the study of a case to give insight into an issue or build theory (Stake, 1995). In this instance, the issue instructional practices that elicit the development of CT skills in a makerspace environment.

This study is an analysis of data from a larger National Science Foundation funded project titled Assessing Computational Thinking for Maker Activities, or ACTMA. The goal of ACTMA was to create an embedded, adaptive, and culturally responsive formative assessment of CT in STEM that can be used in informal learning spaces such as makerspaces, but can also be brought into more formal physics classroom experiences.

This research examined data collected for ACTMA, but used different forms of analysis than what was used in the funded project. All of the data were collected under an IRB that encompasses the use of the data for the purposes of this study. As such, all appropriate consent, permission, and assent were obtained. The research team consisted of six researchers responsible for taking field notes at different points of the academy, two researchers responsible for the video/audio/screen recordings, and one researcher leading the quantitative data collection.

Sample Selection

Participants in the summer program that were observed were high school students (9th to 12th grade) from seven Chicago Public Schools. The full sample consisted of 19 participants. Sixteen of the participants identified as female and three as males. The participants ranged in age from 13 to 18 years old ($M = 15.84$, $SD = 1.12$). Ninety-five percent (18) of the students qualified for free or reduced-price lunch. Sixteen of the participants (84%) indicated that at least one of their favorite subjects in school was STEM-related. One participant dropped out of the program because of other commitments. Additional demographic information of the student population is not presented in order to protect the identity of participants. Targeted schools were a part of a Chicago GEAR UP (Gaining Early Awareness for Undergraduate Programs) partnership. GEAR UP schools are persistently low-achieving schools in Chicago Public School District 299 that have agreed to a partnership with the Chicago GEAR UP Alliance, which is a consortium of universities across Chicago that are committed to working with communities to encourage and prepare more underrepresented students to get a postsecondary education.

The facilitators that were observed included: three of the out-of-school-time educators from the development team, two of which served as leading facilitators, and three library staff members. Additional demographic information of the facilitator population is not presented in order to protect their identities. These library staff facilitators usually work in YOUMedia. YOUmedia is a learning space devoted to teens at 12 Chicago Public Library locations. With a focus on digital media and the maker movement, teens can engage in projects that include photography, video, music, and 2D/3D design. All facilitators had experience working with youth in informal settings, in after-school programs and public library settings. The lead facilitators were part of the development team for the program's activities and one of the facilitators led the writing of the curriculum.

Recruitment

An ACTMA staff member worked closely with the staff in each school to coordinate times for recruitment. Student subjects for observations were recruited in person during homeroom period using the approved recruitment script (Appendix 4). Participants assented via written assent. Parental permission was received via written consent. Participants were instructed to take home both the assent and parental permission form (in both English and Spanish) and return both to an unmarked envelope within a week, regardless of whether they marked "yes" or "no". ACTMA staff collected the envelope at the end of the week.

Facilitators were recruited through their work at YOUmedia at the Harold Washington Chicago Public Library or with GEAR UP. Both sets of instructors had the form read to them and it was made clear that their participation would have no impact on their employment.

Documents that were used:

- Video release form (Appendix 5)
- Participant assent (Appendix 6)
- English parental permission (Appendix 7)
- Spanish parental permission (Appendix 8)
- Facilitator/Mentor consent form (Appendix 9)

Retention

To enhance retention and to discourage attrition, we ensured that all involved participants understood the significance of the study and how their role will play a large part in it.

Study Setting

The ACTMA summer program took place in the multipurpose room at the Harold Washington Chicago Public Library, located at 400 S. State St. in Chicago. Washington Library was selected because of its accessibility by public transportation from all of the schools the participants attend, as well as its connection to the YOUmedia makerspace.

The Case

The summer 2016 ACTMA program was 9 days long, from June 27 through July 9, with a 3-day break for the 4th of July holiday from July 2 to 4. Each session was 4 hours long, from 10 am to 2 pm (lunch was served at noon). The progression of the program took students from learning about basic circuits, to series and parallel circuits, electromagnets and motors, to the Makey Makey and Arduino microcontrollers. Students were asked a couple of times to bring in some artifacts of the hobbies in which they engage. The final two days of the program were devoted to students developing and sharing their final projects, which were an incorporation of what they learned over the two-week program with their interests. The schedule is detailed in Appendix 1.

In the program, the lead facilitators would describe the activities of the day and provide directions for the entire program. Then they, along with the other facilitators, would roam around the room, assisting students when they requested assistance, or when they saw a student silently struggling, or not doing anything at all. Generally, each facilitator focused on a certain area of the room, assisting the same four to six participants over the course of the day, but would sometimes roam to other parts of the room when their participants were involved in their work, or when they needed to consult with another facilitator.

This study focused on one day in the second week of the summer program, July 6th, 2016. One of the lead facilitators was unable to be present most of the day, as well as one of the YOUmedia staff members, but all other facilitators were present. Two students were also absent. This day was selected because it encapsulated four of the types of activities participants were asked to do at different points in the summer program: (1) work in large groups to review the material, using their own bodies to recreate a circuit; (2) play with the provided materials to make an interesting working circuit; (3) use computing tools to build on what was learned about circuits; and (4) consider the ways they could use the material they learned to create a final project aligned with their own interests.

Tabletop Breadboard Activity

In the first week, participants created simple, series, and parallel circuits using alligator clips and battery holders (see Figure 2). They were then introduced to breadboards on the sixth day of the program, as their circuits were getting more complicated and the breadboard would assist in working with the Arduinos, but most participants were confused by them.

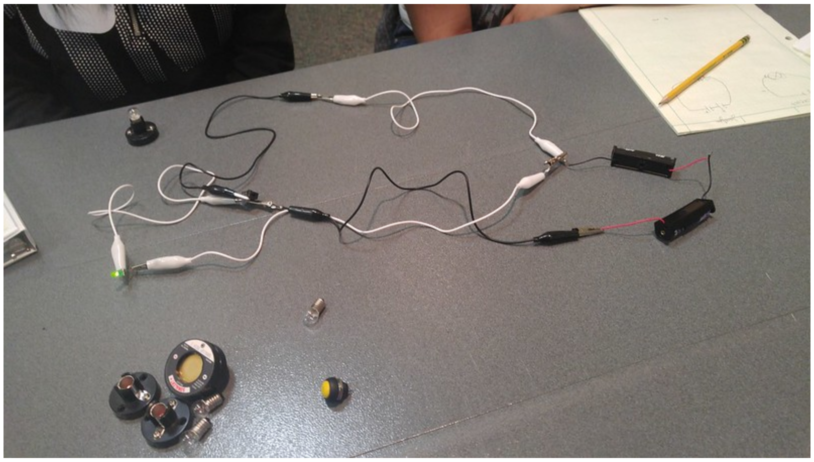


Figure 2. A parallel circuit created with alligator clips, battery packs, and LEDs

A breadboard is a plastic board, usually rectangular in shape, with many small holes in a grid pattern used for prototyping electronics (see Figures 3 and 4). The holes are intended to easily insert electronic components to prototype an electronic circuit. The connections are not permanent, facilitating the ability to change the placement of components. Metal terminal strips run underneath the holes to connect them to one another. A middle divider separates both sides of the breadboard, which requires a user to connect the two sides if they the same power source on both sides. Along both sides of the breadboard are columns of holes called buses. Buses supply power to the circuit by connecting them to an external power supply, like a battery. The buses typically run the entire length of the breadboard, while the main breadboard rows are connected in sets of five holes.

Figure 3. Breadboard (Image from sparkfun.com)

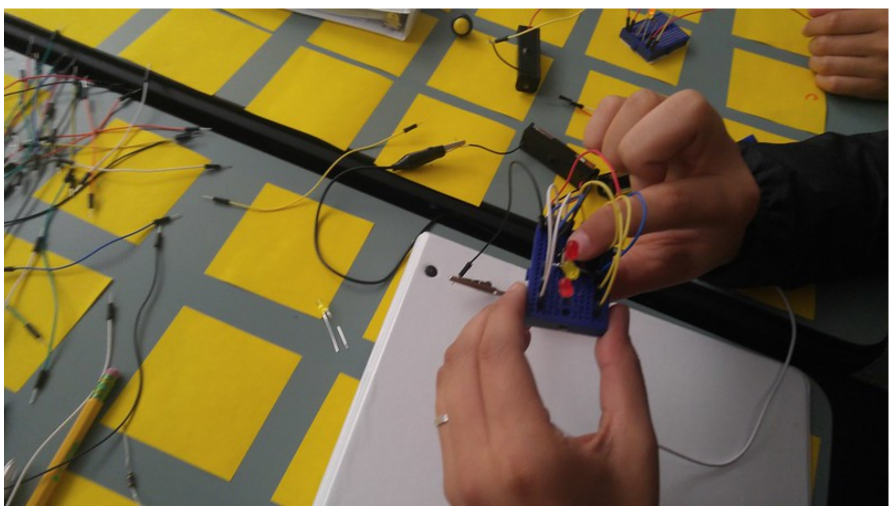


Figure 4. Parallel circuit using a breadboard

To clarify how breadboards work, at the end of the sixth day the lead facilitator suggested adapting the Bodies as Circuits activity used in the first week to review simple, series, and parallel circuits. In Bodies as Circuits, participants each took on a role as a component of a circuit (e.g. battery, LED, wire, etc.) and created circuits as groups using their entire bodies (see Figure 5). For breadboards, the teaching and research staff devised a way for participants to use their bodies as a team to create circuits on large tabletop breadboards using post-it notes, reminiscent of the game Twister (see Figure 6). Post-it notes took the place of the holes in a breadboard, and each column was given a letter, while each row was given a number in order to identify each hole. Because the breadboard was composed of two tables, the space between the two tables served as the middle divider. Different colored post-its were placed in two columns on both sides of the tabletop breadboard to serve as buses. Participants would then asked to create simple, series, and parallel circuits using the tabletop breadboard.

Figure 5. Participants recreating a parallel circuit as a group with their bodies

Figure 6. Participants recreating a series circuit on a breadboard with their bodies

Breadboard Exploration Activity

After working on the tabletop breadboard activity, the participants were given a breadboard and various types of LEDs (light emitting diode), switches, potentiometers, capacitors, batteries, and jumpers (wires used to interconnect the components of a breadboard) with which to play and experiment and create interesting circuits (see Figure 7).

Figure 7. Participants experimenting with breadboards and components

Arduino Activity

Next, participants were shown an Arduino microcontroller, how to connect it to a computer, and how to connect a breadboard to the Arduino. They were then shown some sample code to make an LED blink, which was referred to as the Blink Code. They were asked to play with the variables in the code and the arrangement of the breadboard to see how modifications would allow them to create their own artifacts.

The sample code is seen in Figure 8. As explained in the commented code, this code would control an LED placed in pin 13 on the breadboard. "HIGH" turns the LED on, and "LOW" turns it off. Entering "1000" for the delay means a delay of one second, as the program measures in milliseconds.

Blink

```
/*
  Blink
  Turns on an LED on for one second, then off for one second, repeatedly.

  This example code is in the public domain.
 */

// Pin 13 has an LED connected on most Arduino boards.
// give it a name:
int led = 13;

// the setup routine runs once when you press reset:
void setup() {
  // initialize the digital pin as an output.
  pinMode(led, OUTPUT);
}

// the loop routine runs over and over again forever:
void loop() {
  digitalWrite(led, HIGH);   // turn the LED on (HIGH is the voltage level)
  delay(1000);               // wait for a second
  digitalWrite(led, LOW);    // turn the LED off by making the voltage LOW
  delay(1000);               // wait for a second
}
```

Figure 8. The code participants were asked to modify

Final Project Planning Activity

After these three activities, participants were given time to think about and plan for their final projects, which they were to start working on the next day. For the final project, they were instructed to develop something that aligned to their interests, that utilized some of the skills and materials they learned about in the summer program, and would be somewhat challenging.

Data Collection

This research uses a case study approach, which requires multiple forms of data to draw conclusions (Yin, 2003). The collected data includes field notes, video and audio recordings, screen recordings, participant notebooks, and project artifacts. A Methods Overview in Appendix 2 provides a summary for how the research was planned.

Field Notes

Four researchers took field notes during the session. The field notes were collected during the activities, using a simple descriptive observation instrument (Appendix 3).

Video, Audio, and Screen Recordings

While participating in the program, participants were videotaped and audio recorded. Participants were sitting in groups of five to six at three large tables. A video camera was pointed at each table and at the front of the classroom, where the lead facilitator was sometimes positioned. The four hours from each of the four cameras resulted in 16 hours of footage in total. A digital audio recorder was placed in the center of each of the three tables and lavalier microphones attached to audio recorders were used to record what all of the four facilitators were saying, resulting in 26 hours of recording in total (2 hours from one of the facilitator's lavalier mics was lost). Transcripts of the audio and video recordings where either transcribed by myself, or by Rev.com.

In addition, when work was done on a laptop, the screen was recorded using Camtasia (barring technical difficulties, of which we had a few, as the file sizes of these recordings were sometimes too large for the older computers to handle), resulting in five hours of screen recordings.

Notebooks

Participants were all given "engineering notebooks" to record the work done in the program. Participants were told at the beginning of the program that the notebooks were not only for them to write notes about what the instructor says, but for them to document their own thinking. Participants were made aware that several professions (engineers, scientists, artists, etc.) use notebooks as part of their practice, and that adopting the practice not only prepared them professionally, but ensured all of their ideas and iterations are recorded. They were told they could include questions, ideas, diagrams, sketches, etc. They were also asked not to destroy anything in the notebook; if they decided against an idea or did not like something they had written down, they could cross it out, but the reader should still be able see what the initial idea was. They were made aware that the notebook was not a private diary, and that it will be looked at by their teacher, peers, and researchers to better understand their process. The notebook also featured as part of their final presentation. At the end of the program, the notebooks were kept and scanned. Seventeen notebooks were analyzed.

Artifacts

Participants produced artifacts as a result of the individual activities, as well as part of the final project. Photographic documentation and video were taken of artifacts.

Data Integration

After reviewing all of the data sources individually, they were woven together when possible into integrated documents. For example, when someone mentioned a drawing on audio, the drawing in their notebook they were referring to was inserted into the audio transcript; when someone described some sort of action, the video was matched to that moment in the audio and a screenshot was inserted into the transcript document; or when the facilitator was describing what was happening to code on the screen, a screenshot of that moment in the screen recording would be embedded with the transcript text.

Data Analysis

All of the observation field notes and transcripts were uploaded into Dedoose, a web-based application for qualitative and mixed methods research. From those notes and transcripts, I conducted open coding, noticing facilitator practices and participant actions. I also coded for instances of where CT skills and dispositions were noticed in participants, as well as what cognitive apprenticeship method was being used by facilitators, if any.

Finally, I coded for the impact of different instances, measuring them on a scale of 1-4, with 1 being not a very effective moment, and 4 being a very effective moment. Impact in this case refers to how clearly the CT skill or disposition was demonstrated by instructor, or how well it was adopted by the participant. These instances included the practice of the facilitator (if applicable) with the resulting action of the participant. A code of "1" indicated that either the student was without direction or the facilitator was pushing the project forward without understanding from the student. In instances coded with "2", facilitators did some scaffolding or coaching, but students were still without clear directions. When the code "3" was used when facilitators were clear in their use of methods and students were able to move beyond where they were unsure. Finally, "4" indicated that students were using CT in a sophisticated manner.

To code for CT skills, I used the definition of CT skills from Selby and Woollard (2014) and the ISTE and CSTA (2011) definition of CT dispositions, as detailed in Table 3 and Table 4. To code for the methods of cognitive apprenticeship, I used the definitions from Collins, Brown, & Newman (1989), as detailed in Table 5. I wrote memos when I noticed themes arising or wanted to make note of an individual instance.

Table 3

CT skill definitions used in coding

Term	Definition
Decomposition	Thinking of a problem in its component parts
Abstraction	Reducing unnecessary detail
Pattern recognition	Identifying similarities and connections
Algorithmic thinking	Using a clear set of steps to get to a solution
Evaluation	Ensuring the solution is fit for the purpose

Table 4

CT disposition definitions used in coding

Term	Definition
Confidence in dealing with complexity	Does not avoid difficult problems
Persistence in working with difficult problems	Persevering through challenges
Tolerance for ambiguity	Comfortable with uncertainty and ill-defined problems
The ability to deal with open-ended problems	Knowing there are multiple solutions
Communication	Able to express oneself
Collaboration	Able to work with others towards a solution

Table 5

Cognitive apprenticeship definitions used in coding

Term	Definition
Modeling	Demonstrating a particular task while voicing one's inner thought process
Coaching	Observing and offering hints, feedback, reminders, and new tasks
Scaffolding	Support or suggestions provided when at an impasse
Articulation	Encouraging the clear explanation of thinking
Reflection	Making connections to the work of experts and other learners
Exploration	Pushing students to solve problems on their own.

After I coded one integrated transcript, a colleague and I each coded a section of a transcript individually, we compared codes and discussed discrepancies between us. After the discussion, we coded another section of the transcript individually and then compared them again, again discussing the discrepancies. On the second comparison of codes, inter-rater reliability for CT skills and dispositions was 70%, and 75% for cognitive apprenticeship methods. I coded the rest of the data (the video, audio, screen recordings, and notebooks), creating memos of themes I noticed arising. After reviewing my analysis, I generated and organized themes. I then reviewed the sections where the themes were notable and included the other dimensions of cognitive apprenticeship (content, sequence, sociology) that were evident.

Audio that was duplicated in more than one transcript was coded only once, and field note data were coded only for actions and researcher observations, and not conversations.

Triangulation

With this in mind, in order to demonstrate the credibility of the study, I used Creswell and Miller's (2000) framework for determining validity procedures. Because I mostly operate with a postpositivist paradigm and am relying on my own lens for determining validity, I employed triangulation to maintain credibility (Creswell & Miller, 2000). Triangulation occurred both through involving another researcher in the interpretation of data (investigator triangulation) as well as using multiple sources of evidence (data triangulation) (Patton, 1987). To this point, Yin (2003) states that one of the strengths of case study collection is being able to use various sources of evidence, an element that limits other methods like surveys and experiments. The field notes and media recordings, in conjunction with the notebooks, builds a strong understanding of what occurred in the summer programs. Yin (2003) is clear that for data triangulation to really occur, the events of the case study have to be supported by more than a one source of evidence. In other words, the data has to converge on a conclusion. If we individually analyzed the field notes, video, audio, screen recordings, or notebooks, we would end up with separate conclusions and, therefore, separate sub-studies or a "non-convergence of evidence" (Yin, 2003). The methods I used to analyze the various sources of data simultaneously falls within Yin's definition of triangulation, and therefore meets the definition of validity through convergence of evidence.

Categorical Aggregation and Direct Interpretation

This research is an instrumental case study, or research on a case to gain understanding of a more general concept (Stake, 1995). Specifically, I examined a day in a summer program to reach conclusions about developing CT skills in a makerspace environment. While categorical aggregation would make the most sense with an instrumental case study, I used both categorical aggregation and direct interpretation of individual instances to generate meaning (Stake, 1995). Categorical aggregation was used in order to develop a collection of themes from the data and discover the themes that could generalized. Stake (1995) sees the search for meaning lying in finding consistent patterns in certain conditions. Direct interpretation is valuable because even if a feature of an activity only happens once, it may provide insight into other instances. It also allows for naturalistic generalization to take place (Stake, 1995), encouraging the reader to relate to the accounts, and think about how it may apply to their own experience.

IV. FINDINGS

This study set out to discover what evidence exists regarding promising practices that support youth development of CT skills and dispositions within makerspaces. These findings were based on coding of video and audio recordings, field notes, and participant notebooks. Triangulation was conducted for validity and accuracy.

This chapter will first discuss the most often coded and most effective practices through open coding, followed by how these practices coincided with CT skills and dispositions, and then how they were coupled with cognitive apprenticeship methods. After defining the practices, the chapter will then detail how these practices worked to develop CT skills and dispositions through a cognitive apprenticeship model, focusing on the cognitive apprenticeship methods, but also discussing the content, sequence, and sociology dimensions when appropriate. These descriptions will be supported through examples from transcripts, field notes, screen grabs of video and screen recordings, photographs, and notebooks in order to explain the example and identify the cognitive apprenticeship method, as well as the CT skills and dispositions on display. For reference, a more detailed set of the examples organized in tables can be found in Appendix 11. All of the names used in the examples have been changed to protect the subjects.

Open coding revealed a set of practices that the facilitators and participants engaged in that kept reappearing throughout the activities. The most common seen throughout the day were tinkering, embodying, walkthroughs, drawing, and debugging.

Defining the Terms

Tinkering was identified when participants were seen to be or encouraged to solve problems by "messing around" through play and discovery (Martinez & Stager, 2013). In those instances, they were not following a prescriptive series of steps to achieve a defined goal. Rather, participants were encouraged to creatively use the available materials to explore and determine their own goal, within certain boundaries. Tinkering was the most common practice coded. In the tabletop breadboard activity, participants were tinkering with variations on their human circuits. In the breadboard exploration activity, participants tinkered with different components in series and parallel circuits. In the Arduino activity, participants tinkered with the code and the breadboard it was controlling.

The *embodying* code was applied to instances where participants or facilitators were using their own body or the body of another participant or instructor, to solve a problem or clarify an idea. Despite the fact that an entire activity was devoted to using their bodies, the code was only applied when participants were using the body metaphors as a way to understand or communicate, not in field notes that described the actions they were taking.

The term *walkthrough* is in reference to a review process that programmers do, in which they walk their peers through the code. It is often associated with tracing, which is a process of trying to emulate the process of a computer executing code (Fitzgerald, Simon, & Thomas, 2005). This type of "touring" the listener through how their circuit or program worked (or how they wanted it to work) began with the creation of the tabletop circuit. Walking through their individual concept of their circuit with each other in order to ensure they were on the same page was vital to making their “circuits” viable, especially as each person in the group was playing a role in the circuit. So, they would repeatedly talk through the chain of events that composed their circuit with one another as they thought through how it should work. They were also preparing themselves for presenting their tabletop circuit to the entire class. Walkthroughs then extended to the breadboard exploration activity and to working with Arduinos.

The *drawing* code was applied to instances where participants used drawing to plan, document, or work through a problem with a circuit. Several examples came from the participant notebooks themselves, however, the code was mostly applied to when facilitators would ask participants to draw their circuit, when a participant asked for help with their drawing, or when the facilitator and the participant were using the drawing as a way to explain themselves.

Debugging refers to when participants performed the systematic application of analysis and evaluation to determine why something is not working (Csizmadia et al., 2015). While debugging is a term commonly used while coding, the participants employed a type of debugging in all three activities, including the two that did not use any programming at all. In the tabletop breadboard activity, participants would realize their circuit didn’t make sense and had to talk through which element needed to be reworked. While doing the breadboard exploration activity, participants would try different strategies to determine why their circuits were not working. For the Arduino activity, participants had to debug both the code and the breadboard connected to their computers.

Code Counts

In addition to being the most common practices noticed during the day, as seen Table 6, tinkering, embodiment, walkthroughs, drawing, and debugging were also associated with the most memos and were coded as being the most effective or having the most impact. The tally of the impact score attached to each instance and its associated practice is coded is listed in Table 7, with 1 having the least impact and 4 having the most. Examples of interactions that were coded with a 1, 2, 3, and 4 are listed and explained in Appendix 12.

Table 6

Totals of top 5 practices identified through open coding

Practice	Code counts
Tinkering	202
Embodying	79
Walkthroughs	72
Drawing	72
Debugging	66

Table 7

Total code counts and percentages of impact on CT effectiveness

	1	2	3	4
Tinkering	46 (22.7%)	80 (39.6%)	52 (25.7%)	24 (11.5%)
Embodying	8 (10.1%)	27 (34.2%)	29 (37.7%)	15 (19%)
Walkthroughs	2 (28%)	11 (15.3%)	28 (39%)	31 (43.1%)
Drawing	15 (20.1%)	13 (18.1%)	24 (33.3%)	20 (28%)
Debugging	3 (4.5%)	14 (21.2%)	26 (39.3%)	23 (35%)

The data were also coded for instances of CT concepts as defined by Selby and Woollard (2014) (Table 8), CT dispositions/attitudes as defined by ISTE and CSTA (2011) (Table 9), as well as methods of cognitive apprenticeship as defined by Collins, Brown, and Newman (1989) (Table10).

Table 8

Instances of practices that correspond with computational thinking concepts

	Decomposition	Abstraction	Pattern Recognition	Algorithmic thinking	Evaluation
Tinkering	52	2	53	25	17
Embodying	8	8	0	15	3
Walkthroughs	1	2	0	43	4
Drawing	0	19	0	16	4
Debugging	25	1	16	13	0

Table 9

Instances of practices that correspond with computational thinking dispositions/attitudes

	Confidence in dealing with complexity	Persistence in working with difficult problems	Tolerance for ambiguity	The ability to deal with open ended problems	Communi-cation	Collabo-ration
Tinkering	17	17	64	27	4	7
Embodying	6	1	7	1	34	24
Walkthroughs	6	0	10	8	42	14
Drawing	12	2	9	5	31	1
Debugging	6	46	14	2	10	1

Table 10

Instances of practices that were promoted by methods of cognitive apprenticeship

	Modeling	Coaching	Scaffolding	Articulation	Reflection	Exploration
Tinkering	19	42	25	14	3	48
Embodying	4	11	9	15	1	3
Walkthroughs	15	4	9	17	1	1
Drawing	9	6	3	9	17	2
Debugging	19	16	9	6	2	3

Tinkering

Tinkering was the most widely used of the practices throughout all three activities. Participants were encouraged by facilitators to “play around” or “explore”. All of the activities were developed to allow participants to have agency in how they wanted to design their final project, whether it be a circuit they designed with their peers on the tabletop breadboard, an actual circuit being plugged in on a breadboard, or an Arduino program. The facilitators embraced the open-ended nature of the activities and continually encouraged the participants to "be creative" in the development of their artifacts.

However, the tinkering mindset was not always a comfortable place for the participants. They sometimes seemed unsure how to move forward. The freedom could be overwhelming and the facilitators had to remind that "this is not school" and responding "you're making whatever you want to make" when asked what they were supposed to do. Open-ended exploration was sometimes unfamiliar to them.

Aside from the discomfort from not having a clear goal, at the start of each activity, the participants were a lot of times guessing, plugging themselves into the tabletop breadboard, plugging components into the actual breadboards, or entering code into the Arduino software, without understanding what they were doing. At times, they were able to use the guess as a starting point, understanding their mistake and either self-correcting or correcting through their collaboration with their peers. But more often, until a facilitator noticed what was happening, they would continue guessing randomly, seeming not to have clear understanding behind their movements. One researcher wrote in a set of observation notes, "Kids are just fiddling around … not completely understanding how the circuit works." This arbitrary work resulted in participants being unable to progress in their intended projects, whether because LEDs were continually burning out from not understanding the amount of voltage flowing through them or because the variables in their code were not correctly declared because they were just entering numbers into code without understanding where or why.

A screen recording of two participants, Denise and Ana, working together on some code demonstrated the haphazard way the participants began exploring the Arduino environment. After the lead facilitator, Ezra, had demonstrated the Blink Code to the participants, he asked all of them to "change some of these variables and see what happens." He explained how the code calls for certain pin numbers in the breadboard to be activated and suggests they change the values for the delay.

As shown in Figure 9, instead of changing the value assigned to the delay (1000) in order to change the blink pattern, the participants began changing where the variable "led" was supposed to be called to turn the LED on (HIGH) or off (LOW). They changed it to the pin number in the breadboard where their LED was plugged in (13). They did not understand how the variable should work with the LED they want to control. There was no rhyme or reason to how they were tinkering with the code.

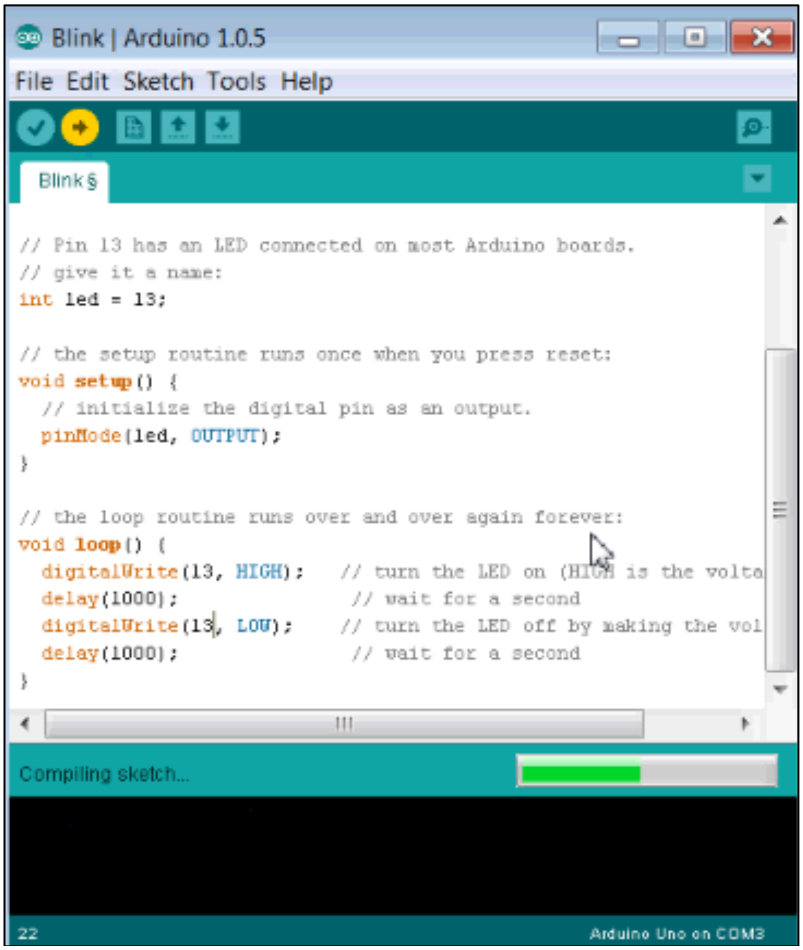


Figure 9. Denise and Ana change where variable "led" is meant to be called to turn on and off

The facilitators countered this futile messiness by utilizing cognitive apprenticeship methods, encouraging the participants to begin to make meaning out of their tinkering, and, by extension, use the tinkering as a vehicle to develop CT skills and dispositions. The tinkering then moved beyond randomly guessing to being intentional and a process for learning. Through modelling, coaching, and scaffolding, the facilitators begin embedding the activities with CT that the participants were not employing effectively on their own.

When Sam, a facilitator, happened upon the changes Denise and Ana had made, he began to coach them in using decomposition in how they do their tinkering. He pointing out how it was the LED plugged into pin 13. Denise then begins to realize the mistake but is still unclear on the path forward, asking, "This has to do with LED 13, if we change that and that, too?" Sam encouraged her to break down the code line by line asking her to look at where the pin is called: "So, what do you think the point of this line is?" Denise correctly responds, "To show where the LED is?" Sam states that she is right, and then asks her to look at the other lines of code: "But, what about down here also? 'Cause, it looks like what you've been doing is changing it here, changing it here, right?"

Sam then begins to step Denise and Ana through the code by asking them questions and providing hints to how they might adjust their code and what might happen as a result of those adjustments, scaffolding the skill of algorithmic thinking.

Denise: Would it, if we put 10,000 ... ?

Sam: Is it going to be longer or shorter?

Denise: Longer. Like, it's going to take longer to turn off and on.

Sam: Okay, let's see. Pretty long time, right? 'Cause, each 1,000 is a second. You've got 8,000s, 10,000s ... so, you can actually kind of predict how long it's going to be on. Exactly how long.

Denise: So, around 10 seconds.

Sam: Yeah. 'Cause, it's 10,000, right? So, what if you wanted it to like, be on for I don't know, 2.1 seconds?

Denise: 2,000.

Sam: About that, mm-hmm (affirmative).

Ana: And, you have to change both of them? All the time?

Sam: Well, that's an interesting question, 'cause what is this number controlling?

Denise: When it's on, and then when it's off. So, for it to be equal, like let's say if you want it to be under five, like one second, you leave it at 1,000. If you want it to be up at, for one second, so you leave it at 1,000. But, if you want to change it up, so I guess-

Sam: Do you want to change it up? It doesn't have to be on and off the same amount of time.

Ana: You gotta put one.

Denise: What did we put first [inaudible]?

Sam: So, you have 2,000, and 100? Okay. Huh.

Ana: So, it's off for ...

Sam: Very short. Mm-hmm (affirmative).

His questions push Denise and Ana to stop guessing and become more thoughtful about how they are adjusting the code and what those adjustments mean. Their tinkering becomes more methodical, and they tinker with the delay times, making the blink program work with two seconds on and .1 seconds off, as seen in Figure 10. Eventually, they realize that .1 seconds is imperceptible, and therefore, they need to make the delay longer to notice any blinking.

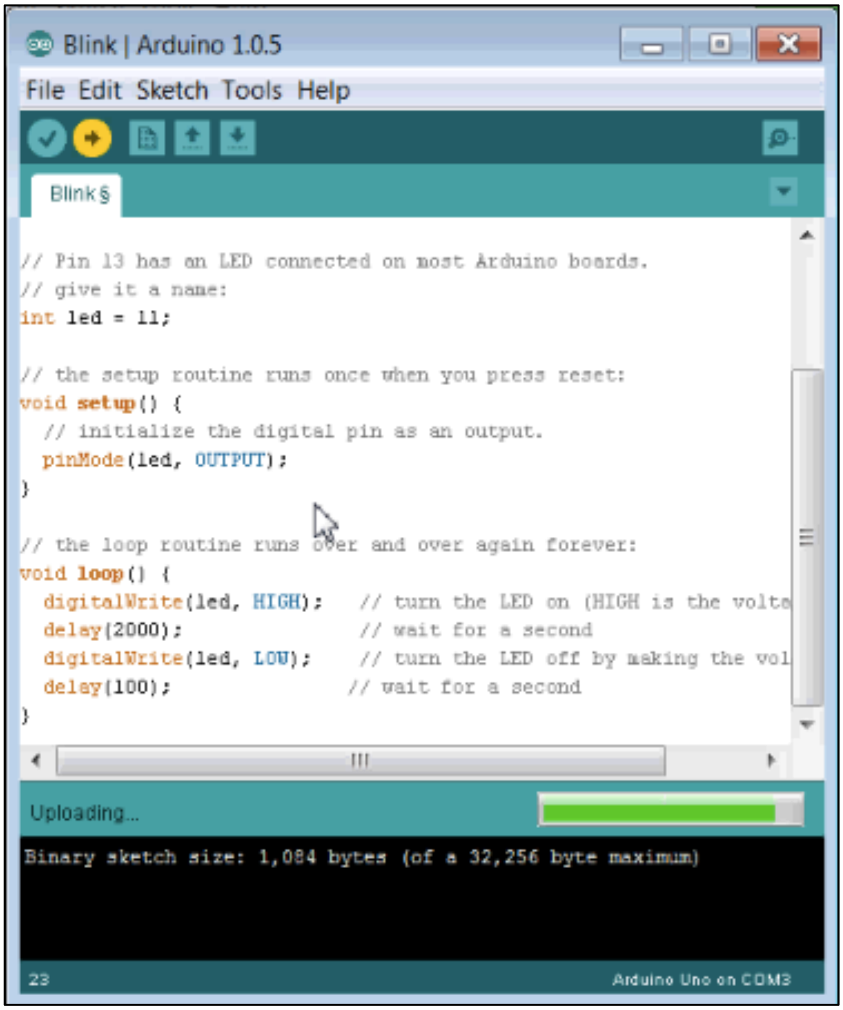


Figure 10. Denise and Ana change the delay times

An earlier instance of the modelling, scaffolding, and coaching of the CT skill of evaluation as a part of tinkering occurred during the tabletop breadboard activity. As a group of participants are moving the people in their tabletop breadboard around, Sam asked, “Is there anything unnecessary in this circuit?”, modelling an evaluation question regarding efficiency. Ezra jumped in, providing more scaffolding questions, “Yeah, can we get rid of jumpers? Will it work with less jumpers? Well, we have to connect things differently than right?” Julio, a participant posing as a jumper, acknowledged he was redundant, stating, "You can get rid of me." Ezra then coaches the participants in how to make Julio more useful in the circuit: "But if Julio wanted to be in the circuit in parallel, as an LED, how would he do it? Could you plug him in this row and then this row? If he put his hands anywhere in this row or this row, he'd be in parallel, right?"

Facilitators also modeled and coached CT dispositions. Juan, another facilitator, modeled tolerance for ambiguity for a pair of participants, Magdalena and Damaris, encouraging them to try out different components to see what they do and how they work.

> Juan: Gotta figure out which pin is which. Do we have another battery?
>
> Magdalena: We need 2 batteries? Or do I switch it?
>
> Juan: Uhhh … just add another battery. So, were you in that group when we were voting?
>
> Magdalena: yeah
>
> Juan: Oh! Did we just burn it out? Do it again. Close it. Close it.
>
> Magdalena: Are you sure?
>
> Juan: Yeah, keep it closed. Do you see that? What am I doing?
>
> Magdalena: Is that one polar specific?
>
> Juan: It's not polar specific, actually. It's called a potentiometer and you can control how much energy flows through it, and you can go one of two ways. You want to experiment with it a little more? It might work with an LED. But here. Use your hands. When you get to the area you want, make small little tiny movements, okay? And it's just what you said, a [inaudible] switch. Now go back and make very small changes.

Juan was unsure how to plug a potentiometer into a breadboard and was concerned he might have burned out a bulb in the process, but kept working with the students to get it working. Once they got it right, he points out how adjusting it slowly dims and brightens the bulb and then encourages further tinkering.

The other characteristic of note that arises from the sociological dimension of cognitive apprenticeship in tinkering is that of intrinsic motivation. For tinkering to take place, there needed to be an interest in wanting to experiment with new components and circuit designs. The ability to explore on their own was motivating to many of the participants. However, as previously mentioned, it was not motivating for all, some of whom seemed to have felt overwhelmed or uninterested in pushing themselves. One participant, who was capable of making complicated circuits, managed to make the LED on her Arduino blink early on in the activity and then stopped working and was doing nothing while everyone else was working, unmotivated to complicate her circuit until a facilitator came by and prodded her to do more.

The facilitators did not just work within the tinkering mindset to help participants develop more sophisticated learning strategies. Instead of just relying on participants arriving at an understanding of electricity or programming organically, they encouraged other practices to expand the participants' learning, namely, embodying, walkthroughs, drawing, and debugging.

Embodying

The participants had to use their own bodies and the bodies of their classmates as props to construct an algorithm for how electrons would move through their circuit, step-by-step. They would take on the roles of the components (e.g. making an elated face if they were a lightbulb and they were turning on) and clarify which sides of their bodies were attached to the positive terminal and the negative terminal.

When the participants were struggling, Ezra encouraged them to hold hands first, in order to make a basic circuit and clarify their roles in that way, and then plug into the breadboard retaining their roles. Both while they held hands and when they plugged in, a participant or a facilitator would often trace a path with their finger following along the participants arms and the table to show the flow of electrons while determining whether their circuit worked or not.

Because they each had a role to play in the circuit, they had to rely a great deal of communication and collaboration amongst themselves, as they thought through the structure and functioning of a circuit in a breadboard, the components it required to function, and how they all needed to play a part in it. The lack of external indicators for who was what component also required the facilitators to repeatedly prompt the participants with “tell me how this works”, “how is this working”, “explain the circuit as you go”, or “tell me about it”, encouraging constant articulation and allowing for formative assessment to happen continually.

For instance, Ezra, who was asked by the participants to take part in one group’s tabletop circuit because they needed another body, asks the participants, “Alright, so how’s this working?” in order to begin the articulation. One participant, Jackie, took the lead and walked him through the circuit, articulating how she sees the circuit working: “Okay so I'm the battery connecting the positive and negative. She's a jumper. He’s a ...” Ezra interrupted her to confirm that a participant that is playing the role of a jumper is still on a path that connects to him, and traced the path by pointing with his hand, as seen in Figure 11. Jackie confirmed that this was correct, and then Ezra encouraged her to continue articulating the path. She did, and then other participants began to jump in when she got stuck and assist in her explanation of the circuit.

Figure 11. Ezra demonstrating where the path of the electrons along the participants' bodies could go

The constant prompting to clearly explain why their bodies were in a certain arrangement established a habit of articulating their algorithmic thinking. The participants had to repeatedly talk through their understanding of the circuit and how the breadboard worked to establish it amongst themselves and then communicate it with the facilitator. The hand holding worked as a scaffold, for participants to better grasp how their circuit worked before they plugged it into the tabletop breadboard.

The embodiment practice relied on the cognitive apprenticeship sociological dimension of exploiting cooperation in order to foster group problem solving. The fun of using their bodies as large components in a breadboard was motivating to participants. Participants who were normally quieter took on leadership roles in directing how the group should place themselves, the lightheartedness of the activity encouraging them to let their guards down. Because they were all in the problem together, they had a better understanding of where their peers may have gotten confused and were able to physically demonstrate how their circuit worked or did not work.

Walkthroughs

The traced articulation from the tabletop breadboard activity then extended throughout the day. Facilitators kept using phrases like “talk me through it” “what do you mean” or “let’s hear it”, prodding participants to explain themselves in detail, and therefore pushing their thinking. This was especially true in developing their algorithmic thinking skills, constructing sequences in order to solve problems or better understand situations. As they begin to explain the paths electrons take in their circuit, they begin to see how their original thinking about the problem was confused. This thinking aloud causes them to reevaluate how their algorithm functioned or should function.

In the tabletop breadboard activity, the participants developed these walkthroughs as a group. In the breadboard exploration and the Arduino activities, participants continued the practice as individuals or pairs at the request of facilitators. They were comfortable giving a tour of their project, even if they were unclear on how it was supposed to work.

An example of articulation of for decomposition occurred when a participant, Angela, called over David, another facilitator, when she was stuck trying to redesign her circuit in the breadboard exploration activity, and the articulation he elicited from her helped clarify the problem.

> Angela: I want to do it parallel.
>
> David: To make it from that? When you say connect these two what do you mean? Let's hear it.
>
> Angela: Because this is one circuit, this is another. I want to connect both of them to control it with this [inaudible]
>
> David: Okay, so you wanna create a parallel circuit, where one part is controlling an LED but turning on with a button and the other one lights up a light, I'm assuming.
>
> Angela: Ohhhh, that's the other one. Oh. These are still with the switch, though. The one that slides to make the change. That's the one we used last time.

By asking her to articulate her thinking ("Let's hear it") David got Angela to explain that she had two circuits she wanted to combine as one parallel circuit controlled by a button. David rephrased what he thought she was trying to do back to her. She realized from what he described as his understanding that she had misidentified what the two parts of her circuit were. She was then able to better identify the components of her circuit.

In the same activity, another participant, Jen, did a walkthrough for understanding for Juan when she was tinkering with a new component, a capacitor, in her circuit. Juan asked her to describe in detail how her circuit works, "Alright, so your capacitor, how is your capacitor flow? Why don't you talk me through it?" She was initially unsure, so he prompted her again, starting it off for her: "Talk me through it. Your battery starts here right?" Jen then to worked her way through the circuit aloud, component by component: "Starts to the battery, which goes to another battery, which goes to the switch, which goes to the [inaudible], which goes to the capacitor, which goes to the LED, the LED goes to the resistor ..." She was tripped up by the capacitor, so Juan redirected her, "But instead of the LED, let's go to the capacitor, it goes and it's connected there." Juan then helped her draw the connections leading to the capacitor in a way that clarified how each component fit into her circuit, walking through each step.

In the Arduino activity, Sam asked Denise and Ana to talk through their program from the beginning. Denise started tracing through what the program was doing, "Okay, we're gonna turn off. Then we go back to high. And, high stays on." Ana stepped in, "Right there it's blinking." Denise was confused: "But, then we'll return, doesn't that go high?" Ana was agreed with her confusion, "It just turned off." Sam then asked them to think through what was happening and to state it out loud: "Why do you think that is?" Denise responded, "Because, right here, it turns the LED off, by making the, I don't know what thing that is, but ... the voltage hold." Sam pushed on to get her to describe what was happening in that moment as she was walking through the program: "Ah-hah. So, the high and low is referring to what?" and Denise articulated, "On and off." Once Denise made it clear she saw how the variables worked, it gave Sam the opportunity to make the connection between the exploratory work they did with the breadboard (changing the voltage with resistors) and the work they were doing with the Arduino (changing the voltage with code).

> Sam: Yeah. Exactly. The voltage that's going to it. The computer is handling all the details and figuring what voltage to send, right? So, if you think about what you guys were doing earlier, you had to connect different numbers with batteries, right? To change the voltage. Well, this is basically, like, there's little people in the computer doing that for you. And, they're sending how much voltage it needs to turn on. And then, taking it away, to turn it off. But, that's the nice thing about the code, is that it has this kind of stored variable, high and low, so that it makes it a lot simpler to just change it on the fly.

The tours participants gave of their work encouraged a practice of articulating decomposition and algorithmic thinking, as participants had to be specific in what they thought the different parts of their circuit should be doing. In having to express with precision the path for their projects, participants were highlighting places of misunderstanding or confusion, as well as establishing new understanding for how their projects worked. In terms of cognitive apprenticeship, walkthroughs not only exploited cooperation through forcing dialogue, but also brought forth control strategies for the participants, where they used a walkthrough as a way of generating alternative courses of action by thinking the problem through aloud with a facilitator or peer. Walkthroughs also helped them evaluate what steps would get them closer to a solution, as they were verbally checking in with every step of their process.

Drawing

Drawing also served as a way for participants to further clarify their thinking. Participants had been asked to draw their circuits in the schematic format from the first day in the program. As the moments observed were on the seventh day of the program, they sometimes drew them out of habit, but often still had to be prodded. The drawing of circuits focused participants on removing unnecessary information when determining the structure of their circuit and how it worked. The physical location of a component on the breadboard sometimes confused participants as to where the component existed in the electrical path of a circuit, so the drawing forced them to better understand how the circuit worked.

Another exchange between Angela and David demonstrated how drawing served to clarify CT. Angela called David over because she was unclear where to draw an LED that she had put on her breadboard circuit: "How would I draw this over this? Like I have the connection. But they are like 2 series that are separated, and this is like the connection. I don't know how they are separated. Like, where would I put this?" David talked it through with her, and asked where the positive and negative ends of her battery were. Angela clarified how the polarity of her circuit worked. Then, David showed her that although her circuit looked like it was a series circuit, it was actually parallel and should be represented as such: "Even though it's one line it's not still parallel. Do you see why? So, draw it as a parallel." Angela then erases the section of her drawing that represented the circuit as a series circuit, and redrew it as a parallel circuit on top of the old drawing. She then redrew the circuit next to the old drawing, abstracting it more and removing the unnecessary drawings of the jumper cables she had on the first drawing, as shown in Figure 12.

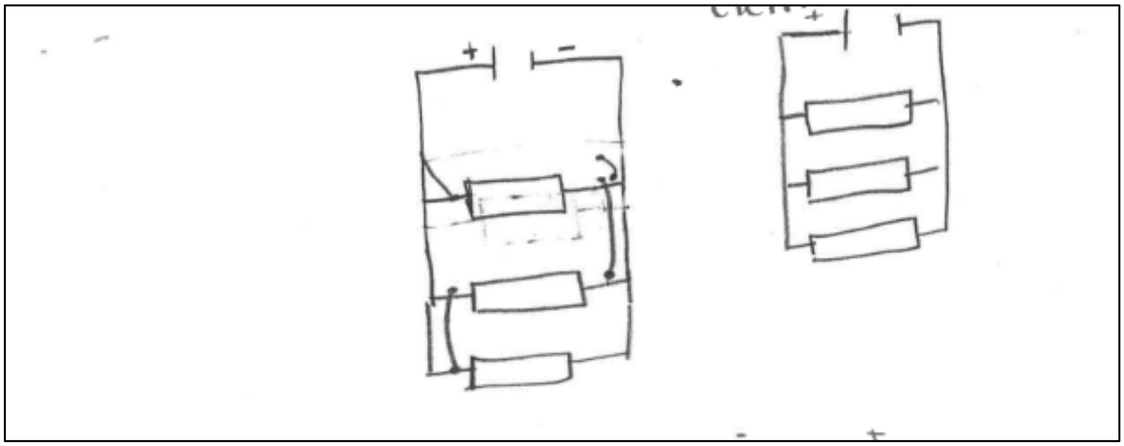


Figure 12. Angela's first drawing of her circuit (left) and her second, more abstracted drawing (right)

The interaction between Angela and David demonstrates how drawing was used as a way to coach participants to think more abstractly about their designs, removing whatever information was unnecessary (how the wires were physically placed on the board, how long they were, etc.), and think through what were the important elements of the circuit and how it was structured. Drawing was a way for facilitators to guide participants in articulating how they saw their circuit, and how they thought it worked, as David did with Angela in helping her see her design was a parallel circuit. Using the drawing as an "object-to-think-with-together" (Stevens, 2011) aided participants in communicating their understanding of the circuit with the facilitators.

Drawing the circuits let facilitators know whether the participants understood if the circuit was series or parallel and what the direction and flow of electrons were in the circuit. Otherwise, participants could just be plugging components into the breadboard and get lucky with a functioning circuit, without actually understanding why it worked or how to add onto it.

Drawing was also a type of cognitive apprenticeship control strategy, as participants and facilitators used it as a way to manage the process of carrying out the task. The participants sometimes used the drawing as another perspective on the problem-solving process, helping them determine how to move forward. They utilized both the monitoring and diagnostic components of control strategies, as participants and facilitators could use the drawings to evaluate their progress in understanding circuits, but also as an analysis of the cause and nature of problems they might be having.

An example is the previously mentioned circuit with the capacitor Jen was creating (Figure 13). In representing the circuit as a drawing, Jen was confused as to how to order the components in her diagram: "I put in the resistors, but I don't know where to go from there." At first, Juan did not see what she was confused about, and then understood that the placement of her LED was what was puzzling her. Once he helped her with that he pointed out she then had to determine where her resistor needed to go, working through her drawing little by little.

> Juan: Is this your ground? And this goes to your switch? It stays on and then comes off. The push button is cool, that little effect. Alright, so the ground goes directly to … is this power or ground? This is power. So, the power goes to the switch, the switch ... I'm ignoring the capacitor right now, because that's almost like another battery. But it is on the other side of the switch, I'm just trying to make sure this is right. So actually, it goes the other way, because the power goes here. And first come the LEDs.
>
> Jen: So, it should be the LEDs and then the capacitor.

Juan: Correct, that's what you've got going on. But your resistor isn't going anywhere. So, your resistor is actually … it's not doing anything. Follow it with … Ignore the capacitor, alright? So, it goes to your LED.

Talking through the drawing with her, Juan demonstrates that after the battery and the button, the LED comes first, then with an external connection to the capacitor (Figure 13). In this way, Juan is facilitating the understanding of the algorithm, the ordering of the steps of the circuit through the drawing, which Jen eventually completes, as seen in Figure 14.

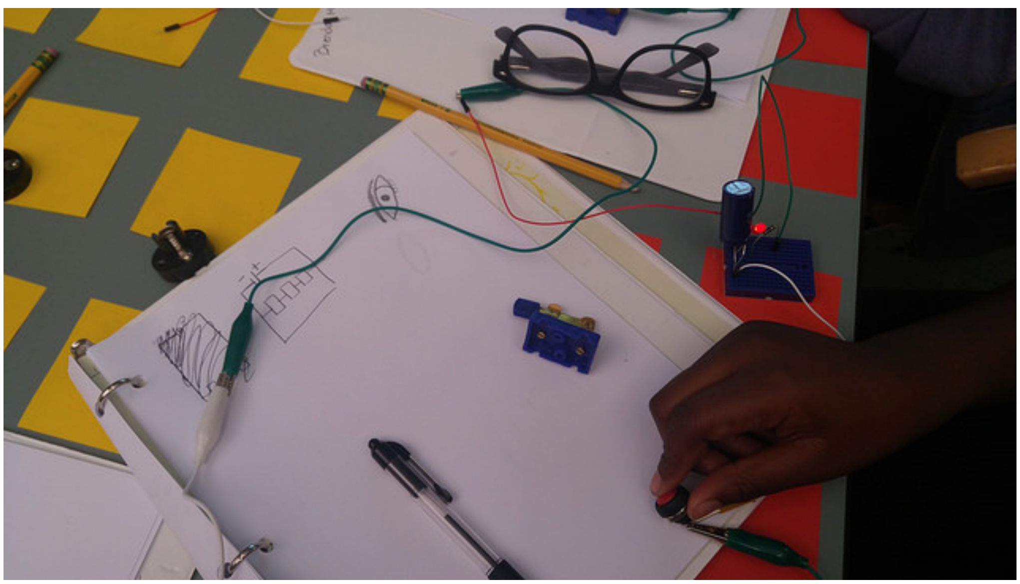


Figure 13. Jen's circuit

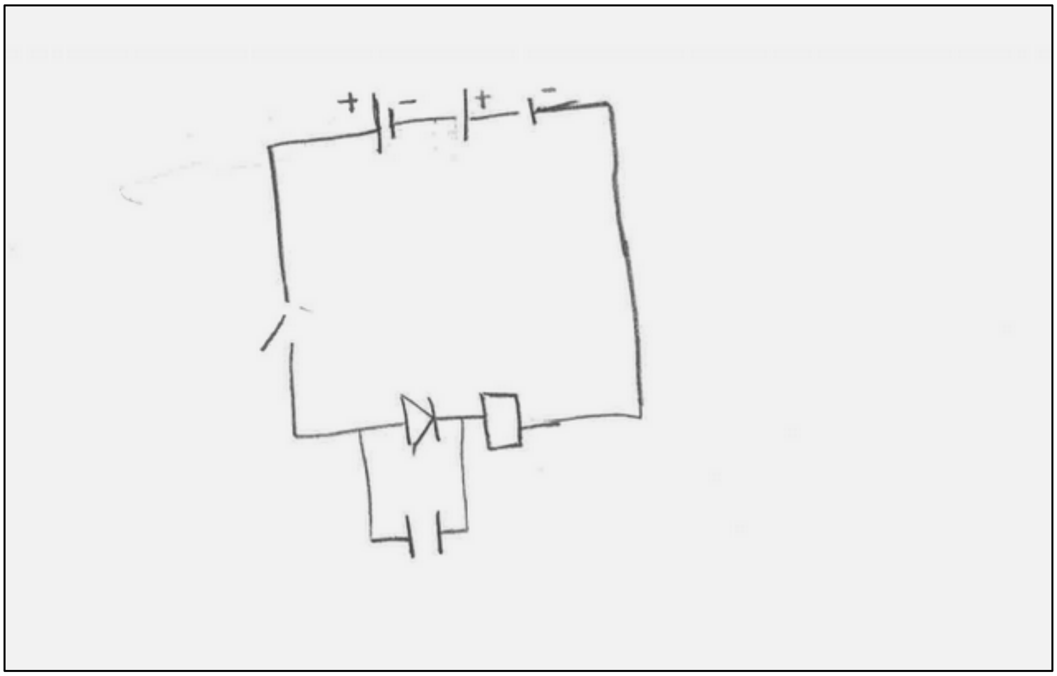


Figure 14. Jen's final drawing for her circuit

Jen's drawing was also a demonstration how drawing enabled the cognitive apprenticeship sequencing strategies of increasing complexity and increasing diversity. By using the basic schematic of a circuit as a foundation to build upon, the facilitators introduced the breadboard and then, through drawing, described how the rows and columns translate into a circuit. As the participants were introduced to new components, such as potentiometers or capacitors, they learned the schematic symbol for it and then devised new ways of using them in increasingly more complex circuits.

Another cognitive apprenticeship method that was documented alongside drawing was reflection. Facilitators discussed how drawing circuits is what is used in industry to communicate a circuit design to someone else, and that the abstraction makes the design clear to other people.

At one point, Rafa, a participant, asked why he had to use the standard circuit drawing symbols and could not just use his own abstraction of the circuits.

> Ezra: So yeah, somebody mentioned that they prefer to use their own symbols. Yeah, if they use their own symbols is that still an abstraction?
>
> Group: No, nope.
>
> Ezra: Actually, it is, it is an abstraction. What's the problem with it, though?
>
> Group: No one can read it.
>
> Ezra: No one can read it. How can you make it that we're able to read it?
>
> Rafa: Add a key to my language.
>
> Ezra: You could add a key, yeah. But we want you also to become familiar with the internationally recognized symbols for these electrical parts.

Ezra wanted to clarify that in order to communicate effectively and efficiently in industry, a standard style of diagram, or abstraction, was necessary. Facilitators mentioned how abstraction helps experts plan their larger designs, allowing for complex solutions to scale to larger projects because only the important information was expressed. In other words, facilitators used the drawing as a way of drawing participants into a culture of expert practice, teaching them to think about and carry out the design of a circuit in the way that an expert would.

Debugging

Where drawing served as a sort of map for the tinkering, debugging provided an engine for it to continue. The clear structure of identifying a problem, determining the various possible reasons for the problem, and then coming up with solutions lent itself to various targeted instructional opportunities for CT that happened alongside the playful nature of tinkering and prevented it from getting frustrating or confusing.

The CT practice most commonly noted with debugging was decomposition. For participants to effectively debug, they needed to understand what the different parts of the problem were, and isolate each part to see where things were not working. Facilitators made sure they were aware of all of the different parts of the problem, prodding them to think of all angles.

For instance, Denise was struggling with the same LED in her circuit blowing out repeatedly, and a researcher called over Sam to help her debug the problem.

Researcher: Why do they keep blowing out?

Sam: Oop.

Denise: And with this one doesn't happen. It's always the first one.

Sam: They're blowing out as they are set up right now?

Denise: Uh, yes, but it's always happening with the first one [inaudible]

Sam: Uh huh. Ah, so that one gets blown out.

Denise: Yeah.

Sam: Okay, so where is your power coming in?

Denise: From …

Sam: Through here?

Denise: Yeah.

Sam: Okay. And where does it go from there?

Denise: Well, it goes out this way and continues this way.

> Sam: So, it's got 2 options. Right? So what kind of a circuit is that?
>
> Denise: Parallel.

Sam helped her break down the problem by analyzing the electricity flow of her circuit arrangement, making her realize that her circuit was parallel and not series, and therefore all of the voltage was going through the LED instead of passing through a resistor she had on a parallel arm of the circuit.

Ezra often modeled debugging practices, voicing aloud what was confusing him, decomposing what all of the possibilities for the problem might be, what he had tried, and what he was about to try. He also projected code on the screen at the front of the room and talked through his thinking with the participants. The other facilitators, when discussing participant artifacts that they were struggling in assisting with, would also describe their thinking processes to one another in front of the participants.

For example, at one point, Angela asked Ezra for help on the circuit she was working on. It was wired in parallel and a switch on a different line was affecting another area of the circuit. Ezra systematically isolated variables, like the breadboard row the switch was on to see what the issue might be, but it began to confound him.

> Ezra: [to Angela] It's a little weird. Because this shouldn't affect this. Once you have it parallel, it should close the circuit. It's kind of weird. Let's say you move this switch to another row, over here and then you took that leg of the LED and plugged it into the same row. Now, the LED is off. Yeah, this is totally weird. I have no idea how this is working. [puts his hands on the board] That is very strange. Well, there is something I don't know about it, so that's cool. But yeah

this is the closest approximation I could make to that. I don't know why it doesn't work the other way.

Ezra could not understand what was going on and brought in another facilitator, David, to take a look, articulating all of the steps he tried. In front of Angela, they discussed their confusion and the different avenues they tried to make sense of it.

Ezra: Do you feel like this is kind of weird that it's working like this? Because this switch is in parallel so it should be on its own. It shouldn't really be affecting anything. They're all in the same row, so I feel like the switch shouldn't be affecting them.

David: It shouldn't.

Ezra: And then when I move this over and then move the switch over to the same line, this just doesn't light up. Like it should light up if I move the LED into another row and move the switch into that row so that it's wired normally, it doesn't work.

David: [To Angela] You defied logic, because we can't get this [laughter].

Ezra: Isn't that weird?

Eventually, by bringing in Juan and discussing it with him, they realized that the circuit had a short circuit.

Ezra also consistently modeled the defining disposition associated with debugging, which was "persistence in dealing with difficult problems". The act of debugging demands sitting with a problem and plugging away at the different solutions, requiring both time and perseverance. Participants followed Ezra's lead. A participant who was having a problem with a circuit on her breadboard and stayed well after it was time to go to lunch as she tried reconfiguring her setup. Others would stare for several minutes at their Arduino code that was not working, looking for what could have gone wrong.

Facilitators often used coaching as a cognitive apprenticeship method in conjunction with debugging, as coaching functioned as a "just in time" method, concerning specific problems as they arise. When participants encountered a problem, such as an LED that did not blink the way they wanted it to, or a capacitor that would not hold a charge, facilitators focused their attention to what might be the causes of the problem, or remind them of what they had learned in regards to parallel circuits or voltage in the previous sessions of the program.

The instances that were coded as debugging also included the cognitive apprenticeship content strategies of domain knowledge and problem-solving strategies and heuristics. The facilitators would point out domain content as they went through the debugging process with the participant, reminding them of the characteristics of a parallel circuit or how a breadboard works. They would also focus on "tricks of the trade", making sure the participants followed what the path of a circuit was in the way they had arranged it, or moving components on a breadboard to see if the problem encountered in one part of the circuit would happen in another part.

For instance, Sam was helping Becky with her circuit that was not working by talking through the path it followed.

> Sam: So, let's follow the path that has the light. So, we come here, again, we branch out here. So now, what's happening? Well, the only other thing in this row is this resistor.
>
> Becky: Mhhmm
>
> Sam: But that comes back to where we started right? So where is this circuit getting broken off? Where are the electrons hitting a wall and not being able to continue?
>
> Becky: Is it right here?
>
> Sam: Right here, yeah, in this stem of the light, right? 'Cause they get here, but then they have got nowhere to go. This is a dead end.

He physically traced the path with his finger and then recalls information they had learned about electrons and how they move, inserting domain knowledge as they debug the problem.

More purposeful tinkering

By the end of the session, in which participants are finishing up working with the Arduino and brainstorming ideas for their final project, the tinkering matured with the facilitators moving from doing less modelling, coaching, and scaffolding, and allowing for more exploration, with participants incorporating CT into more intentional tinkering on their own. They still needed to confer with one another, but the conversations were purposeful and with direction. They were "learning how to learn," looking for instructions online before asking the facilitators for help, and building on the patterns of what they have already learned. For instance, a pair of participants, Damaris and Magdalena, wanted to extend beyond the Blink program and try to make the LED fade in and out. Following the pattern of the first program, but just changing a couple of lines of code they found online, they manage to make it work without any facilitator assistance.

The participants also started showing signs of being more methodical when looking for errors in the code; instead of just randomly punching in numbers in new spots, they looked at other examples and compare and contrast the code line by line. After five minutes of struggling with a piece of code that didn't work, a participant, Elina, compared the code line by line with an example piece of code and found the problem in syntax. She explained to David where there was a missing parenthesis, after David tried to help her and could not.

Participants also began to get comfortable with the Use-Modify-Create progression (Lee et al. 2011), borrowing code and changing it to fit their purposes, experimenting with what's possible, taking risks, and trying a different tack if it did not work. For example, Veronica and Rafa searched online to find code to make a piezo buzzer play music with their Arduino. Then, employing the Use-Modify-Create progression, they built upon the experience and planned to create a light display that was synchronized to music, with several LEDs changing depending on the octave or volume of the note. They decomposed the entire project to make sure they had all of the materials they needed. They drew a schematic of how it would work, including the placement of the piezo buzzer and the lights, and sketched out how all of the pieces would work together, as seen in Figure 15.

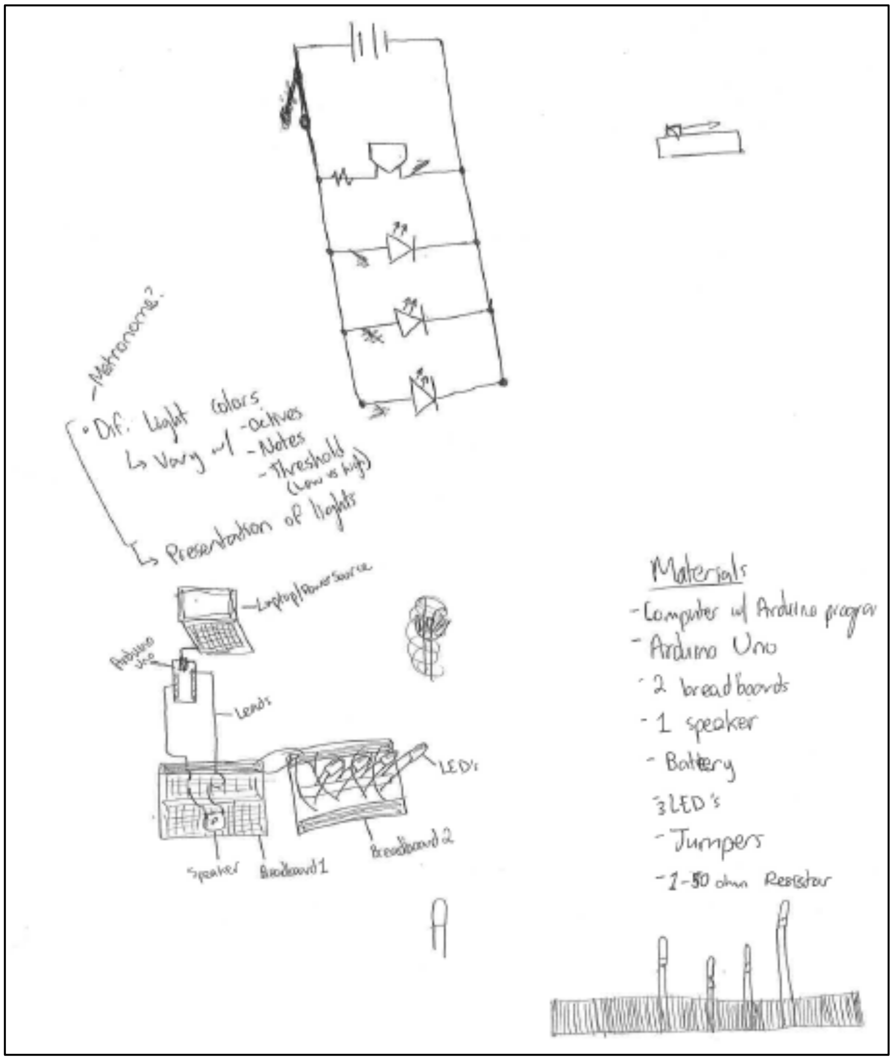


Figure 15. Planning for Veronica and Rafa's music and light display

The participants' ability to abstract information became useful in planning their final projects, as they were able to determine what was important for functionality. For instance, Denise and Ana used Ana's notebook to plan out their final project. They decided to make a teddy bear that lights up with different LEDs. They drew the bear and the different components they wanted it to have (Figure 16). They then created a schematic for each body part for which they wanted functionality, outlining how each part would work. For the head, they drew a series circuit with a switch, battery, and two lightbulbs for the eyes (Figure 17), and for the arm of the bear, the participants included a solenoid, to make the bears arm move (Figure 18). The abstractions helped them determine the materials they needed and how they were going to power each part.

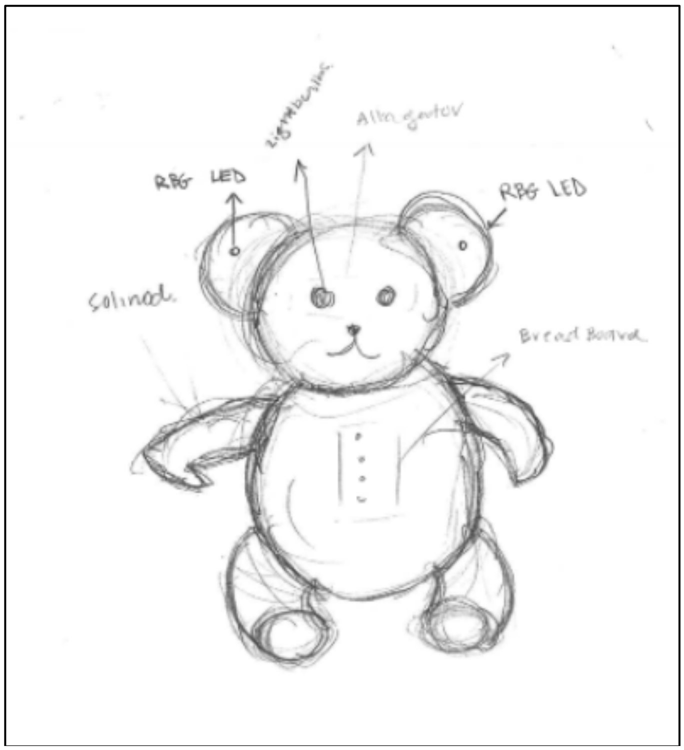


Figure 16. Denise and Ana’s drawing of bear as a whole

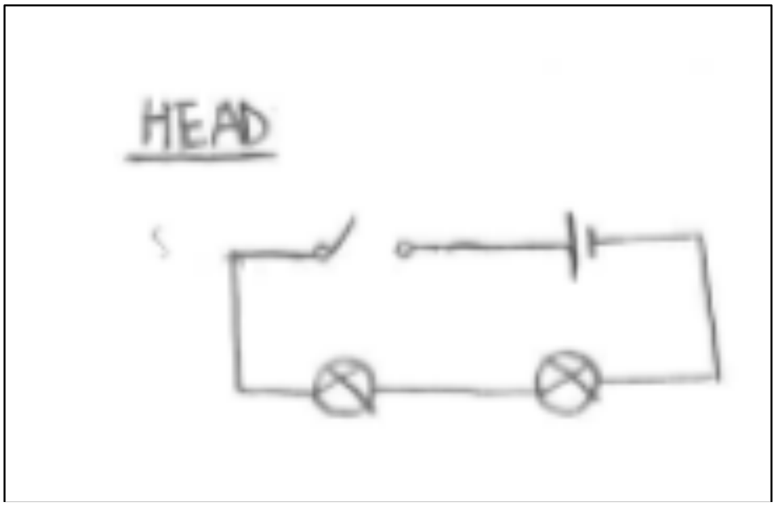


Figure 17. Schematic of head

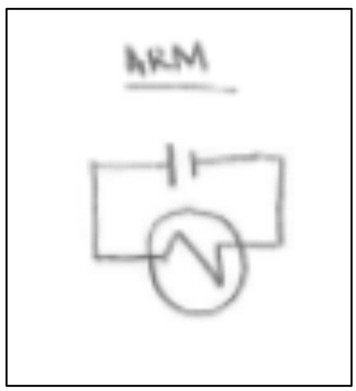


Figure 18. Schematic of arm

The repetition of the practices throughout the tabletop breadboard, the breadboard exploration, and the Arduino exploration activities developed a thread of the CT concepts and dispositions throughout, and established the first two non-computing activities as scaffolding for the final computing activity. When it came time for participants to plan for the final project they would work on the next day, they had the "habits of mind" (Winner, Hetland, Veenema, Sheridan, Palmer, & Locher, 2006) that helped them use CT skills and dispositions for planning and developing.

V. DISCUSSION

In this chapter, I will briefly summarize the study, connect the findings to implications and the literature, discuss the limitations of the study, and suggest potential directions for future studies.

Summary of the Study

The purpose of this study was to examine what evidence exists regarding promising practices that support youth development of CT skills and dispositions within makerspace activities. This study was a single case study of the seventh day in a nine-day summer program where high school students were learning physics and engineering topics in a makerspace environment while developing their computational thinking (CT) skills and dispositions.

The day was divided into three main activities: First, in order to address confusion that arose from the previous day when participants were introduced to breadboards, the lead facilitator had them create large circuits using a table as a stand-in for a breadboard and post-it notes to indicate open connections on the breadboard. The participants then played the roles of various components of a circuit and worked in groups to build series and parallel circuits on the table in front of them. Second, participants were given actual breadboards and a variety of components that they had and had not seen before and were asked to explore and create various kinds of circuits. Then, after being introduced to the Arduino microcontroller, participants were encouraged to explore with the interaction of circuitry and electricity. Finally, as the participants were completing the Arduino activity, participants were asked to begin thinking about their final project, which was entirely up to them, as long as it used some of the knowledge and materials they utilized in the past seven days.

Data on what occurred during the day was collected through participant notebooks, audio, video, and screen recordings, photographs, and observation field notes. Audio recordings were transcribed. The data was then analyzed by coding the transcripts, observation notes, screen grabs of the screen recordings, and the participant notebooks. While several themes arose from the coding, the five that were the most frequent and seemed to have the most impact were tinkering, embodiment, walkthroughs, drawing, and debugging. The activities were also coded for CT skills and dispositions and cognitive apprenticeship methods. When using specific instances as examples, cognitive apprenticeship content, sequencing, and sociology were also considered.

Tinkering was the most prevalent theme, with participants encouraged to be creative and explore in all of the activities. However, it also presented problems when participants lacked motivation, direction, or understanding of the problem. Therefore, on its own, tinkering could not consistently elicit CT skills and dispositions.

By promoting various forms of expression for participants to "think aloud" their understanding in conjunction with cognitive apprenticeship methods, i.e. embodiment/walkthroughs, debugging, and drawing, the facilitators were able to encourage more strategic and focused tinkering to produce artifacts that both met the participants' goals for what they wanted to create and the facilitators' goals for what they wanted them to learn, as conceptualized in Figure 19.

Where the cognitive apprenticeship methods are placed on Figure 19 is only where they were most commonly observed, not that they were only observed during those practices, or that those practices did not coincide with other methods. For example, other methods besides articulation were utilized during embodiment/walkthrough and drawing practices, and articulation was used in the other practices, but articulation was most utilized during embodiment/walkthrough and drawing practices.

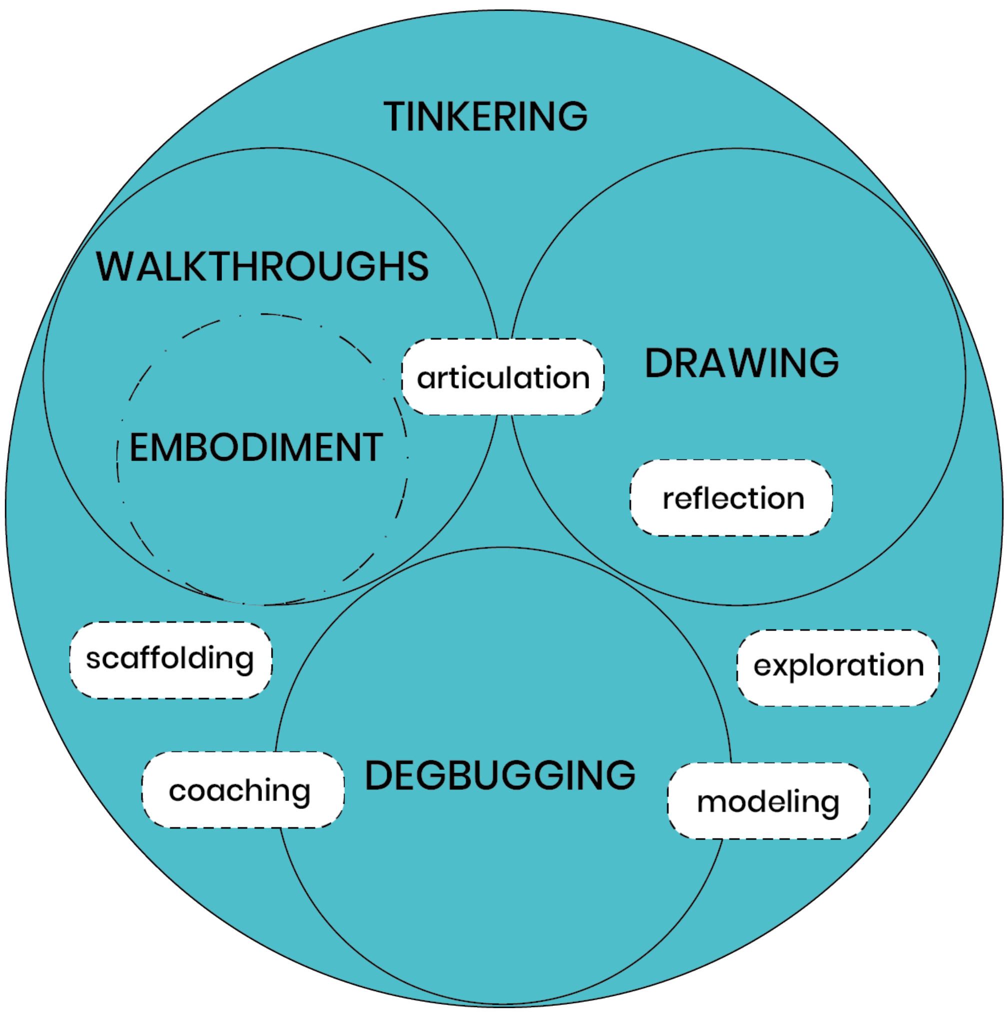


Figure 19. Practices and associated cognitive apprenticeship methods that supported the development of CT skills and dispositions in the observed maker activities

Embodiment served to present breadboard circuits in a larger, physical way that participants could identify with and collaborate on. This embodiment generated a habit of walkthroughs, with participants becoming comfortable with articulating the path of electrons through their breadboard circuit, and then beginning to articulate the sequence of their code in their Arduino projects, throughout the tinkering process.

Drawing schematics of their circuits also provided opportunities for articulation, while encouraging the development of the skill of abstraction, helping participants to focus on what was important in their circuit and think more strategically on how their tinkering should move forward. The facilitators' focus on drawing provided the participants with objects-to-think-with as they talked them through their process, as well as focusing their attention on the abstraction of problem representation. Facilitators also used drawing as reflection, connecting the student work to that of industry.

Once their circuit or their program was put together, the practice of debugging promoted constant inquiry and refinement as facilitators utilized modelling and coaching methods, instituting a more systematic process to the tinkering. Debugging served to help participants analyze and evaluate their work continually throughout the activities.

Through embodiment, walkthroughs, drawing, and debugging, in conjunction with the utilization of the cognitive apprenticeship methods of scaffolding and exploration, participants' tinkering became more sophisticated, as did demonstrations of their CT skills and dispositions.

Implications and Connections to Literature

Tinkering

In contrast to an adherence to rigorous and methodical testing of the validity of theories and hypotheses (Lamers, Verbeek, & van der Putten, 2013), Vossoughi and Bevan (2014) describe how an environment that promotes making and tinkering (a) connects students to familiar practices from their home, school, and community, allowing for more authorship and agency in their work; (b) provides context to STEM concepts in relevant activities that connect to other subject matter; and (c) encourages collaboration and a fluidity of roles between more expert and novice group. In addition, Berland (2016) describes tinkering in computing as an expert practice that novices can do, connecting them with the expert community of practice relatively immediately (Berland, 2016). In these ways, the atheoretic nature of tinkering, i.e. the lack of need for a plan while working on a computer program (Berland, 2016), exists as part of the draw to the tinkering mindset in making. The playfulness taps into a student's desire for an immediate sense of agency. Tinkering is analogous to the CT practice of Use-Modify-Create (Lee, et al., 2011), whereby students' continual modification of an artifact results in a new creation in and of itself.

Yet, tinkering without appropriate modeling, coaching, and scaffolding can lead to a lack of depth in understanding underlying concepts and possibly frustration. Mirroring what was observed in this study, McDermott and Shaffer (1992) found that students' successful completion of circuit design tasks was not correlated with the conceptual understanding of underlying concepts of electricity. Worse, if students are not reflective about their experiences, they may reinforce incorrect suppositions (Bransford, Brown, & Cocking, 2000).

Tinkering is also problematic in that, although it can be fun and attractive to many students, it is not initially comfortable to all, as found in this study. Research on tinkering show females have less exposure to it and are less inclined to do it. “Exploratory learning styles” or “tinkering behaviors” have been shown to be demonstrated more predominantly in males than females in both computer science and more broadly (Parsons, 1995; Beckwith, et al., 2006; Hou, et al., 2006; Krieger, Allen, & Rawn, 2015).

Therefore, in a making environment where tinkering is promoted, instructors need to provide (1) explicit and extensive guidance in the selection, organization, and application of CT while tinkering (Young, 2002) and (2) an environment that encourages and models tinkering behavior for all (Krieger, Allen, & Rawn, 2015). This guidance can happen through the intentional application practices that connect making to CT, using cognitive apprenticeship. This study found the practices of embodiment, walkthroughs, drawing, and debugging to function as this type of connection between making and CT and that the cognitive apprenticeship made this conceptual knowledge more explicit.

Embodiment

Embodiment has been shown to be a powerful learning approach in learning abstract domains, knowledge representation and the bodily states (Barsalou, 2010). Kinesthetic involvement of learning material can be aligned with conceptual knowledge to construct conceptual metaphors from concrete sensorimotor experiences.

(Abrahamson & Trninic 2015; Bamberger & DiSessa 2003; Sung, Ahn, & Black, 2017). Exploring a physical space in communion with a conceptually grounded one provides a familiar environment that serves as an anchor for learning to develop (Sung, Ahn, & Black, 2017). Therefore, an embodied approach is useful for teaching and learning unfamiliar abstract concepts, especially in STEM, as demonstrated through research on embodied cognition and gears (Schwartz & Black, 1996), robotics (De Koning & Tabbers, 2011; Lu et al., 2011; Sung, Ahn, & Black, 2017), and programming (Fadjo, 2012; Bell, Alexander, Freeman, & Grimley, 2009).

For an instructor, embodied approaches provide hooks on which to hang cognitive apprenticeship, especially in regards to situated learning. As suggested by Rambusch and Ziemke (2005) utilizing the body in situated learning activities (1) provides common ground by which to explore more abstract understandings; (2) serves as a means interaction, allowing for the subtleties of gesture, tone, and body language to further inform; and (3) is the foundation for human thought, providing instructors an opportunity to use embodied activities as a way to generate new understandings of thinking and knowing (Rambusch & Ziemke, 2005).

While making is already an embodied activity, as makers construct knowledge through the development of their artifacts, it is generally a type of “surrogate embodiment”, where learners manipulate an external surrogate, not their own bodies (Fadjo, 2012; Sung, Ahn, & Black, 2017). Developing activities within a making environment that provide opportunities for “direct embodiment”, where learners embody themselves as agents executing coding commands or completing a sequence, has shown to help students better grasp conceptual understandings and abstract ideas like (programming approaches) than that of surrogate embodiment. Facilitators may consider embedding these types of activities throughout time devoted to making and tinkering to expand and deepen the understanding of CT concepts.

Walkthroughs

The walkthrough is an informal software developer technique used by development teams as a form of the CT skill of evaluation, where the focus is generally on finding defects or potential opportunities for improvement (Blum, 1992). In addition, by centering on the sequence and process of the artifact to the forefront, walkthroughs encourage the articulation of algorithmic thinking. They function as a way to describe to the group what the thinking was in the structure of the software and why had been designed in that way (Blum, 1992). In this sense, walkthroughs can function as a type of think-aloud, a method of verbalizing thoughts that has been used in reading comprehension to model comprehension strategies (Davey, 1983; Block & Israel, 2004). Cognitive theory demonstrates the need for students to have task-related conversations with others, and studies in mathematics education have shown that task-oriented speech was positively correlated with metacognition and successful task completion (Schoenfeld, 1985; Ostad & Sorenson, 2007).

In addition, walkthroughs encourage students to share their own thinking, developing community around ideas, while making their thoughts explicit (Raihan, 2011). These process-oriented types of interactions also help students learn how to defend their own ideas and be in dialogue with the ideas of others, encouraging the integration of new knowledge into their existing knowledge base (Silver, 1985; Johnson, & Fischbach, 1992). Working through problems in dialogue with one another is a utilization of the cognitive apprenticeship practice of exploiting cooperation, providing students with more scaffolding by distributing knowledge throughout the group (Collins, Brown, Newman, 1989). By integrating walkthroughs as part of the culture of learning, facilitators can bring forth a focus on procedural thinking and the importance of dialogue to the construction of knowledge.

<u>Drawing</u>

Encouraging drawing in a making environment prepares students for future learning, because it helps them determine the key features of new tasks, encouraging students to prioritize what information is important, and what information is noise that only confuses the point (Schwartz & Martin, 2004; Abrahamson & Trninic, 2015). Studies have shown that students have better results on near transfer problems when they use abstracted diagrams of circuits, then when they do not use diagrams (Moreno, Ozogul, & Reisslein, 2011; Moreno, Reisslein, & Ozogul, 2009; Johnson, Butcher, Ozogul, & Reisslein, 2014).

Incorporating drawing engenders the cognitive apprenticeship practice of reflection by bringing students into the scientific community's practice of continually visually representing problems. Research has shown that representing problems visually assists in both processing a problem and resolving it for both novices and experts (Brenner et al., 1997; Collins & Ferguson, 1993; Rittle-Johnson et al., 2001; Zhang, 1997). In order to test ideas, make discoveries, elaborate knowledge, and explain their findings, scientists often rely on visualizations, diagrams, and graphs (Gilbert, 2005; Abrahamson & Trninic, 2015). This reliance on visual representation for sense-making extends to the field of computer science, where experienced programmers often draw when working with new code, diagraming how the code's functions to clarify how a program can best be structured (Lister et al., 2004). That is, programmers use drawing as a way to abstract the information in a way that lets them see the problem more clearly.

Drawing makes student thinking explicit and encourages students to clarify their understanding by examining the clarity, coherence, and content of their drawings (Schwartz, 1995; Abrahamson & Trninic, 2015). This specificity in understanding that drawing provides diagnostic and formative assessment to teachers and facilitators, exposing what students do and do not understand (Ehrlén, 2009). In other words, drawing could possibly serve as a key tool in developing CT skills in a making environment by facilitating students' problem-solving process, connecting learning with real-world practices, and serving as a source of assessment (Hadad, Kachovska, Thomas, & Yin, 2019). By embedding drawing, tinkering would be less aimless and haphazard; it is an engaging form of "minds-on" practice that dovetails with the "hands-on" nature of making.

Debugging

Debugging, like walkthroughs and drawing, is a powerful practice to integrate into a classroom or makerspace culture in that it can serve as a form of metacognitive self-monitoring. Debugging is less about giving the right or wrong answer, and instead focuses on finding where the error is, why it is an error, and fixing it. This approach is useful not only for developing complex problem-solving skills, but also developing a sense of agency in being able to correct one's own errors (National Research Council, 2004). Finding where the error relies on the CT skill of decomposition, in which the student breaks apart the different parts of the problem and examines each part individually and systematically to determine what could be causing the problem, whether that be a light bulb that keeps shorting out or a piece of code that is never triggered. Determining why it is an error uses the CT skill of pattern recognition, comparing the circuit or code to previous artifacts and how they functioned and where theirs might have gone wrong. Being able to fix the code centers on students having the previously mentioned agency to persist in working with difficult problems and having a tolerance for ambiguity.

Modelling and coaching debugging provide students with not only a structure for this practice, but also a kind of permission for debugging. The facilitator presents error-making as part of the process, and not something that needs to be avoided at all costs. Developing a culture of debugging then makes the tinkering process more methodical and uses errors as an opportunity for learning, instead of an annoyance (Bers, 2019).

Computational Thinking Vs Computational Doing Vs Computational Design

What these findings also suggest is that there is no clear-cut delineation between computational thinking, doing, or designing. Tinkering with variables in a piece of code (or doing) is also part of understanding algorithms (or thinking) which is also part of iterating on an artifact (or designing). Computational thinking is computational doing and computational design. The lines between them are blurry and they work in tandem with one another. Collins, Brown, and Newman's (1989) focus on learning skills and content in authentic contexts is critical for students to develop a deep understanding of process, because thinking, doing, and designing are so interdependent.

Limitations

While the case provided a lot of rich and interesting data, the data collection process was not without its problems. Three of the eight recordings of participant screens were corrupted. The audio from some of the tabletop audio recorders and the video recordings could be hard to hear in several places because there would sometimes be a lot of simultaneous conversation or noise from the projects. Two hours from one of the recorders attached to a lavalier mic on one of the facilitators was lost. Also, the video mostly captured a profile view of the participant and facilitator work, making it difficult to see detail of what the participants were doing on their breadboards. An overhead view of the breadboards would have been helpful in seeing the work the participants were doing.

The missing audio from the facilitator's lavalier mic proved to be fairly inconsequential because most of what he said in those two hours was pieced together from the other audio and video sources. Not being able to see or hear some of the students' actions or thoughts, whether through the screen recording, the video camera placement, or the quality of the audio, did cause challenges for fully capturing what happened. However, the patterns that appeared were consistent enough that it is quite likely the additional data would have fallen along the same lines.

In order to produce a rich and thick description for a case study, I focused on one day in a summer program. This research enabled deep investigation to identify promising practices to promote CT as used by the facilitators in the ACTMA project, but it should be seen as exploratory. More rigor in similar studies would be necessary to develop a knowledge base that leads us to practicable solutions in both formal and informal learning environments. Additional or comparative case studies reporting on other makerspaces would help identify more patterns of effective practices to engender CT skills in order to merit generalizable recommendations. In addition, research conducted throughout the duration of a program would be useful, in order to see participant's learning transformations over a larger period of time.

Future Research

These limitations provide fodder for avenues of research from which to proceed. The following are particular areas of research that indicate further research based on this study's findings.

This study was a single case study, therefore its findings are intended as being descriptive. For these findings to be generalizable, other studies on making programs would need to be conducted (Stake, 1996). Multiple case studies would identify a more definitive set of best practices for CT skills in a maker environment, providing guidance to classroom and out-of-school-time instructors.

This study took place over one day in a nine-day program. Being able to conduct the same in-depth analysis over the entire nine days would provide a more complete picture of facilitator practices. Analyzing the entire nine days would also allow for combining this qualitative work with the results from the pre- and post-assessments of CT skills and dispositions, as well as with their demographic information, which were conducted at the beginning and end of the program. In addition, a study that would observe the impact of tinkering, embodiment, walkthroughs, drawing, and debugging over the course of an entire program could begin to examine the progression of these practices and how they can be scaffolded for optimal learning.

A more pointed study that intentionally integrates some of the described practices a priori instead of ex post facto post hoc would also provide clearer directions for future research and classroom practice. A study with an intervention that focuses on tinkering, embodiment, walkthroughs, drawing, and debugging in conjunction with cognitive apprenticeship practices may be useful in developing material for practitioners to utilize these practices in the classroom or after-school space.

There was little attention paid to the incorporation of physics and engineering content into the CT learning, and whether practices like tinkering, embodiment, walkthroughs, drawing, and debugging had an impact on the domain knowledge, in addition to CT skills and dispositions. A study that focused on this overlap could provide more guidance to subject area classroom instructors in incorporating CT as well as making practices in their classroom.

Although the activities of the day were structured to explore different tools and skills participants could use for their final project, which was meant to align with their interests, a future study could be more intentional on measuring the relationship between student interest, knowledge, and strategic processing of CT.

This study provided no examination of how these practices work with students from different demographics (e.g. race/ethnicity, gender, disability, sexual orientation). Analyzing how these practices are received and work with students who bring their own cultural experience to the learning space would provide a more nuanced understanding of how makerspaces can best be structured as a welcoming environment for learning.

The activities that focused on embodiment in this study were strongly linked with the practice of walkthroughs and algorithm design. Further research could explore whether embodiment activities also develop participants' other CT skills like decomposition or pattern recognition, and whether dosage and level of embodiment (direct vs surrogate) have an impact on learning outcomes.

The drawings that were generated from this study mostly focused on the circuit designs. There were few doodles connected with the Arduino programming. This may have been due to lack of time or that the drawing for programming was not as extensively modeled as the drawing for circuit design. A study examining whether more time allotted for drawing, as well as some facilitator modelling of drawing while programming, would encourage students to build upon the drawing practice they developed as well as whether encouraging this practice would assist in their CT skills as they developed their code.

The debugging practices in this study were noticed in isolation, that is, it was observed that debugging occurred in the breadboard exploration activity and the Arduino activity, but there is no in depth understanding of whether any of the debugging practices or dispositions transferred from one activity to another. An exploration of this would be useful in structuring the way maker activities are ordered and scaffolded.

Finally, theoretical coding on a larger data set may further frame and clarify the relationships between the practices and the methods described here, progressing the typology in Figure 19 into an inductive theory.

Conclusion

The development of CT skills and dispositions establish the ability to use computers effectively used to solve problems. In a world that is becoming increasingly shaped by computers -- in our commerce, democracy, relationships, and how we express ourselves – these skills and dispositions are incredibly powerful. Yet, understanding how and when to use elements of a thinking process is challenging. In addition, a focus on CT is often relegated to programming, removed from other subject areas where it can be most useful.

Novel materials, tools, and techniques, as well as celebrating individual interests and connecting them to community, are central to the maker movement's ethos. These characteristics allow for making to serve as an activity that can demonstrate CT's versatility and broad usefulness. However, the interest-driven open-ended nature of its appeal also makes it difficult to structure learning opportunities.

Practices like tinkering, embodiment, walkthroughs, drawing, and debugging, can provide instructors with this structure. They are practices that connect students with the processes experts use and still fit into the messy, explorative, and collaborative nature of makerspaces. By intentionally employing these practices alongside a framework like cognitive apprenticeship, instructors can make the implicit practices of CT explicit, and be able to guide students in making CT useful for themselves and for their communities.

REFERENCES

Abrahamson, D., & Trninic, D. (2015). Bringing forth mathematical concepts: Signifying sensorimotor enactment in fields of promoted action. ZDM Mathematics Education, 47(2), 295–306.

Abramsky, S. & Coecke, B. (2006). Physics from Computer Science — a position statement. Retrieved from http://www.cs.ox.ac.uk/people/bob.coecke/YORKIJUC.pdf on January 29, 2016.

Aikenhead, G. S., & Jegede, O. J. (March 01, 1999). Cross-Cultural Education: A Cognitive Explanation of a Cultural Phenomenon. *Journal of Research in Science Teaching, 36, 3*, 269-87.

Ainsworth, S., Prain, V., & Tytler, R. (2011). Drawing to learn in science. Science, *333*(6046), 1096-1097.

Allen, N., Templon, J., McNally, P., Birnbaum, L., & Hammond, K. (2010). Statsmonkey: Automatic generation of narrative from data. In *Computational Models of Narrative: Papers from the 2010 AAAI Fall Symposium.*

Alpert, J. (2016, January). Adding computational thinking to your science lesson: what should it look like? Retrieved April 06, 2017, from https://www.semanticscholar.org/paper/Adding-computational-thinking-to-your-science-less-Albert/b7bf0ab87f6df1d7fa170749ec000162314b06ad

Anderson, C. (2012). Makers: The new industrial revolution. New York: Crown

Anderson, J. R., Reder, L.M., & Simon, H.A. (2000, Summer). *Applications and Misapplications of Cognitive Psychology to Mathematics Education.* Texas Educational Review.

Anderson, M. D. (2016, February 29). How the Coding Craze Could Lead to 'Technical Ghettos'. Retrieved from https://www.theatlantic.com/education/archive/2016/02/will-the-push-for-coding-lead-to-technical-ghettos/471300/

Barba, L. (2016, March 5). Computational Thinking: I do not think it means what you think it means. Retrieved April 07, 2016, from http://lorenabarba.com/blog/computational-thinking-i-do-not-think-it-means-what-you-think-it-means/

Barr, V., & Stephenson, C. (2011). Bringing computational thinking to K-12: What is involved and what is the role of the computer science education community? ACM. doi:10.1145/1929887.1929905

Barsalou, L. W. (2010). Grounded cognition: Past, present, and future. Topics in cognitive science, 2(4), 716-724.

Basawapatna, A. R., Koh, K. H., & Repenning, A. (2010). Using scalable game design to teach computer science from middle school to graduate school. *Proceedings of the fifteenth annual conference on Innovation and technology in computer science education - ITiCSE 10*. doi:10.1145/1822090.1822154

Basawapatna, A., Koh, K. H., Repenning, A., Webb, D. C., & Marshall, K. S. (2011). Recognizing computational thinking patterns. In *Proceedings of the 42nd ACM Technical Symposium on Computer Science Education* (pp. 245–250). New York, NY: ACM.

Bates, R. (January 01, 2002). The impact of educational research: alternative methodologies and conclusions. *Research Papers in Education, 17*, 4, 403-408.

Beckwith, L., Kissinger, C., Burnett, M., Wiedenbeck, S., Lawrance, J., Blackwell, A., and Cook, C. 2006. Tinkering and Gender in End-user Programmers' Debugging. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. ACM. New York, NY, 231–240.

Bienkowski, M., Snow, E., Rutstein, D., Grover, S. (2015). Assessment Design Patterns for Computational Thinking Practices in Secondary Computer Science: A First Look. Retrieved September 23, 2016, from http://pact.sri.com/downloads/Assessment-Design-Patterns-for-Computational%20Thinking-Practices-Secondary-Computer-Science.pdf

Bell, J. (2015, May 12). School Librarians Push for More 'Maker Spaces'. *Education Week*. Retrieved December 3, 2015 from http://www.edweek.org/ew/articles/2015/05/13/school-librarians-push-for-more-maker-spaces.html.

Bell, T., Alexander, J., Freeman, I., & Grimley, M. (2009). Computer science unplugged: School students doing real computing without computers. New Zealand Journal of Applied Computing and Information Technology, 13(1), 20-29.

Berland, M. (2016). Making, tinkering, and computational literacy. Makeology: Makers as learners, 2, 196-205.

Berry, M., & Csizmadia, A. (2016, November 26). Computational thinking and mathematical reasoning – An open mind. Retrieved April 30, 2018, from http://milesberry.net/2016/11/computational-thinking-and-mathematical-reasoning/

Blikstein, P. (2013). Digital fabrication and 'making' in education: The democratization of invention. In J. Walter-Herrmann & C. Büching (Eds.), FabLabs: Of Machines, Makers and Inventors. Bielefeld, Germany: Transcript Publishers, pp. 1-21.

Block, C. C., & Israel, S. E. (2004). The ABCs of performing highly effective think-alouds. The Reading Teacher, 58(2), 154-167.

Blum, B. I. (1992). Software engineering: a holistic view. Oxford University Press, Inc.

Bobb, K., & Brown, Q. (2017). Access, Power, and the Framework of a CS Education Ecosystem. In *Moving Students of Color from Consumers to Producers of Technology* (pp. 245-260). IGI Global.

Borkowski, J. G. (April 01, 1992). Metacognitive Theory: A Framework for Teaching Literacy, Writing, and Math Skills. *Journal of Learning Disabilities, 25,* 4, 253-57.

Brady, C., Orton, K., Weintrop, D., Anton, G., Rodriguez, S., & Wilensky, U. (February 01, 2017). All Roads Lead to Computing: Making, Participatory Simulations, and Social Computing as Pathways to Computer Science. *IEEE Transactions on Education, 60,* 1.

Brahms, L., Wardrip, P., & Children's Museum of Pittsburgh. (2014, December). The learning practices of making: An evolving framework for design. Retrieved December 6, 2015 from http:// http://makeshoppgh.com/wp-content/uploads/2015/02/MAKESHOP-Learning-Practices-formatted_FINAL_Feb-2015.pdf

Bransford, J. D., Brown, A. L., & Cocking, R. R. (Eds.) (1999). *How People Learn: Brain, Mind, Experience, and School.* Washington, DC: National Academy Press.

Branwyn, G. (2015, April 16). Looking Back at the Launch of the Maker Movement: The First Maker Faire | Make:. Retrieved October 01, 2016, from http://makezine.com/2015/04/16/looking-back-launch-maker-movement-first-maker-faire

Brennan, K. & Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. Retrieved August 12, 2016, from http://web.media.mit.edu/~kbrennan/files/Brennan_Resnick_AERA2012_CT.pdf

Brenner, M.E., Mayer, R.E., Moseley, B., Brar, T., Duran, R., Reed, B.S., Webb, D. (1997). Learning by understanding: The role of multiple representations in learning algebra. American Educational Research Journal, 34(4), 663-689. doi:10.2307/1163353

Brown, J., Collins, A., Duguid, S. (1989). "Situated cognition and the culture of learning". Educational Researcher. 18 (1): 32–42. doi:10.3102/0013189x018001032.

Brown, J. S., & VanLehn, K. (1980). Repair theory: A generative theory of bugs in procedural skills. *Cognitive science*, *4*(4), 379-426.

Buchholz, B., Shively, K., Peppler, K., & Wohlwend, K. (2014). Hands on, hands off: Gendered access in crafting and electronics practices. Mind, Culture, and Activity, 21(4), 278-297.

Buitrago Flórez, F., Casallas, R., Hernández, M., Reyes, A., Restrepo, S., & Danies, G. (August 01, 2017). Changing a Generation's Way of Thinking: Teaching Computational Thinking Through Programming. *Review of Educational Research*, *87*, 4.

Burge, J., Gannod, G., Doyle, M., & Davis, K. (2013). Girls on the go: a CS summer camp to attract and inspire female high school students. *SIGCSE '13 Proceeding of the 44th ACM technical symposium on computer science education*, 615-620. Retrieved August 1, 2017.

Calabrese Barton, A., Tan, E., & Greenberg, D. (2016). The makerspace movement: Sites of possibilities for equitable opportunities to engage underrepresented youth in STEM. Teachers College Record.

Cavalcanti, G. (2013, May 22). Is it a Hackerspace, Makerspace, TechShop, or FabLab? Retrieved December 4, 2015, from http://makezine.com/2013/05/22/the-difference-between-hackerspaces-makerspaces-techshops-and-fablabs/

Chávez, V., & Soep, E. (2005). Youth radio and the pedagogy of collegiality. *Harvard Educational Review*, *75*(4), 409-434.

Chu, S. L., Quek, F., Bhangaonkar, S., Ging, A. B., & Sridharamurthy, K. (2015). Making the Maker: A Means-to-an-Ends approach to nurturing the Maker mindset in elementary-aged children. *International Journal of Child-Computer Interaction*, *5*, 11-19.

Clark, R. C., Nguyen, F., Sweller, J., & Baddeley, M. (2006). Efficiency in learning: Evidence-based guidelines to manage cognitive load. *Performance Improvement*, *45*(9), 46-47.

Code.org. (2020). Code.org Infographic Source Data. Retrieved from https://docs.google.com/document/d/1gySkItxiJn_vwb8HIIKNXqen184mRtzDX12cux0ZgZk/pub#h.haz08q3vji8e

Code.org, CSTA, & ECEP. (2019). 2019 State of Computer Science Education Equity and Diversity. Retrieved from https://advocacy.code.org/2019_state_of_cs.pdf

The College Board. (2014). The 10th Annual AP Report to the Nation. Retrieved September 21, 2016, from http://media.collegeboard.com/digitalServices/pdf/ap/rtn/10th-annual/10th-annual-ap-report-to-the-nation-single-page.pdf

The College Board. (2016). AP Computer Science Principles. Retrieved September 21, 2016, from https://secure-media.collegeboard.org/digitalServices/pdf/ap/ap-computer-science-principles-course-and-exam-description.pdf

Collins, A., Brown, J. S., & Newman, S. E. (1989). Cognitive apprenticeship: Teaching the crafts of reading, writing, and mathematics. Knowing, Learning, and Instruction: Essays in Honor of Robert Glaser, 18, 32-42.

Collins, A., & Ferguson, W., (1993) Epistemic forms and epistemic games: Structures and strategies to guide inquiry. Educational Psychologist, 28, 25-42. doi:10.1207/s15326985ep2801_3

Collins, A., & Kapur, M. (2014). Cognitive apprenticeship. In R. Keith Sawyer, ed., The Cambridge Handbook of the Learning Sciences, 2nd edn., Cambridge University Press. pp. 109-27.

Committee on Highly Successful Schools or Programs for K-12 STEM Education. (2011). *Successful STEM Education: A Workshop Summary*. National Academics Press.

Creswell, J. W., & Miller, D. L. (2000). Determining validity in qualitative inquiry. *Theory into practice*, *39*(3), 124-130.

Csizmadia, A., Curzon, P., Dorling, M., Humphreys, S., Ng, T., Selby, C., Woollard, J. (2015). Computational thinking: a guide for teachers. Retrieved October 1, 2016, from https://community.computingatschool.org.uk/resources/2324.

Cuny, J., Snyder, L., & Wing, J.M. (2010). Demystifying computational thinking for non-computer scientists. Unpublished manuscript in progress, referenced in http://www.cs.cmu.edu/~CompThink/resources/TheLinkWing.pdf

Cuoco, A., Goldenberg, E. P., & Mark, J. (1996). Habits of mind: An organizing principle for mathematics curricula. The Journal of Mathematical Behavior, 15(4), 375-402.

Davey, B. (1983). Think aloud: Modeling the cognitive processes of reading comprehension. Journal of reading, 27(1), 44-47.

de Koning, B. B., & Tabbers, H. K. (2011). Facilitating understanding of movements in dynamic visualizations: An embodied perspective. Educational Psychology Review, 23(4), 501-521.

Deci, E. L., & Ryan, R. M. (1991). A motivational approach to self: Integration in personality.
In R. Dienstbier (Ed.), Nebraska Symposium on motivation: Perspectives on motivation
(Vol. 38, pp. 237-288). Lincoln, NE: University of Nebraska Press.

Deci, E. L., Valleran, R. J., Pelletier, L. G., & Ryan, R. M. (1991). Motivation and education:
The self-determination perspective. Educational Psychologist, 26, 325-346.

Denning, P. J. (2017, January). Computational Thinking in Science. American Scientist, 105(1), 13. doi:10.1511/2017.124.13

Denner, J., Werner, L., & Ortiz, E. (2012). Computer games created by middle school girls: Can they be used to measure understanding of computer science concepts? *Computers & Education*, 58(1), 240–249.

Dewey, J. (1913). *Interest and effort in education.* Boston: Houghton Mifflin Company.

Dewey, J. (1916). Democracy and education: An introduction to the philosophy of education. New York: Macmillan.

DiSessa, A. A. (2000). Changing minds: Computers, learning, and literacy. Cambridge, Mass: MIT Press.

Dijkstra, E.W. (1974). Programming as a discipline of mathematical nature. *American Mathematical Monthly*, *81*(6), 608-612.

Drew, D. E. (2011). STEM the tide: Reforming science, technology, engineering, and math education in America. Baltimore: Johns Hopkins University Press.

Duschl, R. A. (2003). Assessment of inquiry. In J. M. Atkin & J. E. Coffey (Eds.), *Everyday assessment in the science classroom* (pp. 41–59). Arlington, VA: National Science Teachers Association Press.

Dweck, C.S., Walton, G.M., and Cohen, G.L. (2011). *Academic tenacity: Mindsets and skills that promote long-term learning.* White paper prepared for the Gates Foundation. Seattle, WA.

Educause. (2013, April 9). 7 things you should know about makerspaces. Retrieved from https://net.educause.edu/ir/library/pdf/eli7095.pdf

Ehrlén, K. (2009). Drawings as representations of children's conceptions. International Journal of Science Education, 31(1), 41-57.

Elliott, E. S., & Dweck, C. S. (January 01, 1988). Goals: an approach to motivation and achievement. Journal of Personality and Social Psychology, 54, 1, 5-12.

The Emerging Science of Computational Anthropology. (2014, June 10). Retrieved January 17, 2016, from http://www.technologyreview.com/view/528216/the-emerging-science-of-computational-anthropology/.

Educause (2013, April 9). 7 things you should know about makerspaces. Retrieved from https://net.educause.edu/ir/library/pdf/eli7095.pdf

Eglash, R., Gilbert, J. E., Taylor, V., & Geier, S. R. (September 01, 2013). Culturally Responsive Computing in Urban, After-School Contexts: Two Approaches. *Urban Education*, *48*, *5*, 629-656.

Ericson, B. (2014). Detailed AP CS 2013 Results: Unfortunately, much the same. Retrieved January 17, 2016, from https://computinged.wordpress.com/2014/01/01/detailed-ap-cs-2013-results-unfortunately-much-the-same/

Fadjo, C. L. (2012). Developing computational thinking through grounded embodied cognition (Doctoral dissertation, Teachers College).

Farrington, C. A., Roderick, M., Allensworth, E. A., Nagaoka, J., Johnson, D. W., Keyes, T. S., & Beechum, N. (2012). Teaching Adolescents to Become Learners: The Role of Noncognitive Factors in Academic Performance – A Critical Literature Review. Chicago: Consortium on Chicago School Research.

Farris, A. V., & Sengupta, P. (2014). Perspectival Computational Thinking for Learning Physics: A Case Study of Collaborative Agent-based Modeling. *Proceedings of the 11th International Conference of the Learning Sciences (ICLS 2014)*. Retrieved February 2, 2017, from https://arxiv.org/pdf/1403.3790.pdf

Fields, D. A. & King, W. L. (2014). "So, I think I'm a programmer now." Developing connected learning for adults in a university craft technologies course. In Polman, J. L. et al. (Eds.). (2014). *Learning and Becoming in Practice: The International Conference of the Learning Sciences (ICLS) 2014*, *Volume 1*. Boulder, CO: International Society of the Learning Sciences, pp. 927-936.

Fields, D. A., Searle, K. A., & Kafai, Y. B. (2016, October). Deconstruction kits for learning: Students' collaborative debugging of electronic textile designs. In *Proceedings of the 6th Annual Conference on Creativity and Fabrication in Education* (pp. 82-85). ACM.

Fields, D., Vasudevan, V., & Kafai, Y. B. (2015). The programmers' collective: fostering participatory culture by making music videos in a high school Scratch coding workshop. Interactive Learning Environments, 23(5), 613-633.

Fitzgerald, S., Simon, B., & Thomas, L. (2005, October). Strategies that students use to trace code: an analysis based in grounded theory. In *Proceedings of the first international workshop on Computing education research* (pp. 69-80). ACM.

Flores, C. (2016, April 24). Constructionism, a Learning Theory and a Model for Maker Education. Retrieved May 31, 2017, from http://fellows.fablearn.org/constructionism-a-learning-theory-and-a-model-for-maker-education/

Franklin, D., Conrad, P., Boe, B., Nilsen, K., Hill, C., Len, M., Dreschler, G., Aldana, G., Almeida-Tanaka, P., Kiefer, B., Laird, C., Lopez, F., Pham, C., Suarez, J., & Waite, R. (2013). Assessment of computer science learning in a scratch-based outreach program. *SIGCSE '13 Proceeding of the 44th ACM technical symposium on computer science education, 371-376*. Retrieved August 7, 2017.

Freire, P. (1974). *Pedagogy of the oppressed.* New York,: Seabury Press.

Furtak, E., Seidel, T., Iverson, H., and Briggs, D.C. (2012). Experimental and quasi-experimental studies of inquiry-based science teaching: A meta-analysis. *Review of Educational Research*, *82*(3), 300-329.

Gallup & Google. (2016, October 20). More K-12 Computer Science Classes Teach Programming/Coding. Retrieved from http://news.gallup.com/poll/196511/computer-science-classes-teach-programming-coding.aspx

Giannakos, M. N., & Jaccheri, L. (2013, June). What motivates children to become creators of digital enriched artifacts?. In *Proceedings of the 9th ACM Conference on Creativity & Cognition* (pp. 104-113). ACM.

Giannakos, M. N., Jaccheri, L., & Leftheriotis, I. (2014, June). Happy girls engaging with technology: assessing emotions and engagement related to programming activities. In International Conference on Learning and Collaboration Technologies (pp. 398-409). Springer, Cham.

Gilbert, J. K. (2005). Visualization: A metacognitive skill in science and science education. In Visualization in science education (pp. 9-27). Springer, Dordrecht.

Gonzales, P., Williams, T., Jocelyn, L., Roey, S., Kastberg, D., and Brenwald, S. (2008). Highlights from TIMSS 2007: Mathematics and science achievement of US fourth and eighth-grade students in an international context.(NCES 2009-001 Revised). Washington, DC: National Center for Education Statistics, Institute of Education Sciences, U.S. Department of Education.

Good, T. (2013, January 28). What is "Making"?. Retrieved from https://makezine.com/2013/01/28/what-is-making/

Goode, J., Margolis, J., & Chapman, G. (2014). Curriculum is not enough: The educational theory and research foundation of the Exploring Computer Science professional development model. In *Proceedings of the 45th ACM Technical Symposium on Computer Science Education*, 493-498. doi: 10.1145/2538862.2538948. Read the article.

Google. (n.d.). Computational Thinking for Educators. Retrieved from https://computationalthinkingcourse.withgoogle.com/ on January 2, 2015.

Google (2015). *Searching for computer science: Access and barriers in U.S. K-12 education.* Retrieved from https://services.google.com/fh/files/misc/searching-for-computer-science_report.pdf

Grant, C., & Osanloo, A. (January 01, 2014). Understanding, Selecting, and Integrating a Theoretical Framework in Dissertation Research: Creating the Blueprint for Your "House". *Administrative Issues Journal: Education, Practice, and Research, 4,* 2, 12-26.

Greeno, J. G. (1991). Number sense as situated knowing in a conceptual domain. *Journal for research in mathematics education*, 170-218.

Greeno, J. G., & van de Sande, C. (2007). Perspectival understanding of conceptions and conceptual growth in interaction. *Educational Psychologist, 42*(1), 9-23.

Grover, S. (2018, November 5). A Tale of Two CTs (and a Revised Timeline for Computational Thinking). Retrieved from https://cacm.acm.org/blogs/blog-cacm/232488-a-tale-of-two-cts-and-a-revised-timeline-for-computational-thinking/fulltext

Grover, S., & Pea, R. (January 01, 2013). Computational Thinking in K-12: A Review of the State of the Field. *Educational Researcher, 42,* 1, 38-43.

Grover, S., Pea, R., & Cooper, S. (June 01, 2015). Designing for deeper learning in a blended computer science course for middle school students. *Computer Science Education, 25,* 2.

Guzdial, M. (January 01, 1994). Software-Realized Scaffolding to Facilitate Programming for Science Learning. *Interactive Learning Environments.* 4, 1, 1-44.

Guzdial, M. (2017, January 02). Balancing cognition and motivation in computing education: Herbert Simon and evidence-based education. Retrieved January 12, 2017, from https://computinged.wordpress.com/2017/01/06/balancing-cognition-and-motivation-in-computing-education-herbert-simon-and-evidence-based-education/

Hadad, R., Thomas, K., Kachovska, M., & Yin, Y. (2019). Practicing Formative Assessment for Computational Thinking in Making Environments. Journal of Science Education and Technology, 1-12.

Halverson, E. R., & Sheridan, K. M. (January 01, 2014). The Maker Movement in Education. *Harvard Educational Review, 84,* 4, 495-504.

Hambrusch, S., Hoffmann, C., Korb, J. T., Haugan, M., & Hosking, A. L. (March 04, 2009). A multidisciplinary approach towards computational thinking for science majors. *ACM SIGCSE Bulletin*, *41*, 1, 183.

Harel, I. (2018, April 8). American schools are teaching our kids how to code all wrong. Retrieved from https://qz.com/691614/american-schools-are-teaching-our-kids-how-to-code-all-wrong/

Harel, I., & Papert, S. (1991). *Constructionism.* Norwood, NJ: Ablex.

Hou, W., Kaur, M., Komlodi, A., Lutters, W. G., Boot, L., Cotton, S. R., Morrell, C., Ozok, A. A., and Tufekci, Z. (2006). 'Girls Don't Waste Time': Pre-adolescent Attitudes Toward ICT," in CHI '06 Extended Abstracts on Human Factors in Computing Systems. ACM. New York, NY, 875–880.

Heather. (2016, March 07). Why did Maker Faire start? And, what is the Maker Movement? Retrieved October 02, 2016, from https://help.makermedia.com/hc/en-us/articles/204141849-Why-did-Maker-Faire-start-And-what-is-the-Maker-Movement-

Herold, B. (2018, March 01). The 'Maker' Movement Is Coming to K-12: Can Schools Get It Right? Retrieved from https://www.edweek.org/ew/articles/2016/06/09/the-maker-movement-is-coming-to-k-12.html

Hertz, M. B. (2012, November 06). Creating Makerspaces in Schools. Retrieved October 06, 2016, from http://www.edutopia.org/blog/creating-makerspaces-in-schools-mary-beth-hertz

Hmelo-Silver, C. E., Duncan, R. G., & Chinn, C. A. (2007). Scaffolding and achievement in problem-based and inquiry learning: a response to Kirschner, Sweller, and. *Educational psychologist*, *42*(2), 99-107.

Hogan, A., & Roberts, B. (2015, December). Occupational employment projections to 2024 : Monthly Labor Review. Retrieved June 05, 2017, from https://www.bls.gov/opub/mlr/2015/article/occupational-employment-projections-to-2024.htm

Honey, M., & Hilton, M. L., (Eds.). (2011). National Academies Press (U.S.), & National Research Council (U.S.). Committee on Science Learning: Computer Games, Simulations, and Education. *Learning science through computer games and simulations.* Washington, D.C: National Academies Press.

Honey, M., & Kanter, D. (2013). Design, make, play: Growing the next generation of STEM innovators. New York: Routledge.

Howard, T., & Terry, C. L. S. (December 01, 2011). Culturally responsive pedagogy for African American students: promising programs and practices for enhanced academic performance. *Teaching Education*, *22*, 4, 345-362.

International Society for Technology in Education & Computer Science Teachers Association. (2011). Computational Thinking Teacher Resources. Retrieved from http://www.iste.org/docs/ct-documents/ct-teacher-resources_2ed-pdf.pdf?sfvrsn=2 on March 1, 2015.

International Society for Technology in Education & Computer Science Teachers Association. (2011). Operational Definition of Computational Thinking for K-12 Education. Retrieved from http://csta.acm.org/Curriculum/sub/CurrFiles/CompThinkingFlyer.pdf on April 1, 2014.

Inhelder, B., & Piaget, J. (1958). The growth of logical thinking from childhood to adolescence: An essay on the construction of formal operational structures. New York: Basic Books.

Jenkins, H. (2006). Convergence culture: Where old and new media collide. Cambridge, MA: MIT Press.

Johnson, A. M., Butcher, K. R., Ozogul, G., & Reisslein, M. (2014). Introductory circuit analysis learning from abstract and contextualized circuit representations: Effects of diagram labels. IEEE Transactions on Education, 57(3), 160-168.

K–12 Computer Science Framework. (2016). Retrieved November 6, 2017, from http://www.k12cs.org

Kafai, Y. B. (2006). Playing and making games for learning: Instructionist and constructionist perspectives for game studies. Games and culture, 1(1), 36-40.

Kafai, Y. B., & Burke, Q. (2015). Constructionist gaming: Understanding the benefits of making games for learning. Educational psychologist, 50(4), 313-334.

Kafai, Y. B., Desai, S., Peppler, K. A., Chiu, G. M., & Moya, J. (2008). Mentoring partnerships in a community technology centre: A constructionist approach for fostering equitable service learning. Mentoring & Tutoring: Partnership in Learning, 16(2), 191-205.

Kafai, Y. B., Lee, E., Searle, K., Fields, D., Kaplan, E., & Lui, D. (2014). A crafts-oriented approach to computing in high school: Introducing computational concepts, practices, and perspectives with electronic textiles. ACM Transactions on Computing Education (TOCE), 14(1), 1.

Kafai, Y. B., Peppler, K. A., & Chapman, R. N. (2009). The Computer Clubhouse: Constructionism and Creativity in Youth Communities. Technology, Education-

-Connections. Teachers College Press. 1234 Amsterdam Avenue, New York, NY 10027.

Kafai, Y., & Resnick, M. (Eds.). (1996). Constructionism in practice: Designing, thinking and learning in a digital world. Mahwah, NJ: Lawrence Erlbaum.

Kafai, Y. B., Searle, K., Lui, D., Lee, E., Fields, D., & Kaplan, E. (January 01, 2014). A crafts-oriented approach to computing in high school: Introducing computational concepts, practices, and perspectives with electronic textiles. *ACM Transactions on Computing Education*, *14*, 1.)

Karp, R. M. (2011). Understanding science through the computational lens. *算机科学技学：英文版*, *26*(4), 569-577. doi:10.1007/s11390-011-1157-0

Katterfeldt, E. S., Dittert, N., & Schelhowe, H. (2015). Designing digital fabrication learning environments for Bildung: Implications from ten years of physical computing workshops. *International Journal of Child-Computer Interaction*, *5*, 3-10.

Kazimoglu, C., Kiernan, M., Bacon, L., & Mackinnon, L. (2012). A serious game for developing computational thinking and learning introductory computer programming. Procedia-Social and Behavioral Sciences, 47, 1991-1999.

Kenworthy, L., & Kielstra, P. (2015). Driving the skills agenda: Preparing students for the future. The Economist Intelligence Unit Limited.

Khalili, N., Sheridan, K., Williams, A., Clark, K., & Stegman, M. (July 01, 2011). Students Designing Video Games about Immunology: Insights for Science Learning. Computers in the Schools, 28, 3, 228-240.

Knuth, D. E. (1974). Computer science and its relation to mathematics. *American Mathematical Monthly*, *81*, pp. 323-343.

Knuth, D. E. (1985). Algorithmic thinking and mathematical thinking. *American Mathematical Monthly*, *92*, pp. 170-181.

Kolko, B., Hope, A., Sattler, B., MacCorkle, K., & Sirjani, B. (2012, August). Hackademia: building functional rather than accredited engineers. In *Proceedings of the 12th Participatory Design Conference: Research Papers-Volume 1* (pp. 129-138). ACM.

Kolodner, J. L., Camp, P. J., Crismond, D., Fasse, B., Gray, J., Holbrook, J., Puntambekar, S., & Ryan, M (2003). Problem-based learning meets case-based reasoning in the middle-school science classroom: Putting learning by design into practice. *Journal of the Learning Sciences*, *12*(4), 495–547.

Krieger, S., Allen, M., & Rawn, C. (2015). Are females disinclined to tinker in computer science?. In Proceedings of the 46th ACM Technical Symposium on Computer Science Education (pp. 102-107). ACM.

Krapp, A. (January 01, 1999). Interest, motivation and learning: An educational-psychological perspective. *European Journal of Psychology of Education, 14*, 1, 23-40.

Kurti, R. S., Kurti, D., & Fleming, L. (December 01, 2014). Practical Implementation of an Educational Makerspace. *Teacher Librarian, 42*, 2.

Ladson-Billings, G. (1995). Toward a theory of culturally relevant pedagogy. American Educational Research Journal, 32(3), 465-491. Retrieved from http://aer.sagepub.com/content/32/3/465.full.pdf.

Lamers, M. H., Verbeek, F. J., & van der Putten, P. W. (2013, November). Tinkering in scientific education. In International Conference on Advances in Computer Entertainment Technology (pp. 568-571). Springer, Cham.

Lave, J. (1988). Cognition in Practice: Mind, mathematics, and culture in everyday life. Cambridge, UK: Cambridge University Press.

Lave, J., & Wenger, E. (1991). *Situated learning: Legitimate peripheral participation.* Cambridge: Cambridge University Press.

Lee, E., Kafai, Y. B., Vasudevan, V., & Davis, R. L. (2014). Playing in the arcade: Designing tangible interfaces with MaKey MaKey for Scratch games. In *Playful User Interfaces* (pp. 277-292). Springer Singapore.

Lee, I. (2011, April 9). Fostering Computational Thinking Through Modeling and Simulation. Paper presented at the 2011 annual meeting of the American Educational Research
Association. Retrieved March 9, 2017, from the AERA Online Paper Repository.

Lee, I., Martin, F., Denner, J., Coulter, B., Allan, W., Erickson, J., Malyn-Smith, J., & Werner, L. (2011). Computational thinking for youth in practice. *ACM Inroads.* 2(1), 32-37.

Lister, R., Adams, E. S., Fitzgerald, S., Fone, W., Hamer, J., Lindholm, M., McCartney, R., Moström, J.E., Sanders, K., Seppälä, O., & Simon, B. (2004, June). A multi-national study of reading and tracing skills in novice programmers. In ACM SIGCSE Bulletin (Vol. 36, No. 4, pp. 119-150). ACM.

Litts, B. (2015). Making learning: Makerspaces as learning environments Retrieved from http://www.informalscience.org/sites/default/files/Litts_2015_Dissertation_Published.pdf

Lu J, Fletcher G. (2009). Thinking about computational thinking. Proceedings of the 40th ACM technical symposium on computer science education. 260-264.

Malaluan, N. E. & Masangcay, D. B. (November 01, 2015). Physics Instruction Utilizing Culture-Based Pedagogy. *Asia Pacific Journal of Multidisciplinary Research, 3*, 50-58.

Malyn-Smith, J., & Lee, I. (2012). Application of the occupational analysis of computational thinking-enabled STEM professionals as a program assessment tool. Journal of Computational Science Education, 3(1), 2-10.

Mannila, L., Dagiene, V., Demo, B., Grgurina, N., Mirolo, C., Rolandsson, L., & Settle, A. (2014). Computational Thinking in K-9 Education Proceedings of the Working Group Reports of the 2014 on Innovation & Technology in Computer Science Education Conference (pp. 1-29). New York, USA: ACM.

Margolis, J., Holme, J., Estrella, R., Goode, J., & Nao, K. (2017). Stuck in the shallow end: Education, race, and computing, revised ed.

Margulieux, L., Dorn, B., & Searle, K. (2019). Learning Sciences for Computing Education.

Marin, L. M., & Halpern, D. F. (January 01, 2012). Pedagogy for developing critical thinking in adolescents: Explicit instruction produces greatest gains. Thinking Skills and Creativity, 6, 1, 1-13.

Martinez, S. & Stager, G. (2013). *Invent to learn: making, tinkering and engineering in the classroom.* Constructing Modern Knowledge Press: Torrence, CA.

Martinez, S., & Stager, G. (2013b, November 05). Making Matters! How the Maker Movement Is Transforming Education. Retrieved May 30, 2017, from https://www.weareteachers.com/making-matters-how-the-maker-movement-is-transforming-education/

Mayer, B. (2013, July 2). Creating Game-Based Makerspaces. Retrieved August 20, 2017, from https://americanlibrariesmagazine.org/blogs/the-scoop/creating-game-based-makerspaces/

McDermott, L. C., & Shaffer, P. S. (1992). Research as a guide for curriculum development: An example from introductory electricity. Part I: Investigation of student understanding. American journal of physics, 60(11), 994-1003.

McWilliams, J. (2010, May 6). Notes on being the chainsaw you wish to see in the world: Closing remarks for the AERA 2010 annual meeting. Retrieved June 07, 2016, from http://jennamcwilliams.blogspot.com/2010/05/notes-on-being-chainsaw-you-wish-to-see.html

Meece, J., Blumenfeld, P., & Hoyle, R. (1988). Students' goal orientations and cognitive engagement in classroom activities. *Journal of Educational Psychology, 80*, 514-523.

Mellis, D. A., Jacoby, S., Buechley, L., Perner-Wilson, H., & Qi, J. (2013, February). Microcontrollers as material: crafting circuits with paper, conductive ink, electronic components, and an untoolkit. In *Proceedings of the 7th International Conference on Tangible, Embedded and Embodied Interaction* (pp. 83-90). ACM.

Moll, L. C., Amanti, C., Neff, D., & Gonzalez, N. (1992). Funds of knowledge for teaching: Using a qualitative approach to connect homes and classrooms. Theory into practice, 31(2), 132-141.

Moreno, R., Ozogul, G., & Reisslein, M. (2011). Teaching with concrete and abstract visual representations: Effects on students' problem solving, problem representations, and learning perceptions. Journal of Educational Psychology, 103(1), 32.

Moreno, R., Reisslein, M., & Ozogul, G. (2009, October). Pre-college electrical engineering instruction: Do abstract or contextualized representations promote better learning?. In Frontiers in Education Conference, 2009. FIE'09. 39th IEEE (pp. 1-6). IEEE.

Moriwaki, K., Brucker-Cohen, J., Campbell, L., Saavedra, J., Stark, L., Taylor, L. (2012). Adapting an art and design workshop to support STEM to STEAM learning experiences. Integrated STEM Education Conference (ISEC), 2012. IEEE 2nd, IEEE, New York, 1–6.

Moursund, D. (n.d.). Computational Thinking. Retrieved March 07, 2016, from http://iae-pedia.org/Computational_Thinking

Museus, S. D., Palmer, R. T., Davis, R. J., Maramba, D. C., In Ward, K., & In Wolf-Wendel, L. (2011). *Racial and ethnic minority students' success in STEM education.* San Francisco, Calif: Jossey-Bass Inc.

National Governors Association Center for Best Practices, Council of Chief State School Officers. (2010). Common core state standards for mathematics. National Governors Association Center for Best Practices, Council of Chief State School Officers, Washington, DC

National Research Council. (2004). How students learn: History, mathematics, and science in the classroom. National Academies Press.

National Research Council (U.S.). (2010). *Report of a workshop on the scope and nature of computational thinking*. Washington, D.C: National Academies Press.

National Research Council (2011a) *Learning science through computer games and simulations.* The National Academies Press,Washington, DC

National Research Council. (2011b). *Report of a workshop of pedagogical aspects of computational thinking*. Washington, DC: The National Academies Press.

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

National Research Council. (2014). STEM integration in K-12 education: Status, prospects, and an agenda for research. National Academies Press.

National Science Board. (2016). Science and Engineering Indicators 2016. Arlington, VA: National Science Foundation.

Newson, J., & Newson, E. (1975). Intersubjectivity and the transmission of culture: On the social origins of symbolic functioning. *Intersubjectivity and the transmission of culture: On the social origins of symbolic functioning, 28*, 437-466.

Ng, C., & Bahr, N. (2004, December 19). Knowledge structures and motivation to learn: Reciprocal effects. Retrieved January 01, 2017, from http://www.iier.org.au/qjer/qjer16/ng.html

Next Generation Science Standards Lead States (NGSS). (2013) Next generation science standards: for states, by states. The National Academies Press, Washington, DC.

Organisation for Economic Co-operation and Development. (2000). Science, technology and innovation in the new economy. Paris, France: OECD. Retrieved from www.oecd.org/dataoecd/3/48/1918259.pdf

Organisation for Economic Co-operation and Development. (2006). PISA 2006: Science competencies for Tomorrow's world - OECD briefing note for the united states. Paris, France: OECD. Retrieved October 6, 2016 from www.oecd.org/dataoecd/16/28/39722597.pdf

Organisation for Economic Co-operation and Development. (2012). PISA 2012: What Students Know and Can do: Student Performance in Mathematics, Reading and Science (Volume I). Paris, France: OECD. Retrieved October 6, 2016 from www.oecd.org/dataoecd/16/28/39722597.pdf

Ornelas, N. O., Calderon, G., & Blikstein, P. (2014). Makers in Residence Mexico: Creating the Conditions for Invention. In Short paper at FabLearn Europe: Digital Fabrication in Education Conference.

Ostad, A., Sorenson, P. M. (2007). Private speech and strategy-use patterns: Bidirectional comparisons of students with and without mathematical disabilities in a developmental perspective. Journal of Learning Disabilities, 40, 2–14.

Papavlasopoulou, S., Giannakos, M. N., & Jaccheri, L. (January 01, 2017). Empirical studies on the Maker Movement, a promising approach to learning: A literature review. Entertainment Computing, 18, 1, 57-78.

Papert, S. (1996). An exploration in the space of mathematics educations. *International Journal of Computers for Mathematical Learning, 1*(1) doi:10.1007/BF00191473

Papert, S. (1980). Mindstorms: Children, computers, and powerful ideas. New York: Basic Books.

Parsons, S. (1995). Making sense of students' science: The construction of a model of tinkering. Research in Science Education, 25(2), 203-219.

Peppler, K. (2013). STEAM-powered computing education: using E-textiles to integrate the arts and STEM. *Computer, 46*(9), 38-43. doi:10.1109/MC.2013.257

Peppler, K., Danish, J. A., Zaitlen, B., Glosson, D., Jacobs, A., Phelps, D. (2010). BeeSim: leveraging wearable computers in participatory simulations with young children. Proceedings of the 9th International Conference on Interaction Design and Children, 246-249.

Peppler, K., Halverson, E. & Kafai, Y. (Eds.) (2016). *Makeology: Makerspaces as learning environments* (Volume 1). New York, NY: Routledge.

Perlis, A. (1962). The computer in the university. In M. Greenberger, Ed., Computers and the World of the Future, MIT Press, Cambridge, MA. 180–219.

Perner-Wilson, H., Buechley, L., & Satomi, M. (2011, January). Handcrafting textile interfaces from a kit-of-no-parts. In Proceedings of the fifth international conference on Tangible, embedded, and embodied interaction (pp. 61-68). ACM.

Phillips, M. M. (2007). *The influence of situational factors on the nurturance of personal interest and perceived competence* (Order No. 3298099). Available from Dissertations & Theses @ CIC Institutions; ProQuest Dissertations & Theses A&I; ProQuest Dissertations & Theses Global. (304850775). Retrieved from http://search.proquest.com/docview/304850775?accountid=14552

Petrich, M., Wilkinson, K., & Bevan, B. (2013). It looks like fun, but are they learning? In M. Honey & D. Kanter, (Eds.), Design, make, play: Growing the next generation of STEM innovators (pp. 50-70). New York: Routledge.

Piaget, J. (1951). *The psychology of intelligence.* London: Routledge and Kegan Paul.

Pittser, B. (2016, September 01). Why You Need Games for Your Makerspace. Retrieved August 20, 2017, from https://www.filamentgames.com/blog/why-you-need-games-your-makerspace

President's Information Technology Advisory Committee. (2005). Computational Science: Ensuring America's Competitiveness. Retrieved October 1, 2016, from https://www.nitrd.gov/pitac/reports/20050609_computational/computational.pdf.

Pressley, M., El-Dinary, P. B., Gaskins, I., Schuder, T., Bergman, J. L., Almasi, J., & Brown, R. (1992). Beyond direct explanation: Transactional instruction of reading comprehension strategies. The Elementary School Journal, 92(5), 513-555.

Pressley, M., Harris, K. R., & Marks, M. B. (1992). But good strategy instructors are constructivists!. *Educational Psychology Review, 4*(1), 3-31.

Puntambekar, S., & Kolodner, J. L. (2005). Toward implementing distributed scaffolding: Helping students learn science from design. *Journal of Research in Science Teaching, 42* (2), 185–217. doi:10.1002/tea.20048

Qui, K, Buechley, L., Baafi, E., & Dubow, W. (2013). A curriculum for teaching computer science through computational textiles. *IDC '13 Proceedings of the 12th International Conference on Interaction Design and Children*, 20-27. Retrieved August 10, 2017.

Ralph, P. & Wand, Y. (2009) A Proposal for a formal definition of the design concept. In Lyytinen, K., Loucopoulos, P., Mylopoulos, J., and Robinson, W., (Eds.): *Design Requirements Engineering: A Ten-Year Perspective (LNBIP 14): Design Requirements Workshop 2007 Revised and Invited Papers*, pp. 103-136. Springer-Verlag.

Rambusch, J., & Ziemke, T. (2005, July). The role of embodiment in situated learning. In Proceedings of the 27th Annual Conference of the Cognitive Science Society (pp. 1803-1808). Mahwah, NJ: Lawrence Erlbaum.

Redish E.F., & Wilson J.M. (1993). Student programming in the introductory physics course: mUPPET. *American Journal of Physics, 61*, p.222-32.

Rittle-Johnson, B., Siegler, S., & Alibali, M. (2001). Developing conceptual understanding and procedural skill in mathematics: An iterative process. *Journal of Educational Psychology, 93*, 346-362. doi:10.1037/0022-0663.93.2.346

Rode, J. A., Weibert, A., Marshall, A., Aal, K., von Rekowski, T., El Mimouni, H., & Booker, J. (2015, September). From computational thinking to computational making. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing* (pp. 239-250). ACM.

Schneider, B., Bumbacher, E., & Blikstein, P. (2015). Discovery versus direct instruction: Learning outcomes of two pedagogical models using tangible interfaces. In Exploring the material conditions of learning: opportunities and challenges for CSCL," the Proceedings of the Computer Supported Collaborative Learning (CSCL) Conference (Vol. 1, pp. 364-371).

Schoenfeld, A. H. (1985). *Mathematical problem solving*. San Diego, CA: Academic Press.

Schwartz, D. L. (1995). The emergence of abstract representations in dyad problem solving. The journal of the learning sciences, 4(3), 321-354.

Schwartz, D. L., & Black, J. B. (1996). Shuttling between depictive models and abstract rules: Induction and fallback. Cognitive Science, 20(4), 457–497.

Schwartz, D. L., & Martin, T. (2004). Inventing to prepare for future learning: The hidden efficiency of encouraging original student production in statistics instruction. Cognition and Instruction, 22(2), 129-184.

Scott, K., Aist, G., & Zhang, X. (2014). Designing a Culturally Responsive Computing Curriculum for Girls. *International Journal Of Gender, Science And Technology, 6(2)*, 264-276. Retrieved from http://genderandset.open.ac.uk/index.php/genderandset/article/view/361

Schwartz, L., DiGiacomo, D., & Gutierrez, K. (2013). Diving into practice with children and undergraduates: a cultural historical approach to instantiating making and tinkering activity in a designed learning ecology. In *International Conference of the Learning Sciences (ICLS)*, Boulder CO.

Seehorn, D., Carey, S., Fuschetto, B., Lee, I., Moix, D., O'Grady-Cunniff, D., Owens, B. B., Stephenson, C., & Verno, A. (2011). CSTA K–12 computer science standards: Revised 2011. New York, NY: Computer Science Teachers Association (CSTA) and the Association for Computing Machinery, Inc (ACM)

Selby, C. and Woollard, J. (2014) Computational Thinking: The developing definition. In *Proceedings of the 45th ACM Technical Symposium on Computer Science Education, SIGCSE 2014*. ACM.

Sengupta, P., Farris, A. V., & Wright, M. (2012). From agents to continuous change via aesthetics: Learning mechanics with visual agent-based computational modeling. *Technology, Knowledge & Learning, 17*(1-2), 23 - 42.

Sengupta, P., Kinnebrew, J. S., Basu, S., Biswas, G., & Clark, D. (2013). Integrating computational thinking with K-12 science education using agent-based computation: A theoretical framework. *Education and Information Technologies, 18*(2), 351-380. doi:10.1007/s10639-012-9240-x

Sheil, B. A. (1983). Coping with complexity. *Office Technology and People, 1*(4), 295-320.

Sheridan, K. M., Halverson, E.R., Litts, B., Brahms, L., Jacobs-Priebe, L., & Owens, T. (2014). Learning in the making: A comparative case study of three makerspaces. *Harvard Educational Review, 84*(4),505-531.

Sherin, B. L. (January 01, 2001). A Comparison of Programming Languages and Algebraic Notation as Expressive Languages for Physics. *International Journal of Computers for Mathematical Learning, 6*, 1, 1-61.

Silvia, P.J. (2005). What is interesting? Exploring the appraisal structure of interest. *Emotion, 5*, 89–102.

Snow, E. (2014, January). Assessing Computational Thinking. Retrieved February 14, 2014, from https://pact.sri.com/downloads/AssessmentBreakout.pdf

Stager, G. (2005). Papertian Constructionism and the Design of Productive Contexts for Learning. Retrieved from http://stager.org/articles/eurologo2005.pdf on December 29, 2015.

Stevens, G., Boden, A., & von Rekowski, T. (2013, June). Objects-to-think-with-together. In *International Symposium on End User Development* (pp. 223-228). Springer, Berlin, Heidelberg.

Stephenson, C., & Malyn-Smith, J. (2016, September 13). Computational thinking from a dispositions perspective. Retrieved from https://www.blog.google/outreach-initiatives/education/computational-thinking-dispositions-perspective/.

Sullivan, F. R. (2008). Robotics and science literacy: Thinking skills, science process skills and systems understanding. *Journal of Research in Science Teaching*, *45*(3), 373-394.

Stake, R. (1995). The art of case study research. Thousand Oaks, CA: Sage.

Sung, W., Ahn, J., & Black, J. B. (2017). Introducing computational thinking to young learners: Practicing computational perspectives through embodiment in mathematics education. Technology, Knowledge and Learning, 22(3), 443-463

Sweller, J. (1988). Cognitive load during problem solving: Effects on learning. *Cognitive science*, *12*(2), 257-285.

Tierney, J. (2015). The dilemmas of maker culture. Retrieved from http://www.theatlantic.com/technology/archive/2015/04/the-dilemmas-of-maker-culture/390891/ on April 15, 2015.

Touretzky, D. S., Marghitu, D., Ludi, S., Bernstein, D., and Ni, L. (2013) Accelerating computational thinking using scaffolding, staging, and abstraction. *Proceedings of SIGCSE '13*, Denver, CO. Association for Computing Machinery, pp. 609-614.

Tucker, A. (2003). A Model Curriculum for K--12 Computer Science: Final Report of the ACM K--12 Task Force Curriculum Committee.

Turkle, S., & Papert, S. (March 01, 1992). Epistemological Pluralism and the Revaluation of the Concrete. Retrieved November 16, 2016 from http://www.papert.org/articles/EpistemologicalPluralism.html

U.S. Bureau of Labor Statistics. (2013). Careers in the growing field of information technology services : Beyond the Numbers. Retrieved March 12, 2016, from http://www.bls.gov/opub/btn/volume-2/careers-in-growing-field-of-information-technology-services.htm

U.S. Bureau of Labor Statistics. (2019). Employment Projections. Retrieved April 27, 2019, from U.S. Bureau of Labor Statistics website: https://www.bls.gov/emp/tables.htm

Vanderlinde, R., & van Braak, J. (April 01, 2010). The Gap between Educational Research and Practice: Views of Teachers, School Leaders, Intermediaries and Researchers. British Educational Research Journal, 36, 2, 299-316.

Vossoughi, S. & Bevan, B. (October, 2014). Making and Tinkering: A Review of the Literature. *National Research Council Committee on Out of School Time STEM*: 1-55.

Vossoughi, S., Escudé, M., Kong, F., & Hooper, P. (2013). Tinkering, learning & equity in the after-school setting. Paper presented at FabLearn, Stanford, CA. Retrieved March 12, 2016 from: http://fablearn.stanford.edu/2013/papers/.

Vye, N.J., D.L. Schwartz, J.D. Bransford, B.J. Barron, L. Zech, and Cognition and Technology Group at Vanderbilt 1998. SMART environments that support monitoring, reflection, and revision. In *Metacognition in Educational Theory and Practice*, D. Hacker, J. Dunlosky, and A. Graessner, eds. Mahwah, NJ: Erlbaum.

Vygotskiĭ, L. S., & Cole, M. (1978). *Mind in society: The development of higher psychological processes*. Cambridge: Harvard University Press.

Wade, S. E. (1992). How interest affects learning from text. In Renninger, A., Hidi, S., and
Krapp, A. (eds.), *The Role of Interest in Learning and Development*, Erlbaum, Hillsdale,
NJ, pp. 281–296.

Wade, S., Buxton, W., & Kelly, M. (1999). Using Think-Alouds to Examine Reader-Text Interest. *Reading Research Quarterly*, *34*(2), 194-216. Retrieved from http://www.jstor.org/stable/748164

Wagner, A., Gray, J., Corley, J., & Wolber, D. (2013, March). Using app inventor in a K-12 summer camp. *Proceeding of the 44th ACM technical symposium on Computer science education* (pp. 621-626). ACM. doi:10.1145/2445196.2445377

Weintrop, D., Beheshti, E., Horn, M. S., Orton, K., Trouille, L., Jona, K., Wilensky, U. (2014). Interactive Assessment Tools for Computational Thinking in High School STEM Classrooms.

Weintrop, D., Beheshti, E., Horn, M. S., Orton, K., Jona, K., Trouille, L., Wilensky, U. (2015). Defining Computational Thinking for Mathematics and Science Classrooms. *Journal of Science Education and Technology*, 1-21. doi:10.1007/s10956-015-9581-5

West-Puckett, S. (2013, September 13). ReMaking Education: Designing Classroom Makerspaces for Transformative Learning. Retrieved May 30, 2017, from https://www.edutopia.org/blog/classroom-makerspaces-transformative-learning-stephanie-west-puckett

Wilensky, U. (1991). Abstract meditations on the concrete and concrete implications for mathematics education. In I. Harel & S. Papert (Eds.), *Constructionism* (pp. 193–204). New York, NY: Ablex Publishing.

Wilensky, U. (January 01, 1995). Paradox, programming, and learning probability: A case study in a connected mathematics framework. *Journal of Mathematical Behavior*, *14*, 2, 253-280.

Wilensky, U., Brady, C. E., & Horn, M. S. (August 01, 2014). Fostering Computational Literacy in Science Classrooms. *Communications of the ACM*, 57, 8.

Wilkerson-Jerde, M. H. (February 01, 2014). Construction, Categorization, and Consensus: Student Generated Computational Artifacts as a Context for Disciplinary Reflection. *Educational Technology Research and Development*, *62*, 1, 99-121.

Williams, R. F. (2006, June). Using cognitive ethnography to study instruction. *In Proceedings of the 7th international conference on Learning sciences* (pp. 838-844). International Society of the Learning Sciences.

Wing, J. M. (2006). Computational thinking. *Communications-ACM*, *49*, 3, 33-35.

Wing, J. M. (January 01, 2008). Computational thinking and thinking about computing. *Philosophical Transactions. Series A, Mathematical, Physical, and Engineering Sciences*, *366*, 1881, 3717-25.

Wing, J. M. (2010). Computational Thinking: What and Why? Retrieved October 1, 2016 from https://www.cs.cmu.edu/~CompThink/resources/TheLinkWing.pdf

Winner, E., Hetland, L., Veenema, S., Sheridan, K., Palmer, P., & Locher, I. (2006). Studio thinking: How visual arts teaching can promote disciplined habits of mind. New directions in aesthetics, creativity, and the arts, 189-205.

Wolz, U., Stone, M., Pearson, K., Pulimood, S. M., & Switzer, M. (July 01, 2011). Computational Thinking and Expository Writing in the Middle School. *ACM Transactions on Computing Education*, *11*, 2.)

Wood, D., Bruner, J. S., & Ross, G. (1976). The role of tutoring in problem solving. *Journal of child psychology and psychiatry*, *17*(2), 89-100.

Yadav A., & Cooper, S. (2017). Fostering creativity through computing. *Communications of the ACM*, *60*(2), 31-33. doi: 10.1145/3029595

Yang, Z., Lian, D., Yuan, N. J., Xie, X., Rui, Y., & Zhou, T. (March 01, 2017). Indigenization of urban mobility. *Physica A: Statistical Mechanics and Its Applications*, 469, 232-243.

Yin, R.K. (2003). Case study research: Design and methods (3rd ed). Los Angeles, Sage Publications.

Young, M. R. (2002). Experiential learning= hands-on+ minds-on. Marketing Education Review, 12(1), 43-51.

Zhang, J. (1997). The nature of external representations in problem solving. Cognitive Science, 21(2), 179-217.

APPENDICES

Appendix 1: Program Schedule 2016

Week 1 - M-6/27		T-6/28		W-6/29		TH-6/30		F-7/1	
10:00-11:00	Introductions & Team-building -	10:00-10:15	Recap (make a circuit)	10:00-10:15	Recap	10:00	Recap	10:00-10:15	Teambuilding (build circuit with bodies)
11:00-12:00	Demographic Survey & Pre-Test	10:15-10:30	CT (Pattern Recognition)	10:15-10:30	CT (Pattern Generalization)	10:15-10:30	CT (Abstraction)	10:15-10:30	Recap (how does this fit in with your life)
12:00-12:30	Final prompt	10:30-10:45	Team-building: pattern recognition	10:30-10:45	Team-building	10:30-10:45	Team-building	10:30-10:45	CT (Decomposition) - ask students for examples
12:30-12:45	Interactive Demo: Make a simple circuit with incandescent bulb	10:45-11:15	Demo: E-Textiles	10:45-11:00	Interactive Demo: Electromagnet	10:45-11:30	Closed Make: Simple Motor	10:45-11:30	Interactive Demo I: Make Clock with and without Breadboard
12:45-1:30	Open Make: Notebooks // Simple Indicator (Making an Indicator), encourage scale	11:15-12:00	Demo and Open Make: E-textile (individual)	11:00-12:00	Closed make: Paperclip challenge (Ephran and Jeff)	11:30-12:00	Exploration/Demo: Conventional Motors, Dynamos, and Power	11:30-12:00	Interactive Demo II: Using Electrical Components with Breadboards (with Ephran's Drawing)
1:30-2:00	Notebook // Recap // (teambuilding)	12:00-12:30	Lunch	12:00-12:30	Lunch vid on magnet rube machine	12:00-12:30	Lunch	12:00-12:30	Lunch
		12:30-1:45	Open Make: E-textile (individual)	12:30-1:00	Open Make: Stuffed Animals Animatronics	12:30-1:45	Open Make: Integrate Motors Into Circuit	12:30 - 1:15	Open Make: Breadboard with Components; Show circuit boards
		1:45-2:00	Recap	1:00-1:45		1:45-2:00	Recap (strengths/weaknesses)	1:15-2:00	Notebook // Recap // (teambuilding)

T – 7/5		W – 7/6		TH – 7/7		F – 7/8	
10:00-10:15	Recap	10:00-10:15	Recap	10:00-10:15	Recap	10:00-10:15	Recap
10:15-10:30	CT (Algorithm Design)	10:15-11:00	Tabletop breadboard	10:15-10:30	Team-building	10:15-11:15	Post-test
10:30-10:45	Teambuilding: algorithm design	11:00-12:00	Breadboard exploration	10:30-12:00	Open Make: Final Project	11:15-12:00	Final Touches
10:45-12:00	Closed Make: MaKey MaKey (Cardboard challenge)	12:00-12:30	Lunch	12:00-12:30	Lunch	12:00-12:30	Lunch
12:00-12:30	Lunch	12:30-1:00	Closed and Open Make: Arduino	12:30-1:45	Open Make: Final Project	12:30-1:45	Presentations
12:30-1:45	Counting and scratch (as option among Scratch projects)	1:00-1:45	Final Prompt & examples & create groups	1:45-2:00	Recap	1:45-2:00	Recap & Goodbyes
1:45-2:00	Recap	1:45-2:00	Recap	2:15-3:00	Team meeting	2:15-3:00	Team meeting
2:15-3:00	Team meeting	2:15-3:00	Team meeting	3:00-3:45	Research	3:00-3:45	Research
3:00-3:45	Research						

Appendix 2: Methods Overview

Question	Data Collected to Answer the Question	How Analysis of Data will Answer the Question	Imagined Answer
Within makerspace activities what evidence exists regarding promising practices that support student development of CT skills and dispositions?	**Audio, Video, Screen recordings:** Transcripts of audio recordings, supported by video and screen recordings **Observational field notes:** Observance from researchers of daily sessions **Notebooks**: Observance of student notebook entries **Photographs:** Pictures of student artifacts	The researcher will code and memo transcripts, field notes, and notebooks for actions to determine themes for practices that elicit CT. Developed from these themes will be a coherent theory that has explanatory and predictive power for how certain practices contribute to the development of CT skills.	1. Debugging 2. Documentation in notebooks 3. Cultural responsiveness 4. Community building 5. Use-Modify-Create

Appendix 3: Descriptive Observation Protocol

Descriptive Observation Instrument

We are using this tool in order to find "moments of notice," to highlight when students are engaged/challenged, and when they are using computational thinking skills or physics knowledge. In particular, we want to observe instructor-student, staff-student, instructor-staff, and student-student interactions which facilitate engagement and learning.

Observer's Name:

Date:

Time Observation Began:

Time Ended:

1. Before the observation begins, briefly describe what the group or individual you are observing is working on.

What to observe?	Observation Notes
Who are you observing?	
What activity are they working on?	

What is the goal of the activity?	

2. Describe the setting

What to observe?	Observation Notes

How are the students and mentors arranged? Where they are in the room?	(Pictures)
What is their demeanor/mood as they come in?	
Other notes	

3. Describe how the session begins (Who is present, what exactly was said at the beginning).

4. Describe the chronology of events

Time	What happened

Appendix 4: Participant Recruitment Script

Hi! I am here to invite you to be a part of a study. In this study, we hope to understand how students develop problem-solving skills. We will do this while you create projects in a makerspace at the Harold Washington Library downtown. We will observe you as you work, ask you questions, and look at your work.

You will be videotaped, but raw video and sound data will be kept in a locked file cabinet at CCAS. The data will be kept for 3 years after the study ends and then destroyed.

You do not have to be in this study if you do not want to. If you agree to be in the study, but later change your mind, you may drop out at any time. There are no penalties of any kind if you decide that you do not want to participate. Please talk it over with your parents before you decide whether or not to participate. We will also ask your parents to give their permission for you to take part in this study. But even if your parents say "yes" you can still decide not to do this. In addition to parental permission, you will need to sign an assent form and a video release form. There will be an envelope in the GEAR UP office where you can drop off the forms. Please do this whether or not you want to participate in the study; just mark off the "yes" or "no" fields. You must turn in your forms by next week.

If you have questions about the study you can call or email the researchers, whose information is on the form I'm going to hand out.

Appendix 5: Video release form

Northeastern Illinois University

University of Illinois at Chicago

VIDEO RELEASE FORM

Assessing Computational Thinking in Maker Activities (ACTMA)

As part of this project, the researchers of this project will be making video recordings of you/your child during their participation in the research. Please indicate what uses of these video recordings you are willing to permit, by putting your initials next to the uses you agree to, and signing the form at the end. This choice is completely up to you. The researchers will only use the video recordings in ways that you agree to. In any use of the tapes, you will not be identified by name.

Participant signature	Guardian signature	
		The video recordings can be studied by the research team for use in the research project.
		The video recordings can be used for scientific publications.
		The video recordings can be shown at scientific conferences or meetings
		The video recordings can be posted to a web site.

I have read the above descriptions and give my consent for the use of the video recordings as indicated by my initials above.

Name____

____ ___________________

(Signature) (Date)

Guardian Name (if participant is under18)_______________________________________

____ ___________________

(Guardian Signature) (Date)

Appendix 6: Participant assent

Northeastern Illinois University

University of Illinois at Chicago

ASSENT TO PARTICIPATE IN RESEARCH

Assessing Computational Thinking in Maker Activities (ACTMA)

Researchers: Roxana Hadad, Director of Math, Science and Technology at the Center for College Access and Success at Northeastern Illinois University; Yue Yin, Associate Professor of Educational Psychology at University of Illinois at Chicago

Introduction

You are invited to be a part of a research study. Participation is voluntary and the decision whether or not to participate will not affect any relationship with the school, program, NEIU, or UIC. In this study, we hope to understand how students develop problem-solving skills. We will do this while you create projects in the makerspace.

About the Study

This study is a partnership between the Center for College Access and Success at Northeastern Illinois University (NEIU), the Department of Educational Psychology at the University of Illinois at Chicago (UIC), the Chicago Public Library, and the Museum of Science and Industry. The Principal Investigator is Roxana Hadad (NEIU) and the Co-Principal Investigator is Dr. Yue Yin (UIC).

Why is this study being done?

What we learn from the study could help make science programs for children better.

What will I be asked to do?

All you have to do is be a part of the program at the Harold Washington Chicago Public Library. We will observe you as you work, ask you questions, and look at your work. You and your work may also be photographed or video/audio recorded while you work and during focus groups. You and your work may be recorded for the entire 3 hours every day of the two week (9 day) session. We will conduct focus group sessions during the program.

I agree that:

____ I will allow the researchers to observe my activity in the makerspace.

____I will allow questions about my work to be asked of me.

____I will allow the work I create from this project to be looked at.

____I will allow the researchers to record the things I do and say through audio and video.

What are the risks of the study?

There is the risk that a breach of privacy (others will know the you are participating in research) and confidentiality (accidental disclosure of identifiable data) may occur. All data will be protected on a secure network under password protection. The password will be known only to the PI and Co-PI.

What are the benefits of the study?

You may participate in activities that you enjoy and may work with people to make your projects better. However, this study is not designed to benefit you directly. This study is designed to learn more about how we can measure problem-solving skills. What we learn from the study could help make science programs for children better.

How will my information be protected?

Raw video and sound data will be kept in a locked file cabinet at CCAS. Consent forms will be kept separate from any data. Digital files, including video, sound, raw, and other data recordings, will be kept private for at least 3 years after the study ends. The researchers will only access the data from a password-protected computer. At the end of the study, the researcher might share what they learn, but no names will be used.

Can I stop being in the study? What are my rights?

You do not have to be in this study if you do not want to. If you agree to be in the study, but later change your mind, you may drop out at any time. There are no penalties of any kind if you decide that you do not want to participate. Please talk it over with your parents before you decide whether or not to participate. We will also ask your parents to give their permission for you to take part in this study. But even if your parents say "yes" you can still decide not to do this.

What if I have questions?

If you have questions about the study you can call or email the researchers:

Roxana Hadad at 312-563-7218 or r-hadad@neiu.edu

Yue Yin at 765-430-3545 or yueyin@uic.edu

If you are worried about your rights, are worried about anything in this study, or if you have a research-related problem, you can call Northeastern Illinois University's IRB at (773) 442-4670 or University of Illinois at Chicago IRB at (312) 996-1711.

Assent:

I have read this form and decided whether or not I will be a part of Assessing Computational Thinking in Maker Activities (ACTMA) Introduction study. Why this study is being done, what I have to do, and the risks have been explained to me in a way that I understand. I know that I can drop out of the study at any time. I will receive a copy of this form.

Please sign your name and mark a check whether you would like to be a part of the study or not.

____ ____________________

Participant Signature Date

____

Print Name

____

School Name

I would like to participate in the study

Yes ________

No ________

Please mark all that apply

__ I consent for myself and my work to be audio recorded

__ I consent for myself and my work to be video recorded

__ I consent for myself and my work to be photographed

_______________________________________ _____________________

Signature of Person Obtaining Assent (STAFF ONLY) Date

____

Print Name

____ _____________________

Signature of Person Obtaining Assent (STAFF ONLY) Date

____

Print Name

Appendix 7: English parental permission

THE CENTER FOR COLLEGE ACCESS AND SUCCESS

PRINCIPAL AND CO-PRINCIPAL INVESTIGATORS: Roxana Hadad, Director of Math, Science and Technology at the Center for College Access and Success at Northeastern Illinois University; Yue Yin, Associate Professor of Educational Psychology at University of Illinois at Chicago

TITLE OF STUDY: Assessing Computational Thinking in Maker Activities (ACTMA)

<u>ABOUT THE STUDY:</u> This research study is a partnership between the Center for College Access and Success at Northeastern Illinois University (NEIU), the Department of Educational Psychology at the University of Illinois at Chicago (UIC), the Chicago Public Library, and the Museum of Science and Industry. The Principal Investigator is Roxana Hadad (NEIU) and the Co-Principal Investigator is Dr. Yue Yin (UIC). Participation is voluntary and the decision whether or not to have your child participate will not affect any relationship with the school, program, NEIU, and or UIC. Your child may participate in the program without allowing their products to be used for research. Video and voice recording will be done to document evidence of learning throughout the workshop sessions and focus groups.

<u>PROCEDURES INVOLVED IN THIS STUDY</u>: The National Science Foundation is funding this study in which your child will be asked to be a part of the YOUmedia

program at Harold Washington Chicago Public Library. In the program, your child will work on hands-on projects about science. What they do and what they create in the program will be examined. The goal of this study is to measure children's problem-solving skills. There will be about 6-12 students and 2-5 mentors in this study, meeting every day for 4 hours for 9 days over 2 weeks. Students will be asked to create projects and talk about their learning and thinking. Video and sound recording and photographs will be used to study how students and mentors make things and solve problems together. Students will be observed while they work and will be involved in focus groups during this time to better understand their learning.

POSSIBLE BENEFITS: Students may participate in activities that they enjoy and learn from. However, this study is not designed to benefit them directly. This study is designed to learn more about how we can measure problem-solving skills. What we learn from the study could help make science programs for children better.

POSSIBLE RISKS: There is the risk that a breach of privacy (others will know your child is participating in research) and confidentiality (accidental disclosure of identifiable data) may occur. All data will be protected on a secure network under password protection. The password will be known only to the PI and Co-PI.

YOUR CHILD'S PARTICIPATION AS A RESEARCH SUBJECT: Your child does not have to be in this study if you do not want them to. If you agree your child can be in the study, but later change your mind, your child may drop out at any time. There are no penalties of any kind if you decide that you do not want your child to participate. Your child and their work may also be photographed or video recorded during focus groups during the entire duration of the program.

CONFIDENTIALITY: Raw video and sound data will be kept in a password-protected computer at CCAS. Signed forms will be kept separate from any data in a locked file cabinet at CCAS. Digital files, including video, sound, and other data recordings, will be kept private for at least 3 years after the study ends. The researchers will only be able to get the data from a password-protected computer. At the end of the study, the researcher might share what they learn, but no names will be used.

CONTACT FOR QUESTIONS: If you have any questions regarding your participation, please feel free to contact the researchers, Roxana Hadad, at 312-563-7218 or r-hadad@neiu.edu or Yue Yin at 765-430-3545 or yueyin@uic.edu, and they will gladly inform you. If you have any questions regarding your child's rights as a participant you can contact the NEIU Institutional Review Board at (773) 442-4674 or at IRB@neiu.edu, or the UIC Institutional Review Board at (312) 996-1711.

<u>PARENT'S CONSENT:</u> If you would like your child to participate, please read and sign the consent form.

I have read the above information about the study and have been able to express questions and concerns, which have been satisfactorily responded to by the research investigator. I believe I understand the study.

____

Student Name

____ ____________________

Signature (Participant or Legally Authorized Representative) Date

____

PRINT NAME (Participant/ Legally Authorized Representative)

- ❑ Yes, I consent for my child to participate in this research study
- ❑ No, I do not consent for my child to participate in this research study

Raw video and sound data will be kept in a password-protected computer at CCAS. Signed forms will be kept separate from any data in a locked file cabinet at CCAS. Digital files, including video, sound, and other data recordings, will be kept private for at least 3 years after the study ends. The researchers will only be able to get the data from a password-protected computer. At the end of the study, the researcher might share what they learn, but no names will be used. Videotaping consent is required for research participation.

Please mark all that apply

❑ Yes, I consent for my child and their work to be audio recorded	❑ No, I do not consent for my child and their work to be audio recorded
❑ Yes, I consent for my child and their work to be video recorded	❑ No, I do not consent for my child and their work to be video recorded
❑ Yes, I consent for my child and their work to be photographed	❑ No, I do not consent for my child and their work to be photographed

____ ____________________

Signature of Investigator Date

____

PRINT NAME (Investigator)

THIS RESEARCH PROJECT/STUDY HAS BEEN REVIEWED BY NEIU/UIC'S REVIEW BOARD FOR THE PROTECTION OF HUMAN PARTICIPANTS.

Appendix 8: Spanish parental permission

THE CENTER FOR COLLEGE ACCESS AND SUCCESS

INVESTIGADORES: Roxana Hadad, Directora de Matemáticas, Ciencias y Tecnología en el Centro para Acceso y Éxito Universitario de la Universidad de Northeastern Illinois, NEIU (por sus siglas en inglés); Yue Yin, Profesor Asociado de Psicología Educacional en la Universidad de Illinois en Chicago, UIC (por sus siglas en inglés).

TÍTULO DEL ESTUDIO: Evaluando el Pensamiento Computacional en Actividades Manuales, ACTMA (por sus siglas en inglés)

ACERCA DEL ESTUDIO: Este estudio de investigación es llevado a cabo por el Centro para Acceso y Éxito Universitario de la Universidad de Northeastern Illinois (NEIU), el Departamento de Psicología Educacional de la Universidad de Illinois en Chicago (UIC), la Biblioteca Pública de Chicago, y el Museo de Ciencia e Industria. Los investigadores son Roxana Hadad (NEIU) y Yue Yin (UIC). La participación es voluntaria y la decisión de ser o no ser parte del estudio no afectará las relaciones de su hijo con la escuela, el programa, NEIU, o UIC. Su hijo puede ser parte del programa sin que sus proyectos sean utilizados para el estudio. Se emplearán grabaciones de video y audio para documentar y evidenciar el aprendizaje mediante los talleres y grupos de enfoque.

PROCEDIMIENTOS IMPLICADOS EN ESTE ESTUDIO: La Fundación Nacional de Ciencia, NSF (por sus siglas en inglés), financia este estudio en el cual se le pedirá a su hijo ser parte del programa YOUmedia en la Biblioteca Pública de Chicago Harold Washington. En el programa, su hijo trabajará en proyectos prácticos sobre ciencia. Durante el programa se observará lo que los participantes hagan y creen y se les hará preguntas. El objetivo de este estudio es crear una forma de medir las habilidades de pensamiento computacional (destrezas para resolver problemas). Habrá alrededor de 6-12 estudiantes y 2-5 mentores en este estudio. Estos se reunirán 3 horas diarias en un periodo de 9 días sobre un lapso de 2 semanas. Se les pedirá a los estudiantes que creen proyectos y que hablen sobre su proceso de aprendizaje y pensamiento. Video, audio y fotografías serán utilizados para estudiar como los estudiantes y mentores crean cosas y resuelven problemas. Para entender la forma de aprendizaje de los estudiantes ellos serán observados mientras trabajan, al mismo tiempo que estarán envueltos en grupos de enfoque.

<u>POSIBLES BENEFICIOS:</u> Los estudiantes participarán en actividades en las que aprenderán al mismo tiempo que se divertirán. Sin embargo este estudio no está diseñado para beneficiarlos directamente. El estudio está diseñado para aprender formas de medición sobre las destrezas para resolver problemas. Lo que se aprenderá mediante este estudio servirá para la mejorar los programas de ciencias para los estudiantes.

<u>POSIBLES RIESGOS:</u> Existe riesgo de violación de privacidad (otros sabrán que su hijo está participando en el estudio) y confidencialidad (revelación accidental de data identificable). Una red segura con contraseña protegerá la data. Solamente el PI y Co-PI sabrán la contraseña

<u>PARTICIPACIÓN DE SU HIJO COMO SUJETO DE INVESTIGACIÓN:</u> Su hijo no tiene que participar en el estudio si usted no lo desea. Si usted está de acuerdo en que participe pero luego cambia de opinión, su hijo puede salir del programa en cualquier momento. No hay ningún tipo de penalidad si usted no desea que su hijo participe. Su hijo y el trabajo de su hijo pueden ser fotografiados o video grabados durante el fin de semana y durante toda la sesión de dos horas en los grupos de enfoque.

<u>CONFIDENCIALIDAD:</u> La data grabada en videos y audio serán mantenidos en una computadora con acceso restringido en CCAS. Los formularios que contengan firmas serán separados de cualquier otra data y guardados en un gabinete de archivo en CCAS. Archivos digitales, incluyendo video, audio u otra data grabada serán mantenidos en privado por lo menos 3 años después de que termine el estudio. Los investigadores podrán obtener acceso a la data, la cual se archivará en una computadora protegida con contraseña. Al final del estudio, los investigadores tendrán la opción de compartir sus resultados, pero no publicarán nombres.

<u>PREGUNTAS:</u> Si tiene preguntas acerca del estudio o cualquier problema relacionado con la investigación, puede llamar a las personas responsables del estudio:
Roxana Hadad at 312-563-7218 o r-hadad@neiu.edu o Yue Yin at 765-430-3545 o yueyin@uic.edu. Si tiene alguna duda o pregunta sobre la investigación puede ponerse en contacto con la oficina del IRB de la Universidad de Northeastern Illinois al teléfono

(773) 442-4670 o a la oficina del IRB de la Universidad de Illinois en Chicago al teléfono (312) 996-1711.

PERMISO: Por favor lea y firme el permiso si desea que su hijo participe en el estudio.

He leído la información sobre el estudio y he podido hacer preguntas y expresar preocupaciones que han sido satisfactoriamente respondidas por la investigadora. Creo que entiendo el estudio.

____

Nombre del estudiante

____ __________________

Firma del padre o tutor legal Fecha

____ __________________

NOMBRE IMPRESO (padre o tutor) Fecha

____ __________________

Firma del investigador Fecha

____ __________________

NOMBRE IMPRESO (investigador) Fecha

- ❑ Sí doy mi permiso para que mi hijo participe en este estudio de investigación.
- ❑ No doy mi permiso para que mi hijo participe en este estudio de investigación

La data grabada en videos y audio serán mantenidos en una computadora con acceso restringido en CCAS. Los formularios que contengan firmas serán separados de cualquier otra data y guardados en un gabinete de archivo en CCAS. Archivos digitales, incluyendo video, audio u otra data grabada serán mantenidos en privado por lo menos 3 años después de que termine el estudio. Los investigadores podrán obtener acceso a la data, la cual se archivará en una computadora protegida con contraseña. Al final del estudio, los investigadores tendrán la opción de compartir sus resultados, pero no publicarán nombres. El premiso de grabación es un requerimiento para la participación de este estudio.

(VER EL OTRO LADO)

___Si doy permiso para que mi hijo y su trabajo sean audio grabados.	___No doy permiso para que mi hijo y su trabajo sean audio grabados.
___Si doy permiso para que mi hijo y su trabajo sean video grabados.	___No doy permiso para que mi hijo y su trabajo sean video grabados.
___Si doy permiso para que mi hijo y su trabajo sean fotografiados.	___No doy permiso para que mi hijo y su trabajo sean fotografiados.

ESTE PROYECTO DE INVESTIGACIÓN/ESTUDIO HA SIDO REVISADO POR EL DIRECTORIO DE REVISIONES INSTITUCIONALES DE NEIU PARA LA PROTECCIÓN DE LOS PARTICIPANTES.

Appendix 9: Facilitator/Mentor consent

THE CENTER FOR COLLEGE ACCESS AND SUCCESS

PRINCIPAL AND CO-PRINCIPAL INVESTIGATORS: Roxana Hadad, Director of Math, Science and Technology at the Center for College Access and Success at Northeastern Illinois University; Yue Yin, Associate Professor of Educational Psychology at University of Illinois at Chicago

TITLE OF STUDY: Assessing Computational Thinking in Maker Activities (ACTMA)

ABOUT THE STUDY: This research study is a partnership between the Center for College Access and Success at Northeastern Illinois University (NEIU), the Department of Educational Psychology at the University of Illinois at Chicago (UIC), the Chicago Public Library, and the Museum of Science and Industry. The Principal Investigator is Roxana Hadad (NEIU) and the Co-Principal Investigator is Dr. Yue Yin (UIC). Participation is voluntary and the decision whether or not to participate will not affect any relationship with the school, program, NEIU, and or UIC. You may participate in the program without allowing products to be used for research. Video and voice recording will be done to document evidence of learning throughout the workshop sessions and focus groups.

PROCEDURES INVOLVED IN THIS STUDY: The National Science Foundation is funding this study in which children will be asked to be a part of the YOUmedia program at Harold Washington Chicago Public Library. Mentors will learn about hands-on projects that focus on physics that they can do with the students. What the students and mentors do and what they create in the program will be looked at and they will be asked questions. The goal of this study is to be able to create a way to measure students' computational thinking (a type of problem solving) skills. There will be about 6-12 students and 2-5 mentors in this study, meeting every day for 4 hours for 9 days over 2 weeks. The work students and mentors create will be recorded by video, audio, and computer software. Focus groups will take place during the program.

<u>POSSIBLE BENEFITS:</u> The study results will help us make better tools to improve problem-solving skills. What we learn from this study could help educational programs in science, technology, engineering and math in which you may take part in one day.

<u>POSSIBLE RISKS:</u> There is the risk that a breach of privacy (others will know the you are participating in research) and confidentiality (accidental disclosure of identifiable data) may occur. All data will be protected on a secure network under password protection. The password will be known only to the PI and Co-PI.

<u>YOUR PARTICIPATION AS A RESEARCH SUBJECT:</u> You do not have to be in this study if you do not want to. If you agree to be in the study, but later change your mind, you can drop out at any time. There are no penalties of any kind if you decide that you do not want to participate. You and your work may also be photographed or video recorded

during focus groups. You and your work may be recorded for the entire duration of the two hour session each weekend.

CONFIDENTIALITY: Raw video and sound data will be kept in a locked file cabinet at CCAS. Consent forms will be kept separate from any data. Digital files, including video, sound, raw, and other data recordings, will be kept private for at least 3 years after the study ends. The researchers will only access the data from a password protected, university network. At the end of the study, the researcher might share what they learn, but no names will be used.

CONTACT FOR QUESTIONS: If you have any questions regarding your participation, please feel free to contact the researchers, Roxana Hadad, at 312-563-7218 or r-hadad@neiu.edu or Yue Yin at 765-430-3545 or yueyin@uic.edu, and they will gladly inform you. If you have any questions regarding your rights as a participant you can contact the NEIU Institutional Review Board at (773) 442-4674 or at IRB@neiu.edu, or the UIC Institutional Review Board at (312) 996-1711.

<u>PARTICIPANT'S CONSENT:</u> If you would like to participate, please read and sign the consent form.

I have read the above information about the study and have been able to express questions and concerns, which have been satisfactorily responded to by the research investigator. I believe I understand the study.

____ ____________

Signature (Participant or Legally Authorized Representative) Date

____ ____________

PRINT NAME (Participant/ Legally Authorized Representative) Date

____ ________________

Signature of Investigator Date

____ ________________

PRINT NAME (Investigator) Date

Please mark one:

___ Yes, I consent to participate in this research study

___ No, I do not consent to participate in this research study

Please mark all that apply

__ I consent for myself and my work to be audio recorded

__ I consent for myself and my work to be video recorded

__ I consent for myself and my work to be photographed

THIS RESEARCH PROJECT/STUDY HAS BEEN REVIEWED BY NEIU/UIC'S REVIEW BOARD FOR THE PROTECTION OF HUMAN PARTICIPANTS.

Appendix 10: Notice of Videotaping

NOTICE:

RESEARCHERS ARE

OBSERVING PARTICIPANTS IN

THIS AREA

VIDEOTAPING IN PROGRESS

Appendix 11: Examples of Findings

Examples of participants blindly guessing while tinkering

<table>
<tr><th>Activity</th><th>Example</th><th>Notes on example</th></tr>
<tr><td>Tabletop breadboard</td><td>Sam [to Ezra]: There seems to be a little bit of confusion on how to connect batteries in series just using the breadboard. They seem to just be taking the top out.</td><td>One facilitator (Sam) notices that despite working with the breadboards the day before, the participants are still unclear on how to power their breadboards and are randomly placing the battery at the top. He mentions this to the lead facilitator (Ezra).</td></tr>
<tr><td>Breadboard exploration</td><td>Researcher notes: Kids are just fiddling around … not completely understanding how the circuit works. [Another researcher] is reminded of paper on “Hands On, Minds On”</td><td>A researcher’s observation notes point out the participants plugging components into their breadboards randomly without thinking through their actions.</td></tr>
<tr><td>Arduino</td><td>Denise and Ana change where the variable “led” is called to turn on the light to “13” then to “14” then back to “13” and then change where variable “led” is called to turn off to “13”.


</td><td>After lead facilitator led the participants through the process of opening the boxes with the Arduinos, plugging them into their computers and sending them code to make an LED on a breadboard blink, he asked the participants to “change some of these variables and see what happens.” He explains how the code calls for certain pin numbers in the breadboard to be activated and suggests they change the values for the delay.

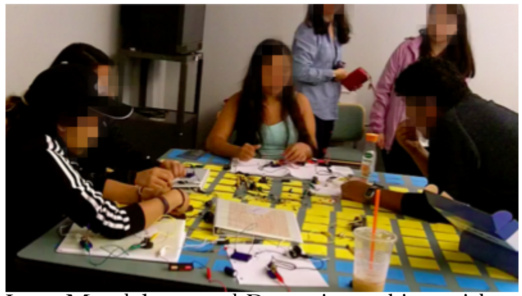
Two participants (Denise and Ana) demonstrate the haphazard way the participants began exploring the Arduino environment. Instead of changing the value assigned to the delay in order to change the blink pattern, the participants began changing where the variable was supposed to be called to turn on (HIGH) to the pin number in the breadboard where their LED was plugged in (13). They did this where it was called to be turned off (LOW), as</td></tr>
</table>

		well. They do not have an understanding of how programming variables work and how they should interact with the LED they are controlling.

Examples of tinkering, cognitive apprenticeship methods, CT skills and dispositions

CA method & CT skill and/or disposition	**Example**	**Notes**
Modelling and Scaffolding Evaluation (Tabletop breadboard activity)	Sam: Is there anything unnecessary in this circuit? Ezra: Yeah, can we get rid of jumpers? Will it work with less jumpers? Well, we have to connect things differently than right? Julio: She has the one... Ezra: Yeah, so what are you connecting to? Julio: I'm just connected to [inaudible] so you can get rid of me, I'm [inaudible]. Ezra: Yeah, we could get rid of Julio, right? But if Julio wanted to be in the circuit in parallel, as an LED, how would he do it? Could you plug him in this row and then this row? If he put his hands anywhere in this row or this row, he'd be in parallel, right?	As a group of participants are moving the people in their tabletop breadboard around, a facilitator (Sam) asks if there is anything a group can do to make their circuit more efficient, an example of what they themselves can ask every time they make a circuit or a piece of code. Another facilitator (Ezra) provides more scaffolding questions, asking if one participant (Julio) who is posing as a jumper, is necessary to the composition of their circuit. Julio acknowledges he is redundant, and then Ezra demonstrates how they could use him instead to
Modeling tolerance for ambiguity (Breadboard exploration activity)	Juan: Gotta figure out which pin is which. Do we have another battery? Magdalena: We need 2 batteries? Or do I switch it? Juan: Uhhh … just add another battery. So, were you in that group when we were voting? Magdalena: yeah Juan: Oh! Did we just burn it out? Do it again. Close it. Close it. Magdalena: Are you sure? Juan: Yeah, keep it closed. Do you see that? What am I doing? Magdalena: Is that one polar specific? Juan: It's not polar specific, actually. It's called a potentiometer and you can control how much energy flows through it, and you can go one of two ways. You want to experiment with it a little more? It might work with an LED. But here. Use your hands. When you get to the area you want, make small little tiny movements, okay? And it's just what you said, a	The facilitator (Juan) is encouraging the participants (Magdalena and Damaris) to try out different components available to them and see what they do and how they work. He is unsure about how to plug a potentiometer into a breadboard and is concerned he might have burned out a bulb in the process, but keeps working with the participants to get it working. Once they get it right, he points out how adjusting it slowly dims and brightens the bulb.

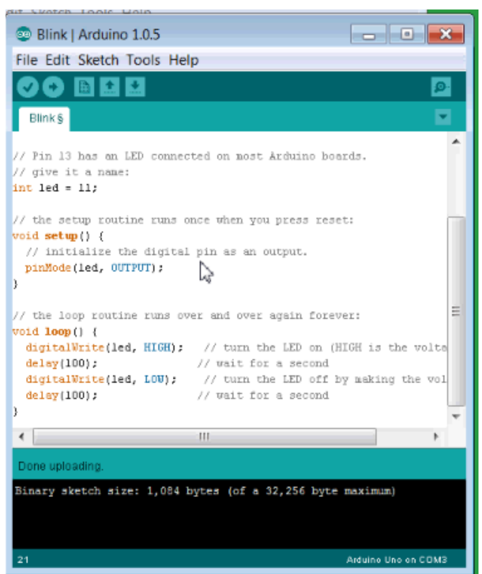
<table>
<tr><td></td><td>[inaudible] switch. Now go back and make very small changes.


Juan, Magdalena, and Damaris working with a potentiometer</td><td></td></tr>
<tr><td>Coaching decomposition (Arduino activity)</td><td>Sam: So, right now it's plugged into 13, right? And, the code is showing 13, so they're matched up.
Denise: This has to do with LED 13, if we change that and that, too?
Sam: So, what do you think the point of this line is?
Denise: To show where the LED is?
Sam: Yes. But, what about down here also? 'Cause, it looks like what you've been doing is changing it here, changing it here, right?</td><td>The facilitator (Sam) observes the participants (Denise and Ana) have been doing by changing where the led variable is being called to turn on and off. He asks them to look at the code line by line and think about what each part of the code is doing, in order for them to understand where they went wrong.</td></tr>
<tr><td>Scaffolding algorithmic thinking & confidence in dealing with complexity (Arduino activity)</td><td>Sam: Yeah, any time you see the numbers. You make it 100 or 1,000, like, what is one second equal to? What number is one second? Like, is one second 100, or is one second 1,000 ... 10,000? What number do you have to put in to make it be one second?
Denise: Oh, for one second.
Sam: Really quickly, before we move on to the next thing-
Denise: We're gonna just need 1,000.
Sam: 1,000. Exactly.
Sam: I want to see, instead of this being funneled indirectly ...[crosstalk] can you plug it in using [crosstalk]?
Sam: So, if you feed it 100, instead of 1,000, how long is it going to be on? How long do you expect it to be on?
Ana: Oh, five, it's half.
Sam: A half, or?
Denise: Maybe more. More and ...
Ana: Oh.
Denise: Would it, if we put 10,000 ...
Sam: Is it going to be longer or shorter?
Denise: Longer. Like, it's going to take longer to turn off and on.
Sam: Okay, let's see. Pretty long time, right? 'Cause, each 1,000 is a second. You've got 8,000s, 10,000s ... so, you can actually kind of predict how</td><td>The facilitator (Sam) then begins to step the participants (Denise and Ana) through the code by asking them questions and providing hints to how they might adjust their code and what might happen as a result of those adjustments.

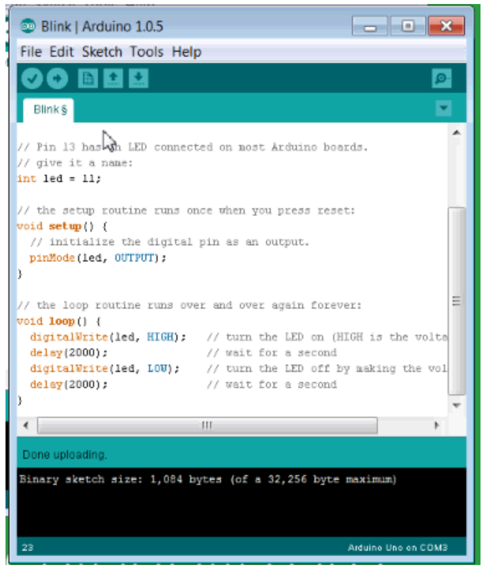
His questions push the participants to stop guessing and become more thoughtful about how they are adjusting the code and what those adjustments mean.</td></tr>
</table>

long it's going to be on. Exactly how long.
Denise: So, around 10 seconds.
Sam: Yeah. 'Cause, it's 10,000, right? So, what if you wanted it to like, be on for I don't know, 2.1 seconds?
Denise: 2,000.
Sam: About that, mm-hmm (affirmative).
Ana: And, you have to change both of them? All the time?
Sam: Well, that's an interesting question, 'cause what is this number controlling?
Denise: When it's on, and then when it's off. So, for it to be equal, like let's say if you want it to be under five, like one second, you leave it at 1,000. If you want it to be up at, for one second, so you leave it at 1,000. But, if you want to change it up, so I guess-
Sam: Do you want to change it up? It doesn't have to be on and off the same amount of time.
Ana: You gotta put one.
Denise: What did we put first [inaudible]?
Sam: So, you have 2,000, and 100? Okay. Huh.
Ana: So, it's off for ...
Sam: Very short. Mm-hmm (affirmative).

They change HIGH delay to 100 and then change HIGH and LOW delay to 2000.



Denise and Ana change HIGH delay to 100

	 Denise and Ana change HIGH and LOW delay to 2000	

Examples of embodying, cognitive apprenticeship methods, CT skills and dispositions

CA method & CT skill and/or disposition	Example	Notes on example
Scaffolding algorithmic thinking, communication, and collaboration (Tabletop breadboard activity)	Ezra: If you're having trouble you could try start holding hands, see if it makes sense, and then plug it into the breadboard. Elina: So what are we again? [laughing] He's a battery. I'm a jumper. Jackie: You're a jumper Elina: You're the other jumper. She could be a …. David: Make a...yeah. Elina: But don't we need a light. Jackie: And you could be a switch. Or you could be a switch and you could be the light bulb. Angela: The jumper. Elina: And then she's the jumper. Jackie: No, because then we we need another wire, don't we? Jackie: Yeah, yeah we do. To connect them Angela: To get the switch.[to Ezra] Can we get another mentor come it?	The facilitator (Ezra) provides the participants with some scaffolding, stating that they could hold hands first, as a basic circuit they are accustomed to, and then they can "plug" themselves into the tabletop breadboard. Another facilitator (David) begins to help, but then lets the participants do it on their own. Three of the participants in the group (Elina, Jackie, and Angela) bounce off of one another in putting together the circuit. They think out loud, telling one another how they think the circuit should be constructed as they are figuring it out themselves. Eventually, they realize that they need more bodies to complete the circuit and request another facilitator (Sam) to come and help them plug into the tabletop board.

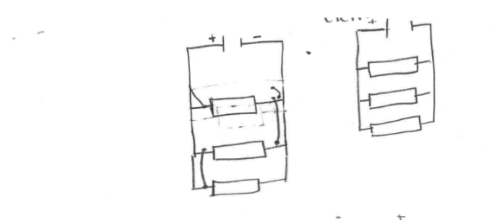
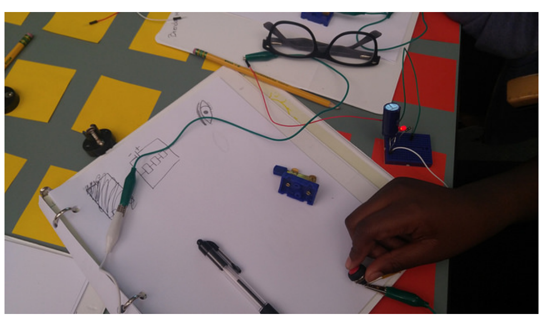
	Ezra: You need another hand? [to Sam] They need another hand.	
Articulation of algorithmic thinking (Tabletop breadboard activity)	Ezra: Alright, so how's this working? Jackie: Okay so I'm the battery connecting the positive and negative. She's a jumper. He's a ... Ezra: So she's the jumper but she's going out into my arm, right? Jackie: Yeah. Ezra: So I'm an LED, connected here. Follow the path. Jackie: Okay, so I'm the battery, so you're the LED light, she's a jumper, he's a jumper, and then he's the switch, he's the... Angela: He's the jumper. Then he's the light bulb.  Ezra demonstrating where the path of the electrons along the participants' bodies could go	The lead facilitator (Ezra), who is taking part in a group's tabletop circuit because they needed another body, asks the participants to describe how their circuit works. One participant (Jackie) takes the lead and walks him through the circuit, articulating how she sees the circuit working. Ezra asks her to confirm that a participant that is playing the role of a jumper is still on a path that connects to him, and traces the path by pointing with his hand. Jackie confirms that this is correct, and then Ezra encourages her to continue articulating the path. She does, but gets stuck when getting to another facilitator. Another participant (Angela) jumps in to continue the articulation and says that another facilitator (Sam) is the jumper and a participant (Julio) is the lightbulb.

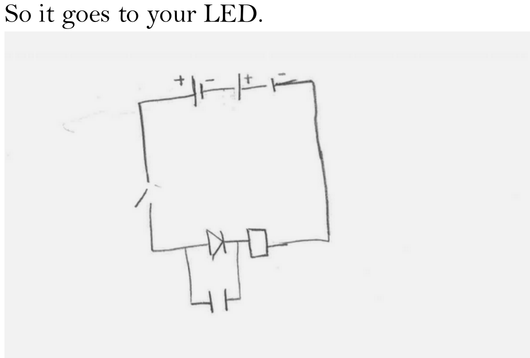
Examples of walkthroughs, cognitive apprenticeship methods, CT skills and dispositions

CA method & CT skill and/or disposition	Example	Notes on example
Articulation of decomposition (Breadboard exploration activity)	Angela: I want to do it parallel. David: To make it from that? When you say connect these two what do you mean? Let's hear it. Angela: Because this is one circuit, this is another. I want to connect both of them to control it with this [inaudible] David: Okay, so you wanna create a parallel circuit, where one part is controlling an LED but turning on with a button and the other one lights up a light, I'm assuming. Angela: Ohhhh, that's the other one. Oh. These are still with the switch, though. The one that slides to make the change. That's the one we	The participant (Angela) states that she wants to make her project a parallel circuit. The facilitator (David) wants clarification and asks her to articulate her thinking ("Let's hear it."). She explains that she has two circuits she wants to combine as one parallel circuit controlled by a button. David rephrases what he thinks she is trying to do back to her. She realizes from what he described as his understanding that she had misidentified what one of the circuits was.

	used last time.	
Articulation of algorithmic thinking (Breadboard exploration activity)	Juan: Alright, so your capacitor, how is your capacitor flow? Why don't you talk me through it. Jen: Hm? Juan: Talk me through it. Your battery starts here right? Jen: Starts to the battery, which goes to another battery, which goes to the switch, which goes to the [inaudible], which goes to the capacitor, which goes to the LED, the LED goes to the resistor ... Juan: But instead of the LED let's go to the capacitor, it goes and it's connected there. So what happens when we open it? Jen: It goes off. But it doesn't totally go off. Juan: So let's draw our capacitor. It comes after the switch but before the resistor? Is that right? No. It's between the LED and the resistor, as well, see that? So, your capacitor goes right here. And it … No. ... It's not coming off your LED, it's kind of connected in series … It's actually .. it's coming off like a little branch off of it. So before and after … yeah, there you go.	A participant (Jen) has used a new component, a capacitor, in her circuit. The facilitator (Juan) asks her to describe in detail how her circuit works ("talk me through it"). He prompts her with the start at the battery and then she begins to work her way through the circuit, component by component. She gets tripped up by the capacitor which is configured in a parallel circuit outside of the LED. Juan helps her draw the capacitor in a way that represents that and clarifies how it fits into her algorithm.
Articulation of algorithmic thinking (Arduino activity)	Sam: You need to start at the beginning. Denise: Okay, we're gonna turn off. Then we go back to high. And, high stays on. Ana: Right there it's blinking. Denise: But, then we'll return, doesn't that go high? Ana: It just turned off. Sam: Why do you think that is? Denise: Because, right here it turns the LED off, by making the, I don't know what thing that is, but ... the voltage hold. Sam: Ah-hah. So, the high and low is referring to what? Denise: On and off. Sam: Yeah. Exactly. The voltage that's going to it. The computer is handling all the details and figuring what voltage to send, right? So, if you think about what you guys were doing earlier, you had to connect different numbers with batteries, right? To change the voltage. Well, this is basically, like, there's little people in the computer doing that for you. And, they're sending how much voltage it needs to turn on. And then, taking it away, to turn it off. But, that's the nice thing about the code, is that it has this kind of stored variable, high and low, so that it makes it a lot simpler to just change it on the fly.	The facilitator (Sam) asks the participants (Denise and Ana) to begin talking about their program from the beginning, as they go through it. Denise describes the code that turns the light on and off, and Sam asks her what is happening in the moment she is describing in the walkthrough. He then makes the correlation between the exploratory work they were doing with the breadboard (changing the voltage with resistors) and the work they were doing with the Arduino (changing the voltage with code).

Examples of drawing, cognitive apprenticeship methods, CT skills and dispositions

CA method & CT skill and/or disposition	**Example**	**Notes on example**
Coaching abstraction and communication (Breadboard exploration activity)	Angela: How would I draw this over this? Like I have the connection. But they are like 2 series that are separated, and this is like the connection. I don't know how they are separated. Like, where would I put this? David: You don't have to take it apart. No. And then it moves to … with the LED, is this the positive side, then? Angela: This is … this is the positive side and this is the negative side. David: Okay. [crosstalk] Okay, um, so you condensed it a lot which is good, to be honest you didn't need the whole bread board. Um, but for simplicity purposes to draw it… um, like. . Even though it's one line it's not still parallel. Do you see why? So draw it as a parallel. 	The participant (Angela) is unclear on how to draw her circuit. The facilitator (David) commends her ability to abstract the information of a complicated circuit. She has removed the unnecessary information from the diagram. However, her abstraction demonstrates to him that she is unclear on how to present a circuit as parallel. She erased her initial drawing that presented the circuit as a series circuit and added on another branch after discussing it with David.
Articulating algorithmic thinking (Breadboard exploration activity)	 Jen: How do you draw this? Juan: how do you draw that? Yeah, let's follow those wires. Gimme a moment, alright. Let's clean this up. You got it? Or... Let's see. Jen: I put in the resistors, but I don't know where to go from there.	The participant (Jen) has a somewhat complicated circuit that involves a capacitor (which the participants had just been introduced to) and a push button. In representing the circuit as a drawing, Jen was confused as to how to order the components in her diagram. Talking through the circuit with her, Juan demonstrates that after the battery and the button, the LED comes first, then with an external connection to the capacitor. In this way, Juan is facilitating the understanding of the algorithm, the ordering of the steps of the circuit through the drawing.

	Juan: Doesn't it just go all the way back around? Oh, are you connected to an LED? Jen: Yeah, I'm connected to an LED Juan: Right there I think, right? Yeah, goes around there. Now you have to figure where the resistor goes, right? Jen: Ohhh Juan: Which is fine 'cause it's in parallel because you drew out one part of the circuit. Let's look at your … your image. And you've have a lot going on. Is this your ground? And this goes to your switch? It stays on and then comes off. The push button is cool, that little effect. Alright, so the ground goes directly to … is this power or ground? This is power. So the power goes to the switch, the switch ... I'm ignoring the capacitor right now, because that's almost like another battery. But it is on the other side of the switch, I'm just trying to make sure this is right. So actually, it goes the other way, because the power goes here. And first come the LEDs. And then ... Jen: So it should be the LEDs and then the capacitor. Juan: Correct, that's what you've got going on. But your resistor isn't going anywhere. So your resistor is actually … it's not doing anything. Follow it with … Ignore the capacitor, alright? So it goes to your LED. 	
Reflection on abstraction (Breadboard exploration activity)	Ezra: So yeah, somebody mentioned that they prefer to use their own symbols. Yeah, if they use their own symbols is that still an abstraction? Group: No, nope. Ezra: Actually it is, it is an abstraction. What's the problem with it, though? Group: No one can read it. Ezra: No one can read it. How can you make it that we're able to read it? Rafa: Add a key to my language.	When a participant says they would rather use their own symbols instead of the standard circuit schematic symbols to draw their circuit, the lead facilitator (Ezra) opens up the discussion to the entire group to reflect on why experts might use a standard version of drawing and what benefits there are to doing that.

	Ezra: You could add a key, yeah. But we want you also to become familiar with the internationally recognized symbols for these electrical parts.	

Examples of debugging, cognitive apprenticeship methods, CT skills and dispositions

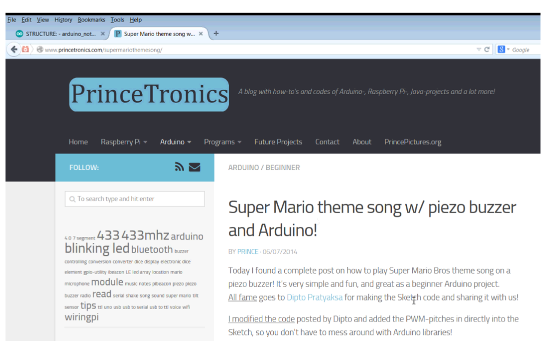
CA method & CT skill and/or disposition	**Example**	**Notes on example**
Modeling decomposition and persistence in dealing with difficult problems (Breadboard exploration activity)	 Ezra and David discussing Angela's circuit in front of her Ezra:[to Angela] It's a little weird. Because this shouldn't affect this. Once you have it parallel, it should close the circuit. It's kind of weird. Let's say you move this switch to another row, over here and then you took that leg of the LED and plugged it into the same row. Now, the LED is off. Yeah, this is totally weird. I have no idea how this is working. [puts his hands on the board] That is very strange. Well, there is something I don't know about it, so that's cool. But yeah this is the closest approximation I could make to that. I don't know why it doesn't work the other way. [to David] Do you feel like this is kind of weird that it's working like this? Because this switch is in parallel so it should be on its own. It shouldn't really be affecting anything. They're all in the same row, so I feel like the switch shouldn't be affecting them. David: It shouldn't. Ezra: And then when I move this over and then move the switch over to the same line, this just doesn't light up. Like it should light up if I move the LED into another row and move the switch into that row so that it's wired normally, it doesn't work. David: [To Angela] You defied logic, because	A participant (Angela) asked for help on the circuit she was working on. It was wired in parallel and a switch on a different line was affecting another area of the circuit. The facilitator she initially asked (Ezra) could not understand what was going on and brought in another facilitator (David) to take a look. In front of her, they discussed their confusion and the different avenues they tried to make sense of it. Eventually, by bringing in a third facilitator and discussing it with him, they realized that it had a short circuit.

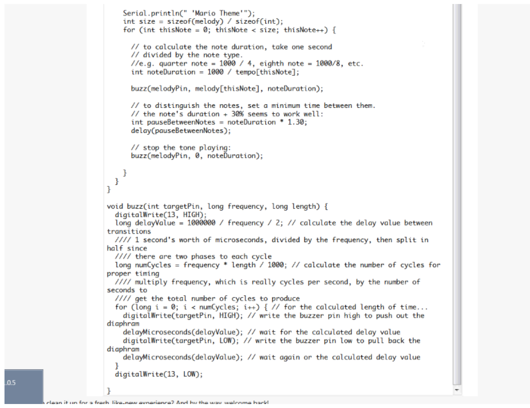
	we can't get this [laughter]. Ezra: Isn't that weird?	
Coaching decomposition (Breadboard exploration activity)	Sam: So let's follow the path that has the light. So we come here, again, we branch out here. So now, what's happening? Well, the only other thing in this row is this resistor. Becky: Mhhmm Sam: But that comes back to where we started right? So where is this circuit getting broken off? Where are the electrons hitting a wall and not being able to continue? Becky: Is it right here? Sam: Right here, yeah, in this stem of the light, right? 'Cause they get here, but then they have got nowhere to go. This is a dead end.	Here, the facilitator (Sam) can focus the participant (Becky) on the circuit she built and the path it follows. He provides some structure, asking her through observation to determine where there might be a problem, and then discusses with her why that might be a problem.
Coaching decomposition (Breadboard exploration activity)	Researcher: Why do they keep blowing out? Sam: Oop. Denise: And with this one doesn't happen. It's always the first one. Sam: They're blowing out as they are set up right now? Denise: Uh, yes, but it's always happening with the first one [inaudible] Sam: Uh huh. Ah, so that one gets blown out. Denise: Yeah. Sam: Okay, so where is your power coming in? Denise: From … Sam: Through here? Denise: Yeah. Sam: Okay. And where does it go from there? Denise: Well, it goes out this way and continues this way. Sam: So it's got 2 options. Right? So what kind of a circuit is that? Denise: Parallel. Steve: Parallel, okay. So, if it goes that way, it does other stuff. But if it comes through this way, what does it hit? What components does it see? Daniela: Well, the resistor. Steve: Okay. Daniela: And then, it goes through the light, the LED. Steve: Does it go through … Is your intention to have it to go through the resistor and the LED? Daniela: Well, at first I had it like this and then the light did turn on, but then it stopped working out of nowhere. Steve: Okay. So, when it's going through here, is it making it through? Is a single flow, a single stream of electricity going through both the resistor and the LED? Daniela: I think it's just going through the	A participant struggles with the same LED in her circuit blowing out repeatedly. The facilitator shows her that of the same problem is due to her circuit arrangement where all of the voltage is going through the LED instead of passing through a resistor.

	resistor. But … uh ... Steve: It's going through both, right? Daniela: Yeah. Steve: What you've set up here is a parallel circuit, right? It can go through the resistor one route, or it can go through the LED the other route. And what do we know about parallel routes and circuits in terms of the voltage that goes through each? Is the voltage the same or different for parallel? The voltage is the same, right? Because they both start at the same and end at the same. So, what's happening is, you're getting the same high voltage. One stream is just passing through the resistor and nothing is happening to it. But the other one, all of the voltage is going through the LED. Daniela: Okay. Steve: So if you want the resistor to decrease the power going to your LED, those would have to be in series. So you would want something like resistor, leg, resistor, leg. LED, leg, LED, leg. You know what I mean? In series. Because right now, they're parallel, they have a choice. They're basically bypassing the resistor, which is not what you want. Cause then all that voltage is going straight to the LED.	
Modeling decomposition and persistence in dealing with difficult problems (Arduino activity)	Ezra: Yeah. So yours is a little different than mine. Ah, it doesn't know what port you're in. Maybe that's the problem. Veronica: Would that be this, or ... Ezra: Wait. Let me see. Rafa: Maybe it's in the wrong USB port. Ezra: No. It's right. Rafa: Okay. Sorry. Ezra: It's okay. We're gonna figure this out. I don't know why it's ... it doesn't change. Oh, there we ... oh. That is just so weird, right? Oh wait, no. This is the original. Ha! Sorry. I was like, it's the same one. Let's see. Let's make that zero, and see if that changes anything. [crosstalk] Ezra: There's something wrong. It's definitely getting it because it's changing. There's something wrong here though, right? Rafa: I thought it worked for a second because it went off. I was like ... Ezra: What's this over here? There's something- Rafa: They're not in sync. Ezra: Juan, theirs is a different problem. Juan: You can just start from the beginning. Ezra: They have this error code that says that it's not in sync. And then even if they change that to zero, and it uploads it. This changes and it says I got it. See? But then it just goes back to its old	Through this exchange the facilitator (Ezra) models decomposition and persistence by systematically going through the various possible problems with the artifact, whether it be the wrong USB port, the part of the breadboard used, or the variables entered. Eventually he calls on another facilitator (Juan), who solved the problem by just turning it on an off again.

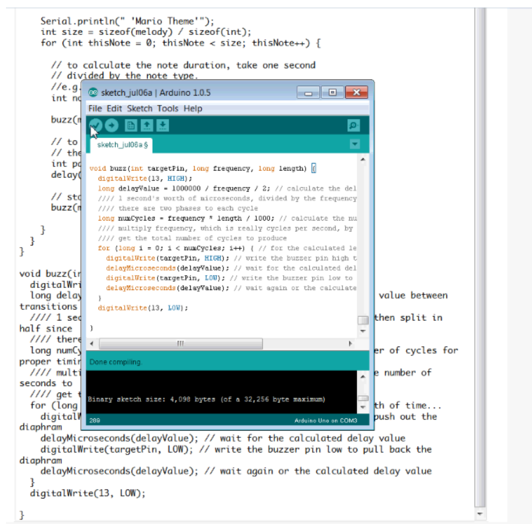
	code, even though this one says it should stay on. Ezra: [crosstalk] Isn't that weird? Juan: It's working now. All I did was unplug it and re plug it. Ezra: What. Juan: Yeah.	

Examples of participants exhibiting more purposeful tinkering on their own

Participant-led CT skills and dispositions	Example	Notes on example
Pattern recognition (Arduino activity)	Damaris: The pin the LED is attached to is 9 Magdalena: Just nine? Damaris: I guess so [she reads the instructions] Damaris and Magdalena create one [an Arduino program] that fades [the LED] in and out Magdalena: it follows the same pattern as the other one [the blinking activity], except it fades out Researcher: so what's different? Magdalena: the fadeout	Two participants (Damaris and Magdalea) want to extend beyond the initial program the participants were introduced to (which made the LED blink) and try a different program (which made the LED fade in and out). Following the pattern of the first program, but just changing a couple of lines of code, they manage to make it work without any facilitator assistance.
Decomposition and persistence in dealing with difficult problems (Arduino activity)	Elina: I know what we did wrong ... we placed it in between the parenthesis David: Wait, what? Elina: So instead of putting this after the parenthesis we put it in between the parenthesis David: Ohhhhhhhh Elina: We're deleting the parentheses David: There you go.	After 5 minutes of struggling with a piece of code that doesn't work, a participant (Elina) begins comparing the code line by line with an example piece of code finds the problem in syntax and explains to the facilitator (David) what went wrong.
Algorithmic thinking and confidence in dealing with complexity (Arduino activity)	 Rafa and Veronica find a site that has the Super Mario theme to play	A pair of participants (Rafa and Veronica) go online and find code to make a piezo buzzer that came with their Arduino kit play music with their Arduino. They go online, and then copy and paste it into their program, attempting to make it work alongside the code they already had in place to make an LED blink, They do not get it to work because they had not declared the variable for the LED and never call it appropriately, but they seem to grasp the concept of concurrent operations.

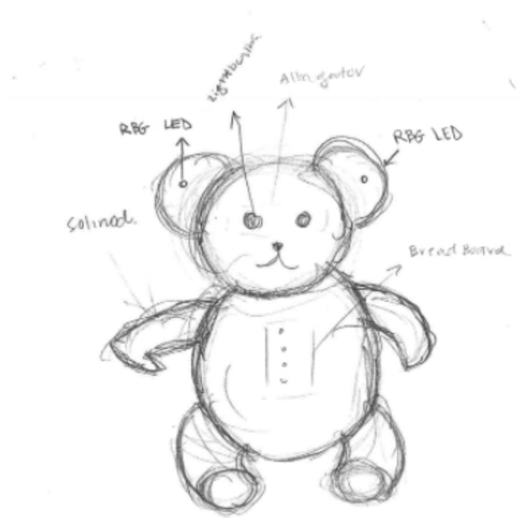
<table>
<tr>
<td></td>
<td>


Rafa and Veronica find the code they want and copy it

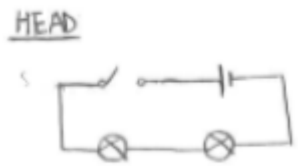


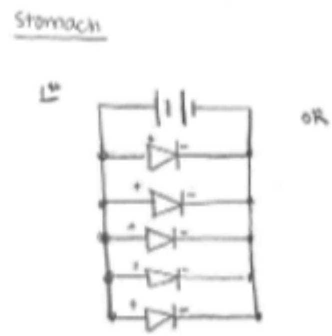
Rafa and Veronica paste the code into their program
</td>
<td></td>
</tr>
</table>

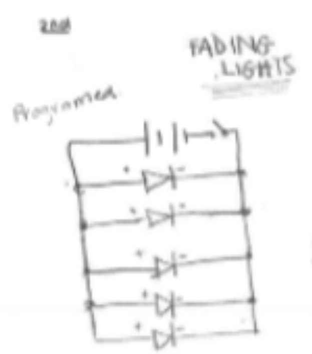
<table>
<tr>
<td></td>
<td>

Rafa and Veronica attempt to add the Blink program to the Super Mario program</td>
<td></td>
</tr>
<tr>
<td>Abstraction and tolerance for ambiguity (Final project design)</td>
<td>Denise and Ana drew sketches and schematics for their final project.


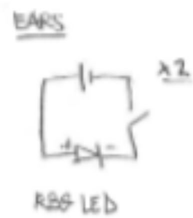
and Ana's drawing of bear as a whole</td>
<td>Two participants (Denise and Ana) use Ana's notebook to plan out their final project. They decide to make a teddy bear that lights up with different LEDs. They first draw out the bear and the different lights it will have, as well as a solenoid to make the arm move. They then drew schematics for each body part they were adding a circuit to. For the head, they drew a series circuit with a switch, battery, and two lightbulbs for the eyes. For the stomach, they drew a schematic consisting of a parallel circuit with 5 LEDs for buttons and a battery, as well as an alternative version of the stomach, consisting of a parallel circuit with a switch and 5 LEDs for buttons and a battery. Although they did not draw the symbol for it, the participants have written what looks like "potentiometer" and "FADING LIGHTS", indicating that they might want to design the bear with a potentiometer that allows them to brighten or fade the lights of the bear. For the ears, the participants drew a simple circuit with an RGB LED (LEDs</td>
</tr>
</table>

<table>
<tr>
<td></td>
<td>


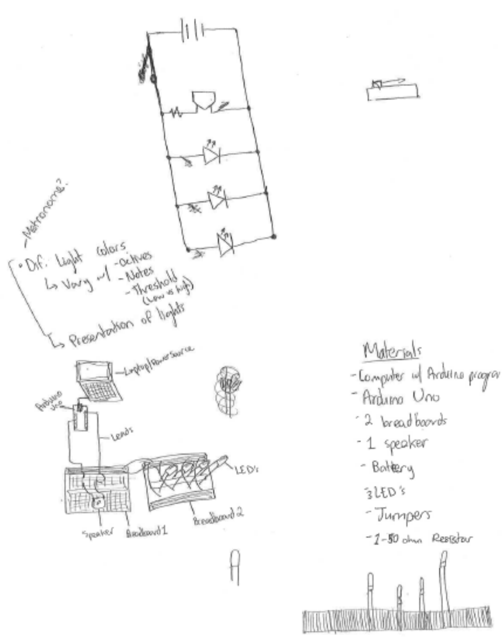
Schematic for head


Schematic for stomach of bear


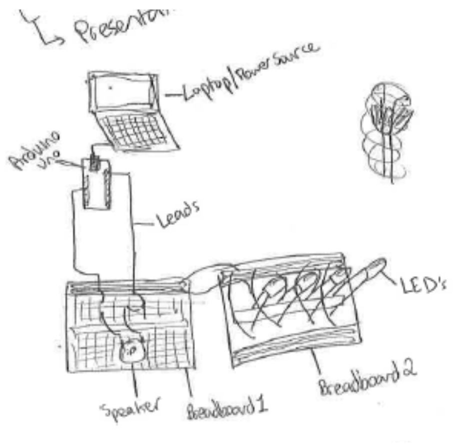
Alternative schematic for stomach
</td>
<td>
that change color), a battery, and a switch. For the arm of the bear, the participants included a solenoid, to make the bears arm move.
The abstractions help them determine the materials they will need and how they will power each part. By including include schematics with both the possibility of using a switch and not, seemingly comfortable with the ambiguity of how to complete the final product.
</td>
</tr>
</table>

<table>
<tr>
<td></td>
<td>

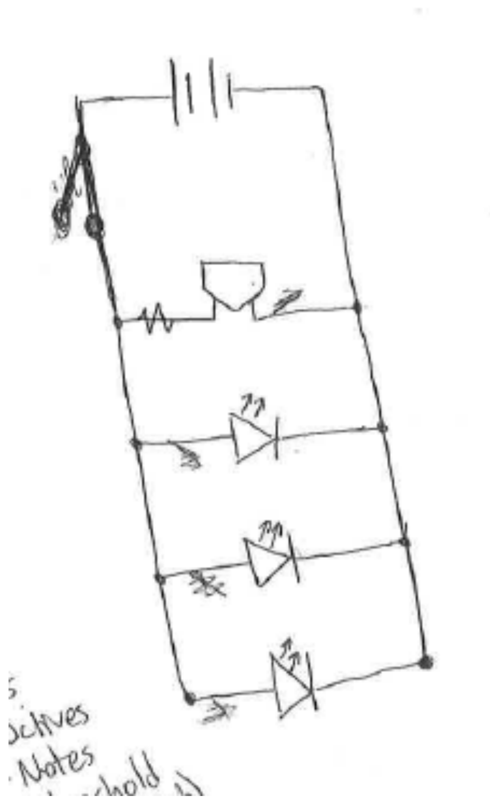
schematic for ears


Schematic of arm</td>
<td></td>
</tr>
<tr>
<td>Abstraction (Final project design)</td>
<td>

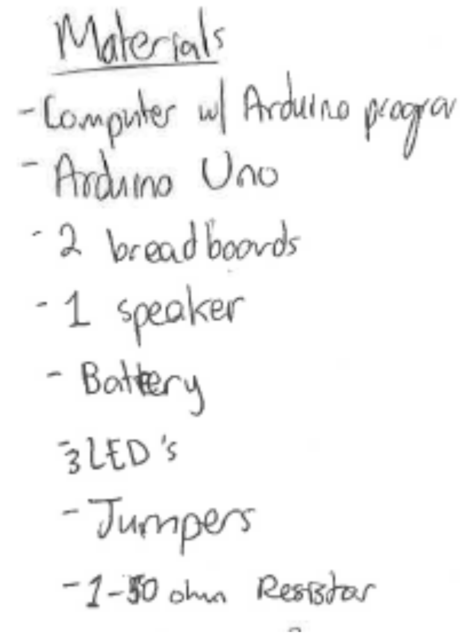
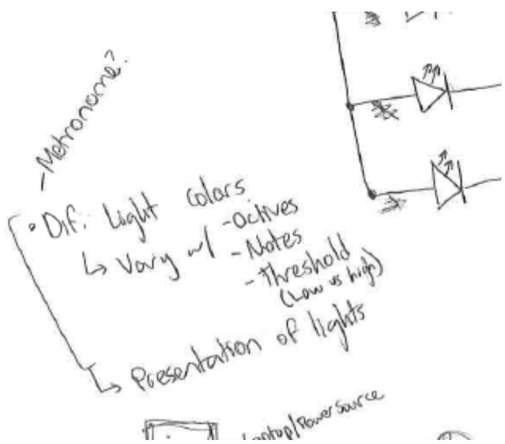
Veronica’s sketch for her final project</td>
<td>A participant (Veronica) draws a schematic and a sketch for her final project with her partner (Rafa). The schematic consists of a breadboard circuit with 3 LEDs and a speaker. Building off of what she did with Rafa in the Arduino exploration where they copied and pasted code for music off of the Internet, she designs a project that connects the breadboard with the lights and speaker to the Arduino and the computer, indicating the computer will serve as the power source. A switch is initially included in the design, but then she decides it is not necessary and scratches it out. She lists all the materials she will need, and writes out all of the ways the light can change with the music, including changes in octave, notes, and threshold. A side view drawing demonstrates her interest in having the LEDs at different heights, to provide a more textured visual experience. The abstracted drawings provide planning and clarification of how the project can be designed.</td>
</tr>
</table>

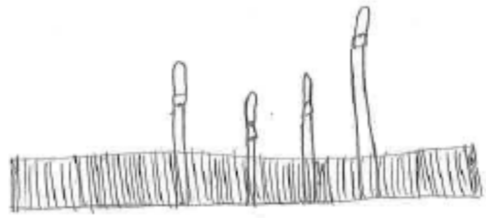


Map of how each element will connect



Veronica's schematic of her final project

	 Veronica's list of materials  Possible variations in light in relation to sound	

	 Side view of LEDs	

Appendix 12: Examples of Interactions Coded with a 1, 2, 3, or 4 for Effectiveness/Impact

Code	Example of coded selection	Notes
1 Tabletop Breadboard Activity	Juan: You guys had it a moment ago Rafa: What else do you want from me?	The participant just wants the facilitator to tell him what to do.
1 Breadboard Exploration Activity	Andre: Can someone explain what we're trying to make?! Ezra: You're making whatever you want to make.	The participant is uncomfortable with not having direction or a goal.
1 Arduino Activity	Juan: I added some stuff to your code and it might help you do what you were trying to do. It's really nice.	The facilitator did the work for the participant, missing out on a learning opportunity.
2 Tabletop Breadboard Activity	Ezra: I could also touch this one and I could touch this, here. I could touch this here. What am I though? Julio: A jumper Ezra: I'm a jumper? I feel like I should be some sort of component because otherwise it'd still be shorting the circuit, right? Julio: A switch? A light?	The facilitator provides some scaffolding, but the participant is still unsure about how to proceed
2 Breadboard Exploration Activity	Juan: Hi team, alright here are jumper wires, I'm going to slowly start taking them away the alligator clips because you don't really need them when you have a breadboard and these.	The facilitator uses materials to scaffold the use of breadboards, but offers little explanation as to how the transition should be made.
2 Arduino Activity	Ezra: Once you have this going, you're going to see your LED blinking, right? Now, I want you to try changing some of these variables and see what happens. Think, this is the number 13 and your positive is plugged into 13. What happens if you change this?	The facilitator indicates that he wants them to begin tinkering with the variables, but participants are unclear on which variables and what they should be doing with them.
3 Tabletop Breadboard Activity	Sam: What do you want your circuit to do? Ezra: Yeah what does this circuit do? Jackie: Turn on a light Angela: Turn on a light bulb.	Facilitators model evaluation for participants to consider

	Ezra: You have two light bulbs in parallel? Or one light bulb? Jackie: We have one that turns on here.	the appropriate design for their circuit.
3 Breadboard Exploration Activity	Juan: So the LED is connected on that side with the power, and it's also connected here, alright. And this part of the LED comes straight out back to the battery. You see that? Jen: What is the point of … is this even needed? Juan: well you could be using it, but you're not using it. How's that meme go? "This could be us, but you be playing." Right? That's what's going on here, because this is not even involved in that. So, if you remove this, it's like … like turning it off. It's like the capacitor is not there. So here it is again. Now let's follow where that is going. The end of your LED here comes out and it goes out here, it connects to the resistor, and connects there, and nothing comes out after that. So where should this one be connected to if you wanted to involve the resistor? Jen: The red one? Juan: Yes. So go ahead and remove the red one. And try elsewhere. You have two other possibilities here. Which end could it be? Give it a try. So now you've included the resistor. So LED first and then the resistor and then you have what you actually have. So now where … let's … let's figure out where	With coaching from the facilitator, the participant evaluates the design of her circuit. The facilitator uses pop culture to connect with the participant and clarify how the circuit could be designed differently.
3 Arduino Activity	Ezra: You know what you could do? Watch this. You could go on the internet and say what's a cool code, or what's a cool theme song or something? Veronica: Okay. I'm sorry. The first one I come to is Downton Abbey. Ezra: Which one? Veronica: Down ton Abbey theme song. Rafa: What's that? Veronica: It's a- Ezra: It's a TV show. Veronica: It's an English- Ezra: What's a cartoon? Something that's easily recognizable. Rafa: I don't know. Arthur? Veronica: Arthur. Ezra: Let's see.	The facilitator guides the participants in finding other codes online to modify, demonstrating the "Use" of Use-Modify-Create.

	Rafa: Artur? Ezra: Some people spell it like that. Sorry. I'm typing with one hand, so it's going real slow. Rafa: If I bring my laptop, then we might be able to use my laptop, where we make a mackey mackey … uh, Makey Makey ... and use it with my music making software. Ezra: Okay. We're just going to do Super Mario for now. All right? Let's see if the code is here. Sometimes you can take the code, and you can just ... [crosstalk 00:59:35]. Control C. Open up your sketch here, your blank sketch and then ... so did we. Now that's in there. You can verify it? Rafa: What? Ezra: So that's verified, but now you need to figure out where everything goes. So use your bread board. There's a Piezo buzzer in there. Veronica: A what buzzer? Ezra: A Piezo buzzer. That's that thing. So look at that code. Look at that code, and try to figure out how everything's wired up. It's going to be in the code, but use your breadboard, unplug all this, and try to plug all of that in there. Okay?	
4 Tabletop Breadboard Activity	Sam: Is there anything unnecessary in this circuit? Ezra: Yeah, can we get rid of jumpers? Will it work with less jumpers? Well, we have to connect things differently than right? Julio: She has the one... Ezra: Yeah, so what are you connecting to? Julio: I'm just connected to [inaudible] so you can get rid of me, I'm [inaudible].	The participant evaluates the circuit and recognizes how to make it more efficient.
4 Breadboard Exploration Activity	Sam: Alright, you're right it's parallel. So let's follow the path that has the light. So we come here, again, we branch out here. So now, what's happening? Well, the only other thing in this row is this resistor. Becky: Mhhmm Sam: But that comes back to where we started right? So where is this circuit getting broken off? Where are the electrons hitting a wall and not being able to continue? Becky: Is it right here? Sam: Right here, yeah, in this stem of the light, right? Cuz they get here, but then they have got nowhere to go. This is a dead end. Becky: Should I move this one over more? Sam: That might be helpful. Why would that help?	The participant understands how to debug a faulty circuit.

<table>
<tr><td></td><td>Becky: Because then electro … this would be going towards this way, not so much over here.
Sam: Okay.</td><td></td></tr>
<tr><td>4
Arduino Activity</td><td>
</td><td>Participants begin copying, combining, and modifying codes on their own, fully employing Use-Modify-Create</td></tr>
</table>

VITA

ROXANA HADAD, PHD

EDUCATION

2020	PhD in Educational Psychology, College of Education, University of Illinois Chicago Dissertation: The Use of Makerspaces for the Development of Computational Thinking Skills and Dispositions Advisor: Kimberly Lawless
2002	MPS in Interactive Telecommunications, Tisch School of Arts, New York University
1998	BA in English and Spanish (with High Honors), University of Illinois at Urbana-Champaign

PROFESSIONAL EXPERIENCE

2018-present	**Associate Director, Computer Science Equity Project, University of California, Los Angeles** Develop research and resources in order to scale up computer science curriculum and professional development programs and expand accessibility to quality computer science education to underrepresented youth in California and across the United States
2009 - 2018	**Director of Math, Science, and Technology, Center for College Access and Success, Northeastern Illinois University** Work with Chicago-area school administration, parents, teachers, and students, as well as community organizations and local government, to prioritize equity and improve STEM-focused academic and career outcomes for students
2002 - 2009	**Project and Instructional Designer, Collaboratory Project, Northwestern University** Worked with teachers from around the world to develop innovative learning units using a collaborative online platform centered around sharing and reflecting on student artifacts

RESEARCH INTERESTS

Equity in computer science education, computational thinking, making and makerspaces, culturally responsive pedagogy, informal education, educational technology

GRANTS

2015 - 2018	Principal Investigator, *Development of Assessment Protocols for Assessing Computational Thinking in Physics and Engineering Making Activities (PI: Roxana Hadad)*. National Science Foundation. Award Id: 1543124; Award Amount: $898,564; Duration: Oct 2015 - Dec 2018.

OTHER GRANT-RELATED WORK

2018-present	Associate Director, *Supporting Computing Access, Leadership, and Equity in California (SCALE-CA) (PI: Jane Margolis)*. National Science Foundation. Award Id: 1837780. Award Amount: $2,000,000; Duration: October 1, 2018-September 30, 2022 (estimated).
2016 - 2017	Advisor, *MakeCS: Activating Computer Science Learning Through Digital Making in Chicago Public School High Schools (PI: Rabiah Mayas)*. Hive Chicago via Chicago Community Trust. Award Amount: $15,000; Duration: July 2016 - June 2017.
2013 - 2016	Senior Personnel, *GEAR UP Chicago (PI: Wendy Stack)*. U.S. Department of Education. Award Id: P334A140132. Award Amount: $34,699,200; Duration: September 2013 – August 2019.
2010 - 2016	Senior Personnel, *GEAR UP Chicago (PI: Wendy Stack)*. U.S. Department of Education. Award Id: P334A110082. Award Amount: $49,190,400; Duration: September 2010 – August 2016.
2009 - 2015	Senior Personnel, *GEAR UP Chicago (PI: Wendy Stack*). U.S. Department of Education. Award Id: P334A100031. Award Amount: $21,508,800; Duration: September 2009 – August 2015.
2003-2006	Project and Instructional Designer, *STI: The Strategic Technology Astronomy Research Team (START) Collaboratory: Broadening Participation in Authentic Astronomy Research (PI: Morteza Rahimi)*. National Science Foundation. Award Id: 0334168. Award Amount: $607,500.00; Duration: September 2003 - August 2006.

REPORTS

Flapan, J., **Hadad, R.** Shorall, C., Twarek, B. (2019). *Computer Science Equity Guide*. http://access-ca.org/csforca/guide/

JOURNAL PUBLICATIONS

Tang, X., Yin, Y., Lin, Q., **Hadad, R.**, & Zhai, X. (2020). Assessing computational thinking: A systematic review of empirical studies. *Computers & Education*, 103798.

Hadad, R., Kachovska, M., Thomas, K., Yin, Y. (2019). Practicing formative assessment for computational thinking in making environments. *Journal of Science Education and Technology* 1-12. https://doi.org/10.1007/s10956-019-09796-6

Yin, Y., **Hadad, R.**, Tang, X., Lin, Q. (2019). Improving and Assessing Computational Thinking in Maker Activities: the Integration with Physics and Engineering Learning. *Journal of Science Education and Technology*, 1-26.

Hadad, R. (2005). "Northwestern University's 'Collaboratory Project' and 'Lexitown.'" *Modern Language Journal 89*, 3, 462-463.

Hadad, R. (2005). "Teaching with Technology—Lexitown: An Illinois Online Foreign Language Learning Initiative." *Learning Languages*, *10*(2), 33-34.

BOOK CHAPTERS

Hadad, R. & Lawless, K. (2014). "Assessing Computational Thinking". *Encyclopedia of Information Science and Technology*. Hershey, PA: IGI Global.

Hadad, R. (2013). "Using Game Design as a Means to Make Computer Science Accessible to Adolescents". *Cases on Digital Game-Based Learning: Methods, Models and Strategies*. Hershey, PA: IGI Global.

CONFERENCE PRESENTATIONS

Flapan, J., Ryoo, J., **Hadad, R.** (2020). *Building Systemic Capacity to Scale and Sustain Equity in Computer Science: A comprehensive model of professional learning for teachers, counselors, and administrators*. Paper presented at the annual Research in Equity and Sustained Participation in Engineering, Computing, and Technology (RESPECT) conference in Portland, WA.

Flapan, J., **Hadad, R.**, Ryoo, J., Margolis, J. Amalong, J. (2019). *Supporting Computing Access, Leadership, and Equity in California (SCALE-CA)*. Poster presented at the Special Interest Group for Computer Science Education, Minneapolis, MN.

Hadad, R., Kachovska, M., Thomas, K., Yin, Y. (2019). *Practicing formative assessment for computational thinking in making environments*. Poster presented at the Advancing the Integration of Interdisciplinary Computational Thinking in the Physical and Life Sciences Conference, College Park, MD.

Hadad, R., Kachovska, M., Thomas, K., Yin, Y. (2019). *Practicing formative assessment for computational thinking in making environments*. Paper presented at the 2019 annual meeting of the American Educational Research Association (AERA), Toronto, Canada.

Lin., Q., Yin, Y., Tang, X., **Hadad, R.**, (2018). *Assessing Circuit-based Artifacts in Maker Activities*. Poster Presented at American Psychological Association Annual Convention, San Francisco, CA.

Lin, Q., Yin, Y., Tang, X., & **Hadad, R.** (2018, April). *A Systematic review of empirical research on maker activity assessments*. Paper presented at the 2018 annual meeting of the American Educational Research Association (AERA), New York, NY.

Khaleghi, S., Yin, Y., & **Hadad, R.** (2018, April). *Improving computational thinking with Arduino activities*. Poster presented at the 2018 DRK12 NSF PI Meeting, Washington, DC.

Tang, X., Yin, Y., Lin, Q., & **Hadad, R.** (2018, April). *Assessing computational thinking: A systematic review of the literature*. Poster presented at the 2018 annual meeting of the American Educational Research Association (AERA), New York, NY.

Tang, X., Yin, Y., Lin, Q., & **Hadad, R.** (2018, April). *Making computational thinking evident: A validation study of a computational thinking test.* Paper presented at the 2018 annual meeting of the American Educational Research Association (AERA), New York, NY.

Khaleghi, S., Yin, Y., & **Hadad, R.** (2018, April). *Improving computational thinking with Arduino activities.* Poster presented at the 2018 annual meeting of the American Educational Research Association (AERA), New York, NY.

Hadad, R., Hausman Jacobson, C. M., Thomas, K., Solórzano, G., Kachovska, M., and Yin, Y. (2018, February). *Using cultural responsiveness to elicit computational thinking in maker environments.* Poster presented at the Special Interest Group on Computer Science Education (SIGCSE) Conference, Baltimore, MD.

Lin, Q., Yin, Y., Tang, X., & **Hadad, R.** (2018, February). *A Systematic review of empirical research on maker activity assessments.* Poster presented at the University of Illinois College of Education Research Day, Chicago, IL.

Tang, X., Yin, Y., Lin, Q., & **Hadad, R.** (2018, February). *Assessing computational thinking: A systematic review of the literature.* Poster presented at the University of Illinois College of Education Research Day, Chicago, IL.

McBride, D., Kachovska, M., & **Hadad, R.** (2017, September). *Instrumentation for Equity: The Development of a Culturally Responsive Assessment Checklist.* Fourth International Culturally Responsive Evaluation and Assessment (CREA) Conference, Chicago, IL.

Tang, X., Yin, Y., **Hadad, R.**, Lin., Q. (2017, August). *Assessing Computational Thinking: a test with a combination of think-aloud and written prompts.* Poster presented at the American Psychological Association Annual Convention, Washington, D.C.

Yin, Y., **Hadad, R.**, Tang, X., Lin, Q., Hausman, C. M. (2017, April*). Improving Computational Thinking Skills and Physics Engineering Learning by Using Makerspace Activities and Formative Assessments.* Paper presented at the National Association for Research in Science Teaching Annual International Conference, San Antonio, TX.

Yin, Y., **Hadad, R.**, Tang, X., Lin, Q., Hausman, C. M. (2017, January). *Improving Computational Thinking Skills and Physics Engineering Learning: Year 1 and the Impact on Students.* Poster presented at the University of Illinois College of Education Research Day, Chicago, IL

Yin, Y., **Hadad, R.**, Tang, X., Lin, Q., Hausman, C. M. (2017, January). *Improving Computational Thinking Skills and Physics Engineering Learning: Year 1 and the Impact on Students.* Poster presented at the University of Illinois College of Education Research Day, Chicago, IL

Hadad, R., Yin, Y. Ramirez, E., Sweeton, J., Hausman, C.M., Tang, X., Lin, Q, Bonomo, S. Solorzano, G., Thomas, K., McBride, D. (2017, January). *Physics Engineering Learning and Computational Thinking in Maker Activities.* University of Illinois College of Education Research Day, Chicago, IL

Hadad, R. (2013, November). *Assessing Computational Thinking.* Northeastern Illinois University Faculty Research Symposium, Chicago, IL.

Stack, W. & **Hadad, R.** (2013, February). *Optimizing Community Partners to Increase STEM Initiatives.* National Council for Community and Education Partnerships (NCCEP), Las Vegas, NV.

Hadad, R. (2012, July). *Adobe Generation: An Online Creative Curriculum.* Adobe Education Leader Summer Institute, San Jose, CA.

Hadad, R. (2012, November). *Google Apps for your Organization.* Mid-America Association of Equal Opportunity Program Personnel, Geneva, IL.

Hadad, R. (2012, February). *Increasing STEM Knowledge Using Real-World Practitioners in Blended Communities.* Illinois Computing Educators Conference, St. Charles, IL.

Hadad, R., Hodgson, G. & Usher, I. (2010, November). *Games Design Workshop: Using a Blended Model to Teach Digital Media.* Virtual School Symposium, Phoenix, AZ.

Hadad, R. (2007, March). *Improving Extended Response Using the Collaboratory's ePortfolio.* Illinois Technology Conference for Educators, St. Charles, IL.

Hadad, R. (2007, March). *Celebrating National Poetry Month with Our Classmates, Our School And The World.* Illinois Technology Conference for Educators, St. Charles, IL.

Hadad, R. (2007, February). *Online Student Learning Portfolios.* Illinois Statewide No Child Left Behind Conference, Chicago, Illinois.

Hadad, R. (2007, February). *Connecting Classrooms Online Using Collaboratory Sponsored Projects.* Illinois Online Conference for Teaching and Learning. http://www.ilonlineconf.org/.

Hadad, R. & Finkel, D. (2006, July). *From Chicago to Durango: An Exchange Of Technology And Culture.* National Education Computing Conference (NECC), San Diego, California.

Hadad, R. (2006, November). *Creating an Online Community for Students Through Collaborative Sponsored Projects.* Illinois Education and Technology Conference, Springfield, Illinois.

Hadad, R. (2006, November). *Collecting Oral Histories On the Collaboratory.* Illinois Education and Technology Conference, Springfield, Illinois.

Piagentini, S. & **Hadad, R.** (2005, November). *Creating An Online Music Skills Test.* Association for Technology in Music Instruction, Quebec City, Quebec.

West, M. & **Hadad, R.** (2005, November). *MindRap.* Hip Hop Education Summit, Bronx, New York.

Hadad, R. (2005, November). *Creating an Online Community For Students Through Collaborative Sponsored Projects.* Illinois Education and Technology Conference, Springfield, Illinois.

Hadad, R. (2004, November). *Lexitown: Learning K-8 Foreign Language Through Project-Based Collaboration.* American Council on the Teaching of Foreign Language (ACTFL), Chicago, Illinois.

Hadad, R. (2004, September). *Lexitown: Learning K-8 Foreign Language Through Project-Based Collaboration.* International Online Conference on Second and Foreign Language Teaching and Research.

Hadad, R. & Greenberg, G. (2004, June). *Lexitown: Learning K-8 Foreign Language Through Project-Based Collaboration.* Computer Assisted Language Instruction Consortium (CALICO), Pittsburgh, Pennsylvania.

Hadad, R. (2004, January). *Creating an ePortfolio.* No Child Left Behind (NCLB) Learning Teams, Gurnee, Illinois.

Hadad, R. (2003, June). *Lexitown: Learning K-8 Foreign Language Through Project-Based Collaboration.* National Education Computing Conference (NECC), Seattle, Washington.

INVITED TALKS

Hadad, R. (2018, October). *Computational thinking, making, and cultural responsiveness.* Invited talk at University of California Los Angeles, CRESSTCon 2018.

Hadad, R. (2018, March). *Computational thinking, making, and cultural responsiveness.* Webinar presented for MSPnet Academy Discussion. http://hub.mspnet.org/index.cfm/33402/

Yin, Y., **Hadad, R.**, Tang, X., Lin, Q., & Hausman, C. M. (2016, December). *Assessing Computational Thinking in Makerspace Activities.* Invited talk at Beihang University, Beijing, China.

Hadad, R. Makerspaces: (2017, April). *What they are and how they can impact your educational organization.* Webinar presented for Adobe Education. https://adobemakerspaces.attendease.com/

Hadad, R. (2009, October). *No fear in art.* Keynote at International Student Media Festival. Louisville, KY.

COURSES TAUGHT

2007-2013 **Online**
Adobe / Edge Gain
Game Design and Development Curriculum Designer and Instructor
Developed Game Design and Development curriculum and taught over 500 students in massive open online course (MOOC)

2012-2013 **Face-to-Face**
Chicago Teachers' Center at Northeastern Illinois University
Developing 21st Century Skills Via Technology-Infused, Inquiry-Based Learning
Developed curriculum on using technology tools with inquiry-based education and provided professional development to in-service and pre-service teachers

CURRICULUM DEVELOPMENT

2011-2013 **Pearson Foundation Learning Labs / Partnership for 21st Century Skills**
Developed Common-Core aligned math, science and technology content for Pearson and Partnership for 21st Century Skills

2007-2009 **Tiz Media Foundation**
Managed, designed, and developed multimedia that used hip-hop to teach STEM concepts

2005-2007 **Youth Technology Corps**
In partnership with the Mexican government, developed online interactive educational communities for Mexican and American students and teachers learning computer hardware repair

SERVICE TO UNIVERSITY AND COMMUNITY

2019 Participant, Advancing the Integration of Interdisciplinary Computational Thinking in the Physical and Life Sciences, College Park, MD (NSF sponsored)

2018-2019 Committee Member, National Science Foundation STEM + C Planning

2017 Participant, Computational Thinking Integration Summit, Boston, MA (NSF sponsored)

2014-2017 Member, Computer Science Teachers Association's Computational Thinking Task Force

2015	Member, Computational Making and Design Pathway, City of Chicago Learning Initiative
2010-2011	Representative, University of Illinois at Chicago Graduate Student Council
2009-2013	Board Member, Adobe Education
2008-2009	Board Member, Adler Planetarium Education Committee

Referee/Reviewer: *American Educational Research Association (AERA), Fablearn, Journal of Science Education and Technology, ACM Transactions on Computing Education*

PROFESSIONAL MEMBERSHIPS

American Educational Research Association

Association for Computing Machinery

COURSES TAKEN

Research Methods	Quantitative Inquiry in Education
	Data & Interpretation in Educational Inquiry
	Advanced Analysis of Variance in Educational Research
	Research Design
Educational Psychology	Introduction to Learning Sciences
	Cognition and Instruction
	Attitudes & Social Cognition
Educational Technology	Assistive Technology & Social Applications
	Applications for Interactive Telecommunications Systems
	Communications Lab
	Introduction to Computational Media
	Physical Computing
	Future of the Infrastructure
	Game Design
	Digital Sound Workshop
	Interactive Telecommunications Technology
	Information Contours
	Multimedia Workshop
	Storytelling
	Online Interactivity
Others	Philosophy of Education Research
	Community Research

SELECTED HONORS/AWARDS

2020	Recipient of the "Best of RESPECT" award at the Research in Equity and Sustained Participation in Engineering, Computing, and Technology (RESPECT) conference in Portland, OR
2017	UIC Healthy City Collaborative (HCC) Featured Researcher
2010 - 2013	UIC Board of Trustees Award
2000 - 2002	NYU Graduate and Professional Opportunity Fellow
2000	National Hispanic Foundation for the Arts Fellow
1998	UIUC Chancellor's Scholar. Based on outstanding academic achievement.
1998	UIUC James Scholar. Based on outstanding academic achievement.

SKILLS

Fluent in Spanish with a bicultural background

SPSS, ATLAS.ti, Dedoose, Blackboard, Moodle, Google Apps, Adobe Creative Cloud Suite