Second Grade Student Engagement in Computational Thinking

Item Type	Dissertation (Open Access)
Authors	Pektas, Emrah
DOI	10.7275/54730
Download date	2026-03-08 11:26:12
Link to Item	https://hdl.handle.net/20.500.14394/54730

Second Grade Student Engagement in Computational Thinking

A Dissertation Presented

by

EMRAH PEKTAS

Submitted to the Graduate School of the University of Massachusetts Amherst in partial fulfillment of the requirements for the degree of

DOCTOR OF EDUCATION

May 2024

College of Education

Second Grade Student Engagement in Computational Thinking

A Dissertation Presented

by

EMRAH PEKTAS

Approved as to style and content by:

Darrell Earnest, Chair

Enrique Suárez, Member

W. Richards Adrion, Member

Shane Hammond, Associate Dean for Students Success
College of Education

ACKNOWLEDGEMENTS

I express my deepest gratitude to my advisor, Professor Darrell Earnest, for his unwavering guidance and support throughout my dissertation journey. Without hesitation, I credit his commitment to regular meetings during my most challenging times and his constructive feedback as pivotal in bringing this dissertation to fruition. Professor Earnest's exceptional mentorship and dedication played a crucial role in shaping the success of my dissertation.

I also extend my deepest appreciation to the late Florence Sullivan, my original advisor, whose initial guidance and support laid the foundation for my academic journey. Professor Sullivan significantly influenced my worldview, particularly in understanding social justice, gender, power, and privilege issues. I consider myself fortunate to have been under her guidance not only as an advisee but also as a student during both my master's and PhD studies. Although her untimely passing left a void during a critical phase, it was then that Professor Earnest emerged as a beacon of unwavering support, seamlessly continuing the legacy of mentorship and guidance left by Professor Sullivan.

I also wish to express my gratitude to my other committee members, Enrique Suárez and Rick Adrion, for their invaluable support and feedback throughout this project. Additionally, I am deeply appreciative of Professor Adrion's financial support, which alleviated any concerns about financial matters throughout most of my PhD years.

I deeply appreciate the friends and colleagues who supported me throughout the different stages of my dissertation, offering their emotional support. Ali Soken, Fatih Cetin, and Ozkan Yildiz, in particular, have been unwavering in their support, standing by me through the triumphs and challenges of my journey.

I extend my heartfelt appreciation to the Education Ministry of Turkey for their generous sponsorship of my master’s degree and Ph.D. studies in the United States.

Last but not least, I extend my heartfelt gratitude to my life partner and beloved, Imran. Without her unwavering support, infinite patience, and flexibility, this journey would not have been possible. My deepest thanks go to her. I am also grateful to our children, Meryem and Kerem, whose boundless joy has been a constant source of motivation throughout my studies.

ABSTRACT

SECOND GRADE STUDENT ENGAGEMENT IN COMPUTATIONAL THINKING

MAY 2024

EMRAH PEKTAS, B.ED. INONU UNIVERSITY

M.ED. UNIVERSITY OF MASSACHUSETTS AMHERST

PH.D. UNIVERSITY OF MASSACHUSETTS AMHERST

Directed by: Professor Darrell Earnest

The purpose of this dissertation study was to explore what and how second graders learn in a CS/CT integrated science and engineering module through their research, construction of physical models such as house models, simulation of house models, and Scratch projects. This is a qualitative research and the data was collected in person through artifact interviews, field notes, and classroom artifacts from three different 2nd-grade classrooms totaling 28 students across two schools in an urban area of western Massachusetts resulting in 30 interviews. Based on Brennan and Resnick’s (2012) CT framework, findings reveal that participants developed the following CT practices while planning, building, testing, and proposing a revision to their house models: abstraction and modularization, being incremental and iterative, testing, debugging, and data collection. Second graders also engaged in CT concepts and CT practices in their Scratch programs: The CT concepts of sequences, events, conditional thinking, data, and the CT practices of being incremental and iterative, testing and debugging, and abstraction and modularization. These findings are consistent with the existing literature. The implications of this dissertation study are as follows. First, this study identified data collection as a novel computational thinking practice, not explicitly addressed in Brennan and Resnick's (2012) framework, yet demonstrated as significant within the context of this research. Second, second graders participated in building a tangible house model, categorized as an unplugged computational thinking (CT) exercise in this dissertation. This hands-on task integrated CT practices from Brennan and Resnick's 2012 framework, suggesting that CT practices extend beyond digital or coding realms to encompass tangible, unplugged activities. Last, the sequential progression from hands-on house model construction to abstract Scratch programming proved pivotal in enriching the learning journey for the second graders in this study. The significance of the present dissertation study comes from the fact that it aims to address the gap in research regarding computational thinking education for lower elementary students, particularly second grade students.

TABLE OF CONTENTS

List of Tables

List of Figures

CHAPTER I

INTRODUCTION

There is a growing consensus (Wing, 2006; Grover & Pea, 2013) that computational thinking (CT) is a basic life skill that every child should have in this computing era. CT refers to "solving problems, designing systems, and understanding human behavior, by drawing on the concepts fundamental to computer science" (Wing, 2006, p. 33). Computational thinking is still in the process of being defined and to date there is not an established definition of CT (e.g., Barr & Stephenson, 2011; Shute et al., 2017). When discussing the role of computational thinking, Wing (2006) argues that CT is a fundamental skill that not only computer scientists but anyone needs to achieve. She argues that this basic skill should be added to every child's analytical ability in addition to reading, writing, and arithmetic. Grover & Pea (2013) believe that those with computational competencies will have a better advantage of being positioned in a world where computing is common. Yet, how is it that young students develop such competencies? This dissertation addresses such a question through a study on second graders' engagement in computational thinking through a CT integrated science and engineering module.

Instruction in the elementary grades focusing on CT has encountered some hurdles. Many elementary schools in the US and other countries lack specialized CS teachers or a constrained schedule. When CS/CT is presented as a stand-alone topic, there are two options: adding CS/CT as another core subject like ELA, Math, Science or as a special subject like music, art, languages. The lack of specialized CS teachers and corresponding likelihood that young students could not engage in CT could potentially deprive students of essential life skills. An alternative approach to the stand-alone options for students to have such a fundamental competency is integrating CS/CT into other subjects that elementary teachers already teach. In addition to providing an option absent of specialized CS teachers, such integration allows students to engage with and understand concepts from multiple disciplines. Through CT integration at elementary level, CT can be accessible and motivating to a broader population (Mannila et al., 2014; Gautam et al. 2017, Wang et al, 2021). This dissertation explores 2nd graders' CT understanding in a CT integrated science and engineering module.

In the CT-integrated science and engineering module, second graders participated in two primary activities: first, building house models to withstand natural disasters, an unplugged CT activity, and subsequently, developing a program in Scratch, a plugged activity. This progression from a hands-on, tangible task to a more digital, abstract one offers a vital insight into how early grade students comprehend and apply computational thinking. Such a structured approach in instruction is essential for exploring and understanding the development of CT skills among younger learners.

While there's a growing emphasis on CS/CT education in content areas, there remains a noticeable gap in research when it comes to lower elementary students. The present dissertation study aims to address this gap in the literature. Systematic reviews examining the integration of Computational Thinking (CT) in STEM fields have been conducted, revealing a noticeable gap at the elementary level (most were with either middle school or upper elementary level students), particularly in the lower grades such as Kindergarten through second grade (Wang et al., 2021; Ogegbo et al., 2021). Yet, few studies target this specific demographic. Therefore, there is a need for empirical studies on student learning/understanding in CS/CT integration with STEM fields such as science and engineering in the context of the lower elementary school level such as K-2. Recognizing this lack of research, this dissertation focuses on the understanding of 2nd-grade students within a CS/CT integrated science and engineering module. Also, in their case study, Yang, Baek, Chittoori, and Stewart (2019) clearly state the limited amount of literature on CT practices in the K-12 engineering design challenge. Students need to have exposure to CT at early ages as mentioned by many researchers (e.g., Wing 2006). So, this dissertation study will be on students' CT understanding in a second grade CT-integrated science and engineering curriculum. So, it aims to address the gap(s) in the literature.

The CSforAll Movement [https://www.csforall.org] is one of the current initiatives to address the challenge of implementing CT in content areas in American schools. The CSforAll Movement is defined as a national initiative in the USA aiming to provide every student access to computer science education. It aims to reach all children in the USA to teach this essential life skill. There have been many initiatives to serve this purpose. This dissertation reports on one such initiative in Western Massachusetts in a mid-size, ethnically diverse city in the New England area of the United States. The CSforAll RPP project funded by the National Science Foundation (NSF) [https://www.nsf.gov/awardsearch/showAward?AWD_ID=1837086; https://www.nsf.gov/awardsearch/showAward?AWD_ID=2219452] is one where the district elementary school teachers are introducing computational thinking across K-5 by integrating CS/CT into other standards-based subject areas. Teachers co-developed CS/CT integrated curricula, and students are learning CS/CT concepts through the engagement of the subject area integrated curricula. The need for CT integration into science and engineering mentioned in the second paragraph is included in the goals of the CSforAll RPP project, and the present dissertation is a part of this effort.

I adopt B&R's CT framework in this dissertation to examine 2nd graders' CT understanding in a CT integrated science and engineering module. Although there are multiple frameworks in the literature for studying computational thinking (Hynes et al., 2016; Cateté et al., 2018; Yang et al., 2018, and Cabrera et al., 2023), Brennan and Resnick's (2012) CT framework for studying and assessing the development of computational thinking. The current dissertation embraces the CT definition proposed by Brennan and Resnick because their extensively utilized framework, capturing essential elements of the domain, provides a vital approach for investigating young children's engagement in computational thinking. They define CT with three dimensions: Computational concepts (the concepts designers engage with as they program, such as conditionals), computational practices (the practices designers develop as they engage with the concepts), and computational perspectives (the perspectives designers form about the world around them and about themselves).

The purpose of this dissertation study is to explore what and how second graders learn in a CS/CT integrated science and engineering module through their research, construction of physical models such as house models, simulation of house models, and Scratch projects. Based on the framework, the purpose of the study, and the gaps in the literature, research questions include:

- RQ1: What CT practices did second graders develop through the engagement with the CT integrated science and engineering module while planning, building, testing, and proposing a revision to their house models, and how did they emerge?
    - RQ1a: Which CT practices did 2nd graders engage in describing their understanding of modeling and simulation?
    - RQ1b: What CT practices did 2nd graders develop in planning their house models?
    - RQ1c: To what extent did participants’ initial house models survive from the simulation/test?

    - RQ1d: Which CT practices did second graders engage in describing and understanding their material choices they made in creating their house models and proposing a revision?
- RQ2: How do they describe in Scratch the process they undertook?

In the following chapter, I will discuss the literature review within the context of the present study: what computational thinking definition is adopted in the present study and what studies are conducted on student CT learning or CT and science learning together in the CT-integrated science and engineering literature in K-12. After this chapter, the methods chapter will be discussed.

CHAPTER II
RELATED WORK

In the following section, I will provide the CT definition that I adopt in the present dissertation study and why I adopt that particular CT definition. After that section, I will discuss the studies on computational thinking integration into science and/or engineering with two approaches: Approach one is that CT is the end while science and engineering is the means. Approach two is treating both CT and science and engineering as equally important goals.

Adopted Definition of Computational Thinking

Computational thinking is still in the process of being defined so there is not an uniformly accepted definition of CT (e.g., Kalelioglu et al., 2016; Yadav et al. 2017). The components in researchers' CT definitions differ from each other. The CT definition adopted in this dissertation study is below.

One may think of computational thinking with respect to particular facts to learn, how one engages in CT in activity, or alternatively the utility of CT more broadly. For this reason, Brennan and Resnick (2012) defined CT in their framework with three dimensions: CT concepts, CT practices, and CT perspectives. In the present study, I adopt Brennan and Resnick's (2012) CT framework for students' CT learning. The reason why the present dissertation adopts their CT definition is that their widely used framework, which captures key aspects of the field, offers a critical method for examining young children's involvement in computational thinking. It also aligns with the MA DLCS standards (2016) which in effect is a framework for teaching the content and practices. These two above-mentioned reasons show that Brennan and Resnick's (2012) CT framework is the appropriate framework for this dissertation study. Table 1 below is the three dimensions and the components of each dimension of the framework. I define each of these dimensions below.

Table 1: Brennan and Resnick (2012) CT Framework

Brennan and Resnick (2012) CT Framework	**CT Concepts**	● **Sequences** ● **Events** ● **Conditionals** ● **Data** ● Loops ● Parallelism ● Operators
	CT Practices	● **Being incremental and iterative** ● **Testing and debugging** ● **Abstraction and modularization** ● Reusing and remixing
	CT Perspectives	● Expressing ● Connecting ● Questioning

Brennan and Resnick define CT concepts as “the concepts designers engage with as they program” (p. 1). In their study, they focus on how these concepts are utilized in Scratch, a block-based programming language popular among young learners. They identify seven key concepts that are not only essential in Scratch projects but also applicable in other programming and non-programming contexts. This approach helps learners grasp fundamental programming principles that have wide-ranging applications. Secondly, computational practices are defined as “the practices designers develop as they engage with the concepts” (p. 1). This involves the methods and strategies that designers adopt, such as testing and debugging, reusing and remixing, or abstracting and modularizing their work, as they apply these CT concepts in real-world scenarios. CT practices like CT concepts are applicable to various other programming or plugged environments as well as contexts outside of programming such as unplugged activities. Lastly, Brennan and Resnick introduce computational perspectives, which they describe as “the perspectives designers form about the world around them and about themselves” (p. 1). This dimension reflects how designers perceive and interact with the broader societal implications of computing, recognizing both its impact and potential, and developing a sense of identity as digital creators and problem solvers. In the following paragraph, I will discuss how CT concepts and practices are two distinct constructs and why they are separated.

In Brennan and Resnick's framework, computational thinking (CT) concepts and practices are distinguished to provide a comprehensive understanding of how individuals engage with and process computational tasks. The importance of this distinction lies in its ability to categorize the cognitive processes (concepts) and the methods (practices) involved in CT. CT concepts, such as sequences, represent the fundamental principles that underlie computational systems. On the other hand, CT practices, such as debugging and iterative refinement, are the methods through which learners apply these concepts in practical scenarios. Separating these two aspects is important as it allows educators and researchers to pinpoint where learners might excel or struggle, be it in the comprehension of abstract computational principles (concepts) or in the application of these principles in real-world scenarios (practices). This separation also guides curriculum development, ensuring that both the theoretical and practical sides of computational thinking are adequately covered in educational programs. In the following three paragraphs, I will incorporate the definitions of each of CT concepts, practices, and perspectives defined by Brennan and Resnick (2012). Let me start with CT Concepts.

Brennan and Resnick group CT concepts into seven categories: Sequences, events, conditionals, data, loops, parallelism, and operators. Sequences in programming refer to the expression of an activity or task through a set of discrete steps or commands that a computer can perform. Events are triggers in which one action initiates the occurrence of another action. Conditionals allow for decision-making based on specific conditions, facilitating the expression of various possible outcomes. Data pertains to the processes of storing, accessing, and modifying values. This dissertation focuses on the four mentioned CT concepts above. Loops are a programming feature that allows for the repetition of a sequence of instructions numerous times. Parallelism involves multiple sequences of instructions executing concurrently. Operators enable the execution of mathematical, logical, and string expressions, allowing programmers to carry out manipulations with numbers and text (Brennan and Resnick, 2012). These last three CT concepts discussed above, which are loops, parallelism, and operators, were not the focus of this dissertation due to the lesson design, and this will be discussed in the methods chapter. In the following paragraph, CT practices defined by Brennan and Resnick (2012) will be discussed.

They further define CT practices in four categories: Being increments and iterative, testing and debugging, reusing and remixing, and abstraction and modularization. Being incremental and iterative refers to the adaptive process of project design, where planning and implementation are not strictly sequential but evolve through small, progressive steps that allow for continuous refinement and adjustment in response to emerging solutions. Testing and debugging involve developing strategies to anticipate and resolve issues that arise during the design process, as the outcome seldom aligns perfectly with the initial conception. Reusing and remixing, refer to the practices of taking existing ideas, designs, or code, and reworking them into new creations, building upon and extending the work of others. Abstracting and modularizing involve creating larger systems or projects by assembling a series of smaller, more manageable components (Brennan and Resnick, 2012). The CT practice of reusing and remixing was not the focus of this dissertation due to the lesson design and this will be discussed in detail in the methods chapter. Below, I will define CT perspectives as characterized by Brennan and Resnick.

CT perspectives are characterized by Brennan and Resnick as “the perspectives designers form about the world around them and about themselves” (p. 1). CT perspectives are categorized into three ideas: Expressing, connecting, and questioning. Expressing, in the context of computational thinking, refers to the use of computational tools and processes as a means for individuals to create and convey their own ideas. Connecting means using communities such as Scratch Online Community and collaboration to expand one's creative abilities and learn from others. Questioning involves confidently using technology to explore and make sense of the world (Brennan and Resnick, 2012). The focus on the CT perspectives in this dissertation was limited, primarily due to the way the lessons were designed, and this will be discussed in detail in the methods chapter.

Note that this dissertation focused specifically on CT concepts and CT practices. The CT concepts incorporated were sequences, events, conditionals, and data. For CT practices, being incremental and iterative, testing and debugging, and abstraction and modularization were utilized.

Research on computational thinking in K-12 typically follows two distinct paths: treating CT as a stand alone subject and integrating CT into other disciplines like science and engineering. This dissertation focuses on the second approach, because the contents and contexts in STEM fields can improve CT learning (Orton et al., 2016; Weintrop et al., 2014). When CT is part of other classes, students can learn it while also learning about different areas like science. Integrating CT is important because it helps students learn how to solve problems in different subjects, which is a key skill for their future. It should be emphasized that the subject matter and situations within STEM fields have the potential to enhance Computational Thinking (CT) learning, as indicated by research from Orton et al. (2016) and Weintrop et al. (2014).

Computational Thinking Integration into Science and Engineering

Computational thinking integration into science and engineering refers to the incorporation of computational concepts and practices such as modularization into science and engineering education. It not only involves solving problems with both computers and physical approaches to understanding and addressing scientific and engineering challenges.

Studies on integrating computational thinking (CT) with science and engineering generally follow one of three methods: 1) CT is the end while science and engineering is the means; 2) treating both CT and science and engineering as equally important goals; 3) science and engineering is the goal whereas CT is the means (Arik and Topçu, 2022). This dissertation focuses on the first two approaches, meaning that it focuses on studies that either use science and engineering as a context for learning CT or treat both as important. I took this approach because CT is the main focus of this dissertation within a science and engineering context. The following diagram presents the roadmap of this dissertation's focus within the extensive literature.

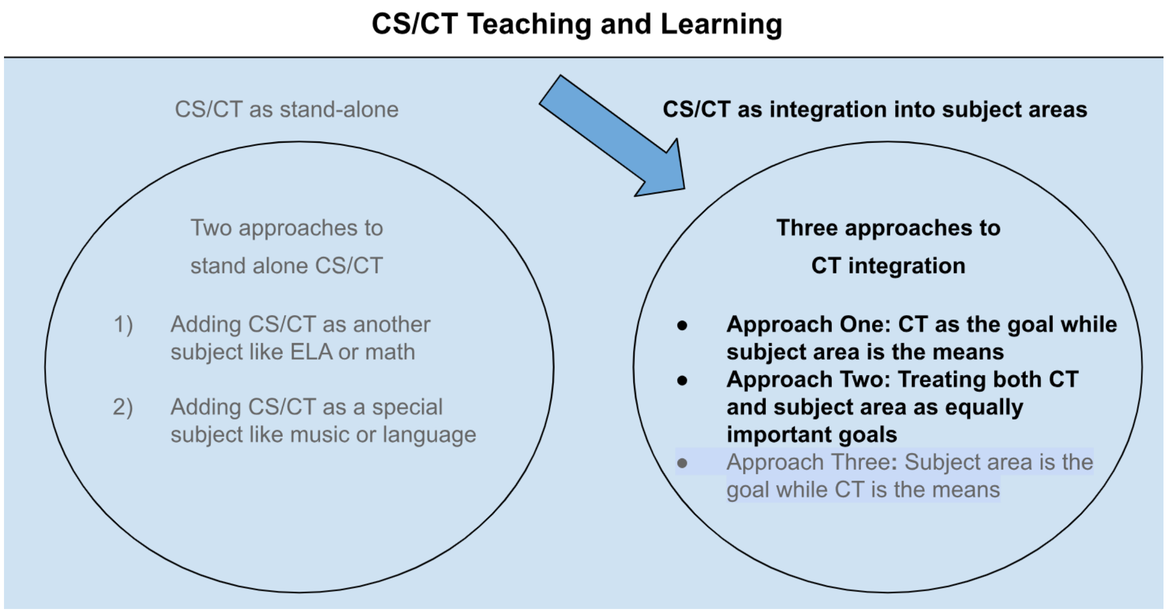


Figure 1: Roadmap of literature review

In the following sections, I will discuss the integration of CT concepts and CT practices in science and/or engineering in K-12 in the literature review based on the first two approaches above, which are using science and/or engineering as a context to enhance CT skills as “Approach One”; and giving equal importance to both CT and science and/or engineering as learning outcomes as “Approach Two”.

Approach One: CT as the Goal and Science and/or Engineering as the Means

Approach One includes studies where computational thinking is the goal and Science and/or Engineering is the context for CT learning and teaching in K-12 literature. The studies I will look at next all use this approach, and they often focus on older kids, not younger ones like second graders such as Kafai et al. (2014), Luo et al. (2020), and Yang et al. (2019). My dissertation fills this gap in those studies by focusing on these younger students. Also, these studies use the same CT framework by Brennan and Resnick (2012) that my dissertation uses. This discussion will be organized sequentially, starting with studies conducted at the elementary school level, followed by those at the middle school level, and concluding with research relevant to high school.

Adopting Brennan and Resnick’s (2012) CT framework, Luo, Antonenko, & Davis (2020) studied the evolution of two elementary school girls’ CT concepts such as loops and conditionals and CT practices such as decomposition and iteration. In the study, not having prior experience with coding, the students were engaged with a four-week CT-integrated science unit on the reproduction cycle of flowerless plants using Dash, a programmable robot, and an app used to program Dash, to drive forward, say “Hi,” have it tell the life cycle of ferns and then say “Bye.” or programming Dash to tell the differences between the fern cycle and the moss cycle at a summer camp through the Dash robot and the Blockly app. Observations, participant drawings, and analysis of Blockly code revealed that whereas both girls developed CT practices such as abstraction, decomposition, iteration, and debugging while engaging in coding tasks described above, their conceptions of sequence, loops, and conditional thinking did not show considerable change. Both children displayed behavioral engagement such as paying attention to instruction, emotional engagement such as showing much excitement about the robot (Dash), and cognitive engagement such as engaging in multiple attempts to have the perfect recordings of the plant story, which emphasizes the CT practice of being iterative. The authors suggested that to support CT and CT integration with science in an elementary-level informal learning context, it is especially important to define computing vocabulary, to utilize checkpoint activities with immediate and corrective feedback, and to scaffold coding concepts with unplugged activities. This dissertation parallels Luo et al.'s study by applying the Brennan and Resnick CT framework, yielding comparable insights. Unlike Luo et al., who explored CT in science education with a focus on comprehensive science and CT learning, this dissertation concentrates solely on students’ CT skills in a combined science and engineering setting. Moreover, it goes beyond using just programming tools like Blockly and Dash Robot, incorporating both Scratch and an unplugged, hands-on engineering activity, which fosters CT practices such as testing and debugging across digital and tangible design realms.

Ehsan, Rehmat, and Cardella (2021) found various CT competencies in their research where they examined the integration of computational thinking (CT) in engineering design activities, specifically with children aged 5 to 7, paralleling the focus of this dissertation on second graders. Their study involved a project where children, with support of their families, built a play area for a puppy in an out-of-school setting. Ehsan et al.’s approach to viewing engineering design as a context for CT development mirrors the methodology of this dissertation, which is Approach One, where science and engineering are not only backdrops but integral to the cultivation of CT skills. They employed two analytical frameworks to examine the relationship between engineering tasks and CT competencies. For instance, the CT skill of troubleshooting and debugging was linked with the engineering design process of revision. The study found that the children engaged in various CT competencies, including abstraction, problem decomposition, and simulation/testing, which align with the CT skills observed in this dissertation. The significance of Ehsan et al.'s study for this dissertation lies in its demonstration that engineering design activities can effectively facilitate the development of CT skills in young learners as in the task of building a play area for a puppy, which is akin to the house model construction activity in this dissertation, where students were required to adhere to specific design criteria, thus engaging in CT practices. Their findings support the concept that integrating CT within science and engineering education enriches the learning experience, enhancing students’ understanding of complex concepts. This is particularly relevant to my dissertation, as it highlights the effectiveness of combining CT with science and engineering tasks in an elementary school setting.

In their study on integrating computational thinking (CT) into STEM curriculum (STEM+C) via project-based learning (PBL), Yang et al. (2018) investigated the CT development of 72 fourth to sixth graders in an after-school program. Through two PBL initiatives, "Life on Mars" and "Building Earthquake-Resistant Bridges," they examined how participation influenced students’ CT competencies and attitudes towards STEM. Their analytical approach incorporated Sfard’s (2008) discursive framework, focusing on scientific conversation elements like word use and visual mediators, and connected to CT elements like vocabulary, abstraction, and decomposition, elements also found in Brennan and Resnick's CT framework. The findings showed students improved in CT practices such as testing and problem-solving strategies like abstraction, decomposition, heuristics/being incremental and iterative, data collection and analysis, simulation, modeling, particularly highlighted in the "Building Earthquake-Resistant Bridges" project. These practices resonate with the CT practices observed in second graders in the current dissertation, who constructed house models and developed Scratch programs in a science and engineering context. The improved attitudes toward Math underline the positive impact of STEM+C integration; however, there was no significant change in attitudes towards Science, Engineering, or Technology. The researchers concluded that the STEM+C projects gave students a setting to explore, learn, and use CT in their scientific research and problem-solving, which emphasizes the hands-on, exploratory nature of project-based learning and its effectiveness in integrating CT with STEM. This dissertation extends the work of Yang et al. by focusing on a younger cohort—second graders—and emphasizes CT within the context of science and engineering education, not just STEM broadly. The hands-on construction of house models alongside Scratch programming provides a nuanced understanding of CT development at an earlier educational grade, contributing to the literature on CT integration in elementary education and reinforcing the tenets of Brennan and Resnick’s framework.

Building on the Building Earthquake-Resistant Bridge project, Yang et al. (2019) delved deeper into the realm of CT in engineering through a case study with 36 upper elementary students engaged in a bridge design challenge over eight weeks. This study, conducted in an afterschool setting, aimed to discern how students employ CT practices in engineering contexts, mirroring the focus of my dissertation on second graders' CT understanding in science and engineering. Using video and artifact analysis like in this dissertation where student artifacts were analyzed, researchers applied the Brennan and Resnick’s (2012) CT framework, paralleling the one in my study, to scrutinize problem-solving steps alongside the K-12 engineering design process. This framework linked CT elements such as decomposition and algorithms to stages like problem identification and research. They found the students engaged in the CT practices of modularization, abstraction, data analysis, pattern recognition, heuristics, data collection/analysis during the engineering design process. Findings revealed that students' CT practices, such as using heuristics in testing and conditional logic in redesigning, were task-dependent. This finding resonated with the CT practices observed in second graders in this dissertation who constructed house models and Scratch programs, as outlined in Brennan and Resnick's CT framework. This dissertation complements Yang et al.’s findings by showcasing CT development through hands-on engineering design in a younger cohort. It adds to the body of research evidencing the adaptability of CT practices across different educational stages and design challenges as in this dissertation, reinforcing the integral role of CT in early science and engineering education.

Yang, Baek, and Swanson (2020) expanded upon their prior work discussed above by exploring how sixth graders utilize CT in a PBL unit on airplane design within a formal education setting, addressing the observation from Yang et al. (2018) that students' CT application was closely tied to specific tasks. In this study, involving 51 students from a suburban elementary school, the researchers employed the same CT analytical framework to align CT elements from Brennan and Resnick's (2012) with the engineering design process stages. A key tool for assessing the development of CT skills involved administering a test, both before and after the PBL unit. The results indicated a significant enhancement in students' CT skills post-intervention, underscoring the efficacy of integrating CT into broader educational activities beyond coding. The use of CT in the airplane design project facilitated students' problem-solving and subject learning, demonstrating CT's versatility in educational settings. In relation to my dissertation, this study provides evidence that integrating CT in project-based learning, such as the airplane design task, can be beneficial for elementary students—a concept that is echoed in my research with second graders engaging in house model construction and Scratch programming. It supports the argument that CT practices and CT concepts can be effectively developed through hands-on activities in science and engineering contexts, aligning with Brennan and Resnick's CT framework. This study, therefore, complements my findings by showing the positive impact of CT integration on older students, thereby enriching the current understanding of CT education across different grade levels.

Litts, Lewis, & Mortensen (2020) examined a place-based approach to CT integration, where young individuals in a rural Western U.S. city developed mobile games focused on civic and environmental issues relevant to their community. Over the course of six sessions totaling 12 hours, 29 participants aged 9-16 used ARIS, a narrative-based platform suitable for novices, to create location-based games. The researchers collected a variety of data, including audio recordings and interviews, to assess the CT practices as per Brennan & Resnick's framework. Their findings indicated that students adopted CT practices such as being incremental and iterative, testing and debugging, reusing and remixing, and abstracting and modularizing, as seen in their use of ARIS game objects for storyboarding and adjusting their games based on the platform's capabilities. This emphasizes the adaptability of CT practices across different age groups and the impact of the disciplinary context on the engagement with CT. The researchers argue that the disciplinary context may influence what CT practices students actually engage with.

Drawing on Brennan and Resnick’s CT framework and doing artifact interviews and observations, Kafai, Lee, Searle, Fields, Kaplan, & Lui (2014) studied computational thinking concepts and practices among a group of 15 high school students aged 16 to 18. These junior and senior students participated in a 10-week e-textiles module. The study explored combining knitting, typically seen as women's work, with coding in e-textiles, challenging traditional gender roles in these activities. Its goal is to spark girls' interest in coding and engineering. In this context, they conducted a study with high school students who were taught about CS concepts through sewing and physics. In this interdisciplinary integrative design, they examined CT concepts and practices that were reflected in students’ electronic textile (e-textiles) designs. They observed students developed three CT practices in their design processes through the engagement of the e-textiles module: Being incremental and iterative, testing and debugging, and reusing and remixing. The CT concepts students were engaged with were variables, sequences, conditionals, loops, and operators. The authors argue that the e-textile activities, which is STEM content, effectively engage students in a variety of computing concepts and practices, simultaneously expanding their perceptions of computing. Similar to Kafai et al.'s study, the current dissertation both utilize Brennan and Resnick's framework to analyze CT concepts and practices within integrative contexts. While Kafai et al. focus on engineering through e-textiles with high school students, this dissertation expands the context to include both science and engineering with elementary students. Both studies employ artifact interviews to uncover cognitive processes. This dissertation differentiates itself by targeting a younger demographic, which is second graders, thus addressing a research gap.

CT concepts and practices were not only studied by researchers as in “Approach One” discussed above, which is a lens to consider CT as the end and science and/or engineering as the means, but also examined in the literature as in “Approach Two”, which underscores the significance of both computational thinking (CT) and science/engineering as equally important disciplines. In the following sub-section, Approach Two followed by studies in K-12 in this approach will be discussed.

Approach Two: Computational Thinking and Science and/or Engineering as Equally Important

Studies taking Approach Two in K-12 address computational thinking and science and/or engineering as equally important. Studies that fall under Approach Two are important because CT is one of the main goals of those studies. This stance is relevant to the present dissertation since it treats CT with the same level of importance as science or engineering. A shared feature across various studies under Approach Two is the alignment of CT elements—such as CT concepts and practices—with the practices of science and/or engineering. Although these new frameworks differ from each other and from that of Brennan and Resnick, they essentially emphasize the primacy of CT concepts and practices, most of which were outlined in Brennan and Resnick’s CT framework. These components are consistently recognized as core across various distinct frameworks. One gap that has not been addressed in most of these studies in this approach is early elementary in a school setting such as second graders so this dissertation fills that gap.

In the upcoming paragraphs, I will explore the literature pertaining to K-12 education within Approach Two. This discussion will be organized sequentially, starting with studies conducted at the elementary school level such as Waterman, Goldsmith, & Pasquale (2020), followed by those at the middle school level such as Wilensky, Bready, and Horn (2014), and concluding with research relevant to high school such as Weintrop et al. (2016).

In their study, Waterman, Goldsmith, & Pasquale (2020) collaborated with third-grade teachers to revise a CT-integrated science unit titled "Survival of Organisms," focusing on population dynamics between deer and wolves. This unit, aligned with the Massachusetts Digital Literacy and Computer Science (MA DLCS) framework (2016) as in this dissertation, the Next Generation Science Standards (NGSS Lead States 2013), and Computer Science Teachers Association (CSTA) Standards (2017), incorporated CT concepts such as abstraction, algorithms, data, and modeling and simulation. Waterman et al.’s research reveals that students are not only mastering computational thinking (CT), but they are seamlessly weaving it together with science and math, which is consistent with Approach Two in this chapter. This remarkable synergy is most apparent in their adeptness at data collection and analysis. Moreover, the study underscores the pivotal role of both manual and technological tools in fostering engagement in computational thinking. This approach reflects the CT concepts and practices in the Brennan and Resnick (2012) framework employed in my dissertation. The parallel between this study and my dissertation lies in the focus on integrating CT into science education. However, a notable distinction is that while Waterman et al.'s study examined both science and CT comprehension, my dissertation is concentrated solely on exploring CT knowledge and skills in young learners. This difference highlights my study's specific emphasis on CT, distinguishing it as an in-depth investigation into computational thinking within the context of elementary science education.

Hynes, Moore, Cardella, Tank, Purzer, Menekse, & Brophy (2016) conducted research to bridge the gap in understanding how young children (ages 5-8) demonstrate engineering and computational thinking (CT). They aimed to incorporate CT elements into an existing STEM curriculum, PictureSTEM, and analyzed two units, Designing Paper Baskets and Designing Toy Box Organizers, for CT integration. Their observations, grounded in the CSTA & ISTE (2011) CT framework, align with the CT concepts and practices in this dissertation. In their study, they found that young students engaged in CT practices during activities. For instance, in the Designing Paper Baskets unit, students explored patterns through weaving, applying the CT practice of abstraction to recognize and follow patterns like "ABAB." Additionally, some students developed their own "AABAAB" weaving pattern, akin to creating an "algorithm." In the Designing Toy Box Organizers unit, students displayed CT concepts such as parallelization and practices like problem decomposition. They worked in teams, dividing tasks and collaborating to understand design constraints, similar to the approach in my dissertation where second graders engage in CT practices through hands-on activities like house modeling. This study highlights the feasibility of integrating CT in STEM education for young learners through unplugged activities. It demonstrates that even at a young age, students can grasp and apply fundamental CT concepts and practices, a key aspect of my dissertation's focus on CT integration in science and engineering education for second graders. The PictureSTEM curriculum thus provides a valuable model for aligning STEM learning with CT, reinforcing the importance of this integration in early education.

Cabrera, Ketelhut, Mills, Killen, Coenraad, Byrne, & Plane (2023) developed a CT framework aimed specifically at pre-service elementary teachers, addressing a gap in existing computational thinking (CT) frameworks which did not focus concurrently on elementary science and teacher education. Their work is an important addition to the field, especially considering the frameworks including those by Weintrop et al. (2016) and Sengupta et al. (2013). Their novel CT framework for integrating computational thinking into primary school science education comprises four key components, each with its own sub-categories. These include "Using Data," which focuses on activities like collecting and analyzing data with computational devices; "Programming," which encompasses problem decomposition, coding, and iterative testing and adjusting; "Simulations," involving the use of and assessment of computational simulations; and "Systems Thinking," which includes identifying and analyzing quantifiable parts of a system and understanding their interrelationships. Their framework aligns closely with the focus of my dissertation, which explores the integration of CT in a second-grade science and engineering context. The "Using Data" and "Systems Thinking" components are particularly relevant, as they mirror the CT concepts and practices my dissertation investigates through activities like house model construction and Scratch programming. The emphasis on practical, hands-on applications in Cabrera et al.'s framework resonates with the approach taken in my dissertation, demonstrating the effectiveness of integrating CT concepts and practices in early science education and providing valuable insights for teacher training in this area. In the following subsection, the learning theory of constructionism with its tenets, design of Scratch, and house modeling activity respectively will be discussed.

Lee, Martin, and Apone (2014) explored the integration of computational thinking (CT) in the K-8 curriculum during regular school hours. They proposed three strategies to enhance students' CT skills: the "Puzzles to Open Sandbox" in Digital Storytelling, the "Use-Modify-Create" progression in Computational Science Investigations, and "Computational Thinking with Data." Each approach is adaptable for teaching both science and CT, yet the "Use-Modify-Create" progression in Computational Science Investigations is particularly pertinent to this dissertation due to its focus on life sciences and ecosystems. This approach aligns with CSTA's K-12 standards and NGSS standards, emphasizing modeling and simulation to understand natural phenomena. An example provided by the authors is the use of StarLogo TNG for studying a fish and plankton ecosystem with 6th to 8th graders. In the "use" phase, students interact with pre-existing models, like determining a sustainable number of fish in a pond ecosystem, which involves the CT practice of reusing and concepts like variables and conditional thinking. The "modify" phase allows students to alter the ecosystem model, encouraging the creation of new algorithms and pattern recognition through iterative modifications. Finally, the "create" phase enables students to develop their own projects, applying skills like abstraction and analysis. Lee et al. highlight that despite challenges such as curriculum constraints, these methods for embedding CT into existing subject areas are feasible for K-8 educators. This study's insights into the "Use-Modify-Create" progression offer valuable guidance for integrating CT into science education, reflecting the efforts in this dissertation to blend CT with science and engineering in an elementary school context. The study demonstrates how CT concepts and practices can be effectively incorporated into regular classroom activities, resonating with the approach of using hands-on projects such as house modeling and Scratch to foster CT skills in younger students. In the following paragraphs, studies conducted in middle school under Approach Two will be discussed followed by one in high school.

Sengupta, Kinnebrew, Basu, Biswas, & Clark (2013) conducted a study with middle school students to integrate computational thinking (CT) into science education, focusing on physics and biology in order to see if scaffolding instruction is significant in students’ CT and science learning. Students learned kinematics and ecology before delving into CT concepts like conditionals and loops. The researchers used two groups of students: one receiving scaffolded guidance (S-Group) and the other with minimal scaffolding (C-Group), finding that scaffolding significantly enhanced students' learning in CT and science, which corresponds to Approach Two where they learned about CT and science together. This underscores the importance of tailored instructional support in CT integration. The study’s emphasis on abstraction as a core practice in both CT and science is particularly important. It reflects the integration of CT in my research, where second graders apply CT practices in an unplugged and plugged manner, using models and simulations created with tangible materials such as popsicle sticks. Sengupta et al.’s work reinforces the idea that CT can effectively support science and engineering education, whether through programming environments like Scratch or through physical model construction, as in the case of my dissertation with younger learners. This study is relevant to my dissertation as it demonstrates the application of CT, particularly through programming simulations, in a different age group and educational setting. Unlike my research where second graders engage in largely unplugged simulations, Sengupta et al. used a platform called Computational Thinking in Simulation and Modeling (CTSiM) for older students to write code for simulations. This difference in approach highlights the versatility of CT applications across different educational levels and the adaptability of CT practices such as abstraction and decomposition in varying contexts.

Cateté, Lytle, Dong, Boulden, Akram, Houchins, ... & Boyer (2018) conducted a study focusing on the integration of CS/CT into life sciences for middle school students. They collaborated with two local middle schools to incorporate CS/CT concepts into a life sciences curriculum through a combination of plugged and unplugged activities. This approach aligns with the central theme of my dissertation, which explores CT integration, but in a younger, elementary school setting. Their CT framework encompassed four key elements: Decomposition, pattern recognition, abstraction, and algorithmic design. The life science topics included epidemics and invasive species, explored through block-based programming environments like Cellular and Netsblox. This method of teaching science and CT through block-based programming tools allowed students to learn about both CT practices, such as decomposition, and life science concepts, which is Approach Two in this dissertation. Key insights from their study regarding student learning include the importance of differentiated instruction. They noted that providing extension activities for advanced learners and scaffolded, self-paced guides for those with less experience was crucial in maintaining engagement and ensuring comprehension. Moreover, they emphasized the necessity of familiarizing students with programming tools, advocating for ample time and scaffolded tutorials to learn these tools before starting the curriculum. Drawing parallels with my research, the need for differentiated instruction and tool familiarization in Cateté et al.’s study reflects similar considerations in teaching second graders CT concepts and practices through hands-on activities and Scratch programming.

A key aspect of a study conducted by Gautam, Bortz, and Tatar (2017), relevant to my dissertation, is the use of multiple modes of scientific representation. The researchers undertook a study in Nepal to integrate computational thinking (CT) into a 7th grade science curriculum, focusing on 16 students learning about photosynthesis and the carbon cycle. This integration, aimed at enhancing science learning and fostering CT skills, was conducted in a low-resource setting, with only four computers available for the entire class. Students engaged with various forms of representations, both digitally through NetLogo, an agent-based modeling tool, and physically, such as through textbook content and group posters. This approach is similar to the methodology in my dissertation, where second graders utilized both physical models (house model construction) and digital tools (Scratch programming) to represent and understand CT concepts and practices. The layered representation approach in Gautam et al.’s study is particularly instructive for my dissertation. It demonstrates how transitioning between different forms of representation, from static textbook depictions to dynamic digital simulations, can enrich students' understanding of complex concepts. This aligns with the approach in my dissertation, where the combination of constructing physical models and creating digital artifacts in Scratch serves a similar purpose. It provides a comprehensive learning experience, catering to students at various levels and reinforcing the effectiveness of multimodal representation in teaching CT and science.

Wilensky, Bready, and Horn's (2014) study on using NetLogo for agent-based modeling (ABM) in secondary science education aligns with both Gautam et al.'s (2017) study and the approach in this dissertation. Both studies utilize NetLogo to help students understand complex scientific concepts through modeling and simulation, akin to using Scratch for scientific exploration in this dissertation. Wilensky et al. demonstrate how NetLogo enables students to set up scenarios and explore outcomes, similar to Scratch, where students manipulate characters to understand different situations. This method effectively engages students in key science topics and CT skills such as abstraction. For instance, the Fire model in NetLogo allows students to simulate natural disasters, aiding their understanding of both the science involved and CT concepts. Interestingly, this parallels the unplugged activity of house modeling in the current dissertation. While the Fire model in NetLogo represents a plugged scenario, house modeling is an unplugged activity. Yet, both provide rich opportunities for students to engage with and understand scientific and CT practices, like abstraction and simulation. In both scenarios, students manipulate variables and observe outcomes in a controlled environment, whether it's adjusting parameters in a digital simulation or altering design elements in a physical model. These activities foster a deeper understanding of the underlying scientific principles and enhance CT skills, demonstrating the flexibility and effectiveness of both plugged and unplugged approaches in CT and science education. Wilensky et al.'s study highlights the value of tools like NetLogo and Scratch in enhancing science and CT education. These tools provide interactive experiences that help students visualize and experiment with scientific concepts, enriching their learning. This resonates with the methodology in this dissertation, where integrating CT into science education through programming tools like Scratch and non-programming activities like house modeling has potential to offer opportunities in enhancing student learning.

Aksit and Wiebe (2020) conducted a study in a middle school to explore the integration of computational thinking (CT) in a science class through a one-week Scratch program. Focusing on force and motion, 82 seventh-graders with minimal coding experience participated. The program aimed to enhance their understanding of these scientific concepts and CT skills like loops and conditionals, which is Approach Two in this dissertation. Post-program analysis revealed significant improvements in both science knowledge and CT skills. This study offers insights that resonate with the approach in my dissertation. Like Aksit and Wiebe’s participants using Scratch for simulating scientific scenarios, my dissertation involves second graders using Scratch for science and engineering projects. Both studies highlight the effectiveness of Scratch in enhancing students' grasp of CT concepts and practices within a scientific context. A notable finding from Aksit and Wiebe's research was the challenge students faced in creating algorithms and the importance of planning before coding. This parallels the iterative design process in my study, where students engage in planning, building, and revising their models and being iterative and step-by-step in Scratch. The study also underscores the necessity of teaching specific CT skills, such as debugging, which aligns with the emphasis in my dissertation on developing a comprehensive understanding of CT through hands-on activities such as house modeling and Scratch. Aksit and Wiebe's suggestion of using platforms with built-in debugging features for a more in-depth understanding of CT is relevant to considering the choice of educational tools. Their study, like this dissertation, demonstrates the potential of integrating CT into regular classroom activities, offering valuable lessons on how to effectively blend computational and scientific learning.

Arık and Topçu (2022) conducted a study to examine the effects of integrating unplugged computational thinking (CT) practices into a 6th-grade science unit on the digestive system, comparing it to traditional lecture-based methods. They creatively linked various stages of digestion to CT concepts such as algorithms, conditional thinking, and loops. Over a three-week period, they observed an experimental group engaged in unplugged CT activities and a control group receiving conventional instruction. The results showed that the group involved in CT activities demonstrated a superior understanding of the digestive system, evidenced by better model-based explanations. This study bears relevance to my dissertation, which investigates CT integration in a second-grade science and engineering context. While Arık and Topçu focused on the digestive system in sixth graders, my research encompasses younger students engaging in house model construction and Scratch programming. Both studies highlight the efficacy of CT practices in deepening students' comprehension of complex topics. My dissertation extends the application of CT practices to a younger cohort, illustrating the adaptability of these concepts and practices in enhancing learning experiences for various age groups in different educational settings. The following study under Approach Two is done in high school level.

In the context of my dissertation, which investigates the development of computational thinking (CT) in second graders within a science and engineering module, Weintrop et al. (2016)’s study offers valuable insights. While their research develops a taxonomy for CT in high school math and science, its significance goes beyond educational levels, aligning with key components of the CT framework used in this dissertation, as outlined by Brennan and Resnick (2012). Weintrop et al.’s taxonomy categorizes CT into data practices, modeling and simulation, computational problem solving (including debugging), and systems thinking, which parallel the CT practices observed in my study, such as debugging in model construction and Scratch programming. This is particularly pertinent to my research as it demonstrates how complex CT practices, like debugging and systems thinking, can be identified and developed even in younger students. By exploring how high school students engage with CT in math and science, Weintrop et al. provide a model for CT integration that is informative for my study, underscoring the adaptability of CT practices and concepts in different learning contexts and age groups. Thus, their research contributes a foundational perspective on CT integration that enriches the understanding of how CT skills can be developed and applied in early science and engineering education.

Theoretical Framework

In light of the literature discussed earlier, the learning theory of Constructionism, as articulated by Papert (1990), is not just appropriate but central to the methodology of this dissertation. Rooted firmly in the principle that active creation is the cornerstone of learning (Papert & Harel, 1991), Constructionism resonates powerfully with the core activities of this study. Participants in this research didn't merely engage with the concepts; they actively brought them to life through the construction of house models and the creation of Scratch programs. This emphasis on making and creating as integral to instruction is important in the development of computational thinking (CT) knowledge and skills. From a constructionist perspective, it is through this active process of creation that deep, meaningful learning occurs. The activities in this study exemplify this principle, demonstrating that learning in a constructionist environment is dynamic, engaging, and profoundly effective in fostering CT. Based on these ideas grounded on constructionism such as learning by making, the researcher of this dissertation defines learning as an active and reflective process where learners engage in the creation of artifacts—both physical and digital—that serve as external and shareable expressions of their understanding. These artifacts, conceptualized as 'objects-to-think-with,' embody the learners' intellectual journey, facilitating personal identification and cultural connection. This process emphasizes 'learning by making,' where the act of building and sharing something tangible catalyzes the development of knowledge structures. Through constructionism, learners are not mere recipients of pre-defined knowledge; instead, they are the architects of their learning, using the artifacts they create as tools for thinking, reflection, and social exchange. In the following paragraphs, I will discuss the learning theory of Constructionism that is aligned with this dissertation by focusing on its three principles, learning by making, building something external and shareable, and objects-to-think-with.

Constructionism

Constructionism, as the guiding learning theory for this dissertation, emphasizes the concept of “learning by making” with the creation of artifacts embodying the learner's journey and understanding over time. This principle aligns well with the activities undertaken by the participants in this study, who engaged in making house models and Scratch artifacts. Papert's constructionism, rooted in Piaget’s constructivism, emphasizes the learner's active role in building their own intellectual structures to understand the world. As Papert (1990) states, he views children as builders of their intellectual structures, drawing from Piaget's model. Papert and Harel (1991) describe constructionism as “learning-by-making,” highlighting the active construction of knowledge rather than its passive reception. This idea is echoed by various scholars, including Harel (1993), Papert (1993), Kafai and Resnick (2011), and Papavlasopoulou et al. (2019), who all underscore the active role of learners in creating their understanding. Within this framework, Kafai and Resnick (2011) assert that children don't just receive ideas; they make them. This is a fundamental tenet of constructionism, emphasizing that learning is most effective when learners are actively involved in creating something tangible, be it a physical object or a conceptual theory. The process of creation is central to learning, allowing learners to construct knowledge through their active engagement and exploration.

At the heart of Constructionism is the belief that learners actively construct knowledge through the process of creating objects. This aligns with the view of Kafai and Resnick (2011), who suggest that the creation of external, shareable artifacts often leads to the formulation of new ideas. Kafai (2011) points out that “constructionism suggests that learners are particularly likely to make new ideas when they are actively engaged in making some type of external, shareable artifact — be it a robot, a poem, a sand castle, or a computer program, which they can reflect upon and share with others” (p. 1). Therefore, objects that learners create could be either physical objects such as a house model on natural disasters or digital objects such as an artifact on Scratch. Furthermore, Papert (1990) expands on Piaget's constructivism by emphasizing the construction of something external or shareable, such as a sand castle or a computer program, as integral to learning. He describes a cyclical process of internalizing external elements and then externalizing internal thoughts. Papert (1990) argues the following in the sense of learning as something external or shareable while comparing constructivism with constructionism:

> We understand “constructionism” as including, but going beyond, what Piaget would call “constructivism.” The word with the v expresses the theory that knowledge is built by the learner, not supplied by the teacher. The word with the n expresses the further idea that this happens especially felicitously when the learner is engaged in the construction of something external or at least shareable...a sand castle, a machine, a computer program, a book. This leads us to a model using a cycle of internalization of what is outside, then externalization of what is inside and so on. (Papert, 1990, p. 3)

Kafai and Resnick (2011) emphasize that Constructionism builds on constructivist principles by highlighting the creation of external, shareable artifacts by learners. In this dissertation, such an approach is evident as second graders create both physical models and digital Scratch projects, which serve as concrete manifestations of their learning in computational thinking (CT). These artifacts, embodying the learners' understanding, are central to the study's exploration of CT learning.

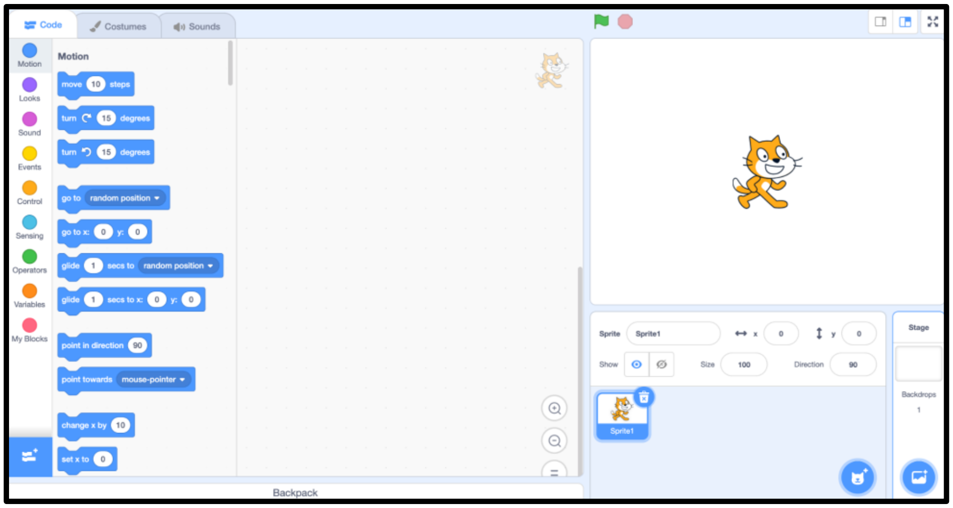
Papert (1980) introduced the concept of “'objects-to-think-with” which are entities that combine cultural presence, embedded knowledge, and personal identification. According to

Papavlasopoulou et al. (2019), coding activities provide such objects, allowing children to engage in thoughtful reflection. Therefore, both house models and Scratch projects in this study serve as “objects-to-think-with” in the realm of CT.

Papavlasopoulou et al. (2019) highlight the role of modern technological tools in facilitating constructionist learning, where children take charge of their creations, leading to meaningful learning experiences (p. 417). Scratch is an exemplary tool in this context, known for supporting constructionist learning and offering children enriching, personally relevant experiences (Maloney et al., 2010; Brennan and Resnick, 2012; Papavlasopoulou et al., 2019). In the following paragraphs, the design of Scratch for learning by making will be discussed.

Design of Scratch for Learning by Making

Visual programming languages have become ubiquitous in educational settings, particularly for teaching programming concepts and various subject areas to younger students, such as those in second grade. One of the most used visual programming languages is Scratch created by the Lifelong Kindergarten group at the MIT Media Lab. Scratch is based on the ideas of constructionism and Logo (Maloney et al., 2010). Scratch is a user-friendly, drag-and-drop, and visual programming tool, which makes it easy for young learners like second graders to utilize it. Scratch interface can be divided into three categories as shown in the screenshot below:



Sprite Sprite1 x 0 y 0
Show Size 100 Direction 90

Figure 2: A screenshot of Scratch interface

On the right is an area called stage where users can choose their characters and backdrops and see and manipulate them on the screen. On the left is an area where there are more than 100 programming blocks categorized into eight different groups (motion, looks, sounds, events, control, sensing, operators, and variables), and each category has multiple blocks in it. In the middle is an area called code area into which users can drag and drop blocks on the left and snap those blocks to create scripts for their character(s). Resnick et al. (2009) state “We wanted to make it easy for everyone, of all ages, backgrounds, and interests, to program their own interactive stories, games, animations, and simulations, and share their creations with one another” (p. 60). Importantly, “everyone” can include young children.

Resnick et al. (2009) designed Scratch based on three constructionist design principles to make it (1) more tinkerable, an idea I further elaborate below, (2) more personally meaningful, and (3) more social than other programming environments. In the following paragraph these three principles are discussed.

First, Maloney et al. (2010) say that “Scratch is tinkerable because it lets users experiment with commands and code snippets the way one might tinker with mechanical or electronic components. Tinkerability encourages hands-on learning and supports a bottom-up approach to writing scripts where small chunks of code are assembled and tested, then combined into larger units” (p. 4). Second, Resnick et al. (2009) argue that “people learn best, and enjoy most, when working on personally meaningful projects” (p. 64). They state that they developed Scratch based on two criteria to make it more meaningful: Diversity and Personalization. Since Scratch supports many different types of projects (stories, games, animations, simulations), Scratch appeals to widely varying interests. So, users can work on projects they like. With regards to personalization, the authors state that they designed Scratch in a way that people are easily able to personalize their Scratch projects by importing photos and music clips, recording voices, making stories, and even connecting external devices to their projects such as the Makey Makey kit. Third, Resnick et al. (2009) argue that the Scratch online community makes Scratch more social than other programming languages because it enables people to support, collaborate, and critique one another and build on one another’s work. As a result, Scratch was designed based on constructionism with the support of the above mentioned design principles.

Moreover, Maloney et al. (2008) argue: “...the design of the Scratch blocks simplifies the mechanics of programming by eliminating syntax errors, providing feedback about placement of command blocks, and giving immediate feedback for experiments (p. 371).” In the sense of “for everyone, of all ages, background, and interests”, Scratch was used at Computer Clubhouses (Maloney, 2008) in its early stages to offer an opportunity for children from underserved communities such as African American and Latino youth ages 8-18 to meet their programming needs. After being used in after-school contexts informally with the help of voluntary, inexperienced mentors such as in computer clubhouses, Scratch is now being used in school settings and has become one of the most used visual programming tools around the world. In the following two paragraphs, I will discuss why I think Scratch interviews were likely to reveal computational thinking.

Scratch interviews were likely to reveal CT concepts and CT practices for two reasons. One is that Brennan and Resnick (2012) developed the CT framework, which this study uses, in the context of Scratch. They state that “The phrase computational thinking helps us think about learning with Scratch, and, in turn, we believe that programming with **Scratch provides a context and set of opportunities for contributing to the active conversations about computational thinking**” (p. 2) This suggests that Scratch is inherently designed to engage users in Computational Thinking, making it a suitable tool for exploring CT concepts, practices, and perspectives.

The other reason is that as Maloney et al. (2010) quote above where they argue that “Scratch is tinkerable…” (p. 4) highlights several features of Scratch that make it an excellent platform for revealing Computational Thinking (CT) concepts and practices. Incremental and Iterative: Scratch's "tinkerability" aligns with being incremental and iterative. Users start small and build up, much like CT processes. Abstraction and Decomposition: Tinkerability encourages breaking problems into smaller pieces (decomposition) and then combining them into larger units (abstraction). The tinkerability of Scratch is closely aligned with key CT concepts and practices, making it ideal for cultivating and observing Computational Thinking.

Design of House Model Activity for Learning by Making

In this dissertation, second graders initially engage in constructing physical house models before transitioning to Scratch, where they program the three key phases of the house model activity: planning, testing, and proposing revisions. This progression from an unplugged, tangible activity to the digital realm of Scratch programming embodies the essence of constructionist learning. Particularly for children of this age group, beginning with physical/tangible activities is a highly effective way to facilitate learning. This strategic shift from concrete modeling to abstract representation in Scratch not only aligns with constructionist principles but also forms a core methodological approach of this dissertation. The sequence from constructing house models to reflecting and expanding on these experiences in Scratch is a deliberate and pivotal aspect of the learning design. Below, I will elaborate how the house model activity is aligned with the learning theory of constructionism.

Incorporating the principles of constructionism, the house model activity designed for second graders in this curriculum serves as a prime example of learning by active making. This module effectively integrates computational thinking (CT) into science and engineering for young learners, specifically catering to the developmental needs and capabilities of second-grade students. The curriculum is meticulously structured to first engage these young minds in hands-on activities, where they delve into research, gather data, and use this information to construct and test physical house models. This initial focus on tangible, physical activities is crucial for this age group, as it aligns with their developmental stage, which favors concrete experiences as a foundation for learning. Following this, the curriculum seamlessly transitions students from the tactile experience of model building to the abstract realm of coding in Scratch. This deliberate sequencing - from manipulating physical materials to digital expression - is not arbitrary. It is rooted in constructionist learning theory, underscoring the importance of starting with tangible experiences for younger learners before moving to more abstract computational representations. This approach of beginning with physical creation and then transitioning to digital coding is crucial, particularly for second graders, as it scaffolds their learning in a manner that is both developmentally appropriate and consistent with constructionist principles. The reverse sequence, starting with coding and then moving to physical modeling, would not be as effective for this age group, underscoring the thoughtful design of this curriculum in aligning hands-on activities with digital exploration.

In this hands-on learning process, students were not passive learners; rather, they were creators, using tangible materials to construct models that represented their understanding of scientific principles. This approach mirrors the core idea of constructionism, where learning is most effective when students are actively involved in creating something meaningful and shareable.

The integration of Scratch programming further enriched this learning experience, enabling students to digitally represent and reflect on their projects. This combination of physical model building and digital representation aligns with constructionist ideals, providing a holistic learning experience that blends the physical and digital worlds.

Overall, the house model activity, like the bridge design challenge in Yang et al. (2019), the airplane projects in Yang et al. (2020), and the puppy house in Ehsan et al.’s study (2021) discussed in the literature review, demonstrates the efficacy of constructionist learning in a real-world context, enhancing students' understanding of both scientific concepts and CT skills. In the following chapter, methods used will be discussed.

CHAPTER III

METHODS

The research design of the present dissertation study is qualitative study. Methods for data collection are artifact interviews, field notes, and pictures of students’ classroom artifacts taken during lessons. I used a data analysis method that uses theory-driven codes and a data-driven code for house model interviews, and only theory-driven codes for Scratch interviews. In the following sections, research design, context of the study, settings and participants, curriculum design/tools will be discussed followed by gaining entry and IRB, data sources, data collection, data analysis, and trustworthiness.

Research Design

The current research was done with 2nd grade students from three different classrooms in two different schools in an urban area of western Massachusetts. Two teachers (one teacher in her one classroom at her school and another teacher in her two different classrooms at her school) at two different schools implemented a 7-lesson CT integrated science and engineering module and students mostly worked in groups of two and each group built a house model and a Scratch program. I interviewed them about their house models and Scratch programs and asked them open-ended questions during their interviews.

In the study, I sought to understand second graders' experiences and insights related to their house models and Scratch projects. To accomplish this, I conducted a series of semi-structured interviews. Specifically, there were 15 interviews focused on the house models and another 15 interviews centered on the Scratch programs. In total, 28 second-grade students participated in these interviews. According to the way the module was designed, teachers formed the student groups (mostly groups of two where possible) during the lesson implementation but teachers purposefully grouped students who were allowed to be interviewed. Students in three classrooms participated in the lesson sequence and mostly worked in groups of two but some groups were three due to group conflict and some students worked alone due to parental consent issues. As a result, groupings during lessons informed interview groups for this study. After interviewing the student groups about their house models, the same groups were interviewed about their Scratch programs. Thus, across the entire study, 28 students were interviewed in 30 separate sessions for both house model and Scratch. These interviews were conducted during and after the students had completed the curriculum adopted by their two teachers.

The researcher sought to produce a thick, rich description of the participants' lived experiences on their house models and Scratch artifacts because rich description is an important characteristic of qualitative research as Merriam and Tisdell (2015) state "[...] the product of a qualitative inquiry is richly descriptive. Words and pictures rather than numbers are used to convey what the researcher has learned about a phenomenon" (p. 17).

Second graders were interviewed about their house models and Scratch programs which they created during the integration of the module. Ginsburg (1997) argues that clinical interviews are a powerful technique for gaining insight into a child's way of thinking. Ginsburg writes that during a clinical interview the interviewer asks open-ended questions such as "how did you do it"? or "why" [...] (p. 34). Artifact interview evolved out of clinical interview and it will be discussed in detail under data collection.

Context of the Study

This dissertation project was a part of a larger NSF-funded CSforAll research-practice partnership (RPP), which is taking place in the elementary public schools of the district, Massachusetts (Adrion et al., 2022). The district has 58 schools serving approximately 25,000 students. Our RPP Project is targeting approximately 12,000 K-5 students in 33 elementary schools. The student population in the district public schools is majority historically minoritized: 66.6% Latinx, 18.9% African American, 10.2% white students, and 4.3% Asian, Native American, non-Hispanic, and multi-race students. 83% of district students are considered high needs, and 76.7% are economically disadvantaged. On the other hand, 68.7% of teachers are White, and 79% are female.

The CSforAll-RPP began in 2018, and this study took place in the second semester of year four, which is the 2022 spring semester. Teachers in the larger project were recruited from two grades each of the first three years to work with researchers to co-design and implement integrated CS/CT curricula. In year 3, eight 2nd grade teachers worked in dyads to integrate the CS/CT concepts and practices defined in the Massachusetts Digital Literacy and Computer Science (DLCS) standards (2016) with other standards-based curricula in English language arts, mathematics, science and social studies. Using an iterative process, the teachers implemented, assessed, refined, and documented the curriculum. In year 4, all classroom teachers in 11 "Cohort A" schools were collaborating with newly hired computer science teachers in each school to teach the integrated K-5 curriculum to approximately 3500 students. Eleven additional schools adopted the curricula in all classrooms in the two following academic years, reaching all 33 district schools and all 12,000 students. During the co-design process, each dyad developed a module /unit of 5-10, 45-minute integrated lessons. Four modules made up 2nd-grade CS/CT integrated curriculum, each expected to be taught in one quarter of the academic year. The current module on natural disasters was module 3 at the time of data collection, meaning that it was supposed to be implemented during the third quarter of the academic year, which was around spring semester in 2022. Data were collected in three different 2nd-grade classrooms in two different schools among the 11 Cohort A participating classrooms.

Settings and Participants

It's important to note that all names mentioned in this dissertation have been replaced with pseudonyms. The CSforAll curriculum was taught in all classrooms in the 11 Cohort A district Elementary Public Schools and some classrooms in the other 22 schools during 2021-2022. However, the present small study is concerned with only 2nd grade students in the larger project. Data were collected from three second grade science classrooms in two different schools when the CT integrated-science and engineering module was implemented. Below is Table 2 in which schools, classrooms, teachers who implemented the module, classroom size, and total number of students interviewed are shown followed by a description of it based on the report of the teachers.

Table 2: Settings and Participants

School	Teacher	Classroom	Classroom size	Total number of interviewees
School X	Mary	Classroom A	22	12
		Classroom B	20	10
School Y	Kimberly	Kimberly's classroom	17	7

In this study, I collected data from three different 2nd-grade classrooms across two schools, where teachers Mary and Kimberly implemented a CT-integrated science module. At School X, Mary taught the module at different times in two of her second grade classrooms, and I observed her in both classrooms while she was implementing the module. In her first classroom, known as Classroom A, there were 22 students according to her report. Out of these, I was able to interview 12 students because the rest did not have parental permission to be interviewed. In Mary's second classroom, Classroom B, there were 20 students according to her report. Here, I interviewed 10 students,, as only these students had parental consent. At School Y, Kimberly taught the module in two different classrooms but I could only observe one of her classrooms due to a scheduling conflict with Mary's Classroom B. Kimberly's observed classroom had 17 students. Out of these, I interviewed 6 students. At the time of data collection, the ages of the students across the three classrooms ranged from 7 to 8 years old. In total, although the plan was to interview all students across the three classrooms, I was limited to interviewing 28 students in a total of 30 interviews (15 house model interviews and 15 Scratch interviews) due to issues of parental permission in each classroom. Data was collected from both of Mary's classrooms and only one of Kimberly's classrooms. While Kimberly's school and classroom accurately reflected the ethnic diversity of the district, Mary's school and classrooms also mirrored the district's ethnic composition. However, Mary's school did not represent the district's socioeconomic diversity, as it tended to be wealthier.

Who was allowed to be interviewed in those classrooms? Opt-out parental consent forms were sent to the parents through students’ teachers at least twice in English and Spanish, as 69% of the district is Hispanic. Only students whose parents gave consent were interviewed. Besides parental consent forms, I asked interviewees to sign an assent form written in language that was easy for them to understand.

I would like to provide some background about the two teachers who implemented the module, as they were also participants of the current dissertation. Based on my field notes, Mary shared that they started their teaching journey in 2001 and transitioned to being a technology teacher around 2015. That same year, 2022, they embarked on teaching the second grade at the school from which dissertation data was collected. Besides their teaching role, Mary currently serves as the Digital Literacy and Computer Science (DLCS) coordinator for the CSforAll RPP project and they contributed to the CT integrated science and engineering module by taking a role in revising the module. On the other hand, according to my field notes, Kimberly revealed that she began teaching in 1999 and, while she's had a career span as a technology teacher, the recent five years were specifically spent in a classroom setting. Notably, out of these years, it was her first year teaching at her school where the dissertation data was gathered.

Curriculum Design/Tools

In the larger RPP project, the 2nd grade teachers in four dyads integrated CS/CT concepts into their existing curriculum, which consisted of four second grade modules that focused on different content areas such as ELA and science, during 2020 and 2021. After piloting the curricula in their classrooms, they revised and documented the curriculum so that all 2nd grade teachers in the district Public Schools can use the documented curriculum in their classrooms. The participants of the present study were engaged with one of those modules, which was a 7 lesson CT-integrated science and engineering module. Students built house models and Scratch programs through the engagement of the module.

Before delving into the details of the module, it's important to note that there have been multiple revisions of it. The data for this study comes from the version dated August 14, 2021. This version was updated by the original creators of the module, with input from several district teachers and three coordinators from the CS4All RPP project. Mary, one of the teachers in this study, was also among those coordinators. While there are different versions of the module, the standards, objectives, and the core content has largely remained consistent. Below, I discuss the module from which I collected data and what standards the module was developed on.

Module Three/Natural Disasters Module

In module 3, which is named Natural Disasters, instruction targeted Massachusetts DLCS standards (2016) such as modeling and simulation, data, and abstraction and Science Technology and Engineering (STE) standards such as “Investigate and compare the effectiveness of multiple solutions designed to slow or prevent wind or water from changing the shape of the land.”. A full script of all the standards that module 3 was developed on can be found in Appendix.

Module three was a 7-lesson science and engineering module that focuses on natural disasters, engineering, and computational thinking. Modeling and simulation are the core CS/CT concepts that were integrated into the natural disasters topic in science. Students usually worked in pairs in the implementation of this module but there were occasions where they only listened to their teacher. In this entire module, students designed, built, tested out, and proposed a revision to house models that were meant to be strong enough to sustain natural disasters. The curriculum was structured into seven distinct lessons, which I will discuss below along with what CT concepts and practices I expected students to engage in during the lessons.

Lesson 1: Students began by exploring how models can represent real-world systems, utilizing Makey Makey (a kit designed to connect everyday objects to computer keys) to do so. In this lesson, teachers introduced students to the concepts of models/modeling, simulation, and real life systems such as the solar system.

Lesson 2: The focus shifted to simulations, where students learned what specific concepts are illustrated by a given simple simulation. Teachers defined what a simulation is and had students watch simulation videos and play a simulation game on their laptops to make students familiarize with the idea of simulations.

Lesson 3: Students engaged in independent research on natural disasters like floods and hurricanes using informational texts. They collected data on building materials and their resistance to natural disasters. Within the same lesson, they also began planning their house models on a poster board. The board had three columns: one for materials they planned to use, a second for simulation results, and a third for proposed revisions. In this lesson, students engaged in data collection and filled out the first column of their poster board, which is what materials to use and why, based on their research. Therefore, I expected that students' research on natural disasters could be a theme called data collection.

Lesson 4: Based on the plans made during lesson 3 and getting an approval of their plan from the teacher, students in groups began constructing their house models, aiming to make them robust enough to withstand simulated natural disasters. I expected students to build their house models in an iterative manner.

Lesson 5: The teacher conducted tests on the house models using fans for wind and spray bottles for rain, which represents a simulation of a hurricane. Students observed all simulations and recorded the outcomes of their house model simulation on their poster board, for which I expected the CT practice of testing/simulating to arise in their house model interviews. Their observation of simulations allowed them to reflect on the effectiveness of their designs. I expected that their suggestions for modifications aligned with the CT practice of debugging.

After Lesson 5, I began interviewing students about their house models. This timing was crucial because it was during this lesson that they observed simulations of their designs.

Lesson 6: Students were given an introduction to Scratch programming and were taught the purposes of various blocks, the basic units of code in Scratch, resembling puzzle pieces, that represents a specific command or function, such as moving a character, playing a sound, or making a decision. For instance, students learned that event blocks serve to initiate a program. However, for this specific project, the selection of blocks was pre-determined in the module. Thus, teachers guided the students on which blocks to use. Typically, for each of their three sprites, students utilized both a Makey Makey event block and a sound block. In their Scratch projects, students discussed their house model plans, the outcomes of the simulations, and proposed changes or revisions to improve their designs based on the data collected from the simulations. In other words, students visualized their whole data on Scratch. The fact that scientists use data visualization to help show others their data was emphasized in this lesson.

Lesson 7: In the final lesson, students connected Makey Makey with Scratch, enabling them to control their Scratch projects. They showcased these projects to their classmates. It was at this point that I began interviewing them about their Scratch creations. The timing of these interviews was deliberate, as this lesson marked the completion of their Scratch projects. Additionally, during this session, students were asked to suggest improvements to their house models and to draft new plans based on their proposed changes. My primary objective was to delve into their comprehension of their own Scratch program—understanding why they designed it the way they did. I was also interested in learning about the modifications they suggested for their house models, including their rationale for choosing or avoiding certain materials in their revisions. Therefore, I expected the CT practice of debugging to come up not only in their house model interviews as mentioned above, but also in their Scratch interviews. Also, since students used Scratch in lessons 6&7, I expected their Scratch interviews to reflect CT concepts and CT practices defined by Brennan and Resnick (2012), as they argue Scratch is likely to enable students to engage in those (Please refer to the methods chapter where I discussed the design of Scratch for enabling Scratchers to develop those concepts and practices).

In summary, each lesson served a unique role, from the introduction of models and simulations in Lessons 1 and 2, to data collection and planning in Lesson 3, model building in Lesson 4, testing in Lesson 5, and finally, digital representation and presentation and improving their house model plan, which is proposing a revision or debugging, in Lessons 6 and 7. Even though each lesson was designed to be taught in one day, some took longer than intended. For example, Lesson 4 extended over multiple days because students needed more time to build their house models.

Gaining Entry and IRB

The data were collected after receiving permission from the Institutional Review Board (IRB) at the University of Massachusetts Amherst. Opt-out parental consent forms were sent to the parents through students' teachers at least twice in English and Spanish, as %69 of the district is Hispanic. Only the students whose parents opted out of providing their children were interviewed. Besides parental consent forms, I also asked interviewees to sign an assent form written in language that was easy for them to understand. In case they did not understand what was written in the assent form, I requested the classroom teachers to explain to their students what was written in it because I assumed that students would trust their teachers. Student participation in the research was voluntary and they could drop out at any time with no consequences.

Data Collection

In this section I will discuss how data was collected, data sources of the present study followed by chronology of the data collected, data collection instrument, which is artifact interviews, and interview protocols.

The data was collected in person through artifact interviews (Brennan and Resnick, 2012), field notes, and pictures of students' classroom artifacts such as poster boards. Artifact interview methods will be discussed in detail below. 11 interviews on house models were conducted with groups of two students while 3 house model interviews were done with three individual students and one house model interview was carried out with a group of three. Same groups were interviewed about their Scratch programs. Although all students in a group had permission to be interviewed, only one student participated in their house model interview and did not partake in their Scratch interview, likely due to the absence of the other student.

For the house model interviews, I positioned my camera to focus on the students' house models, which were placed on a table. The students sat around this table, and I recorded their verbal responses while ensuring that their faces were not captured on camera. These interviews typically lasted between 10 to 15 minutes each. Similarly, for the Scratch interviews, I recorded the students' verbal responses as well as their laptop screens, which displayed their open Scratch programs. Again, I took precautions to avoid recording their faces. These interviews lasted between 10 to 15 minutes each.

The data sources were multiple including students' house model interviews, Scratch interviews, Scratch programs as well as photographs of classroom artifacts such as house models, poster boards, and Scratch programs. Additionally, field notes were conducted through classroom/lesson observations. Table 3 below shows what data sources were used to address research questions.

Table 3: Data sources used to address research questions

Research Questions (RQ)	**Data sources used to address the RQ**
RQ1: What CT practices did second graders develop through the engagement with the CT integrated science and engineering module while planning, building, testing, and proposing a revision to their house models, and how did they emerge?	● House model interviews (main data source) ● Scratch interviews (students frequently discussed their house models during Scratch interviews, particularly their revision plans) ● Field notes conducted (through classroom/lesson observations) ● Photographs of classroom artifacts such as house models and poster boards
RQ1a: How do 2nd graders describe their understanding of modeling and simulation?	● House model interviews ● Scratch interviews (students frequently discussed their house models during Scratch interviews, particularly their revision plans) ● Field notes (Please note that only field notes were used to answer the modeling part of this RQ)
RQ1b: What CT practices did 2nd graders develop in planning their house models?	● House model interviews (main data source) ● Scratch interviews (students frequently discussed their house models during Scratch interviews, particularly their revision plans) ● Field notes conducted (through classroom/lesson observations)

	● Photographs of classroom artifacts such as house models and poster boards
RQ1c: To what extent did participants' initial house models survive from the simulation/test?	● House model interviews (main data source) ● Scratch interviews (students frequently discussed their house models during Scratch interviews, particularly their revision plans) ● Field notes conducted (through classroom/lesson observations) ● Photographs of classroom artifacts such as house models and poster boards
RQ1d: Which CT practices did second graders engage in describing and understanding their material choices they made in creating their house models and proposing a revision?	● House model interviews (main data source) ● Scratch interviews (students frequently discussed their house models during Scratch interviews, particularly their revision plans) ● Field notes conducted (through classroom/lesson observations) ● Photographs of classroom artifacts such as house models and poster boards
RQ2: How do they describe in Scratch the process they undertook?	● Scratch Interviews (main data source) ● Students' Scratch programs ● Photographs of their Scratch artifacts ● Poster boards

Chronology of the Data Collected

To provide a clearer understanding of the data collection process, it's important to discuss its chronology. The data of the current dissertation study was collected during 2022 Spring semester from three different classrooms at two different schools.

Students were interviewed twice during this study. The first round of interviews occurred right after the simulation tests on their house models (Lesson 5). These interviews focused primarily on the students' experiences and learnings from building and testing their models. The second round of interviews was conducted after students completed their Scratch projects (Lesson 7). In the following paragraphs I will discuss in how many days the two teachers implemented the 7-lesson module followed by the structure I took in my field notes and the chronology of the data collected.

Mary took 13 days to complete the 7-lesson module in both of her classrooms. In contrast, Kimberly finished the module in 9 days. Due to scheduling conflicts with Mary's first classroom, which I call Mary' classroom B, I couldn't attend the first four lessons in Kimberly's classroom. These initial lessons covered topics like modeling, simulation, research on natural disasters, and material selection for their house models. As a result, I have no field notes from those first four lessons in Kimberly's classroom. My data collection from Kimberly's classroom started from lesson 5, where students began building their house models. In the subsequent paragraphs, I will discuss the chronology of the data collected. Below is Table 4 showing the timeline of data collection.

Table 4: Timeline of the data collected

Classrooms	Mary A	Kimberly	Mary B
Dates	3/16/22 to 4/6/22	4/1/22 to 4/8/22	5/16/22 to 6/6/22
Data collected	● Field notes taken on day 1, 2, 3, 4, 5, 6, 7, 12 out of 13 days ● Video recordings of house model simulations ● House model interviews ● Scratch interviews	● Field notes taken on day 5, 6, 7, 8, and 9 out of 9 days (No lesson observations in the first four lessons– scheduling conflict) ● Video recordings of simulations	● Field notes taken on day 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 out of 13 days ● Video recordings of simulations ● House model interviews ● Scratch interviews

		● House model interviews ● Scratch interviews	

The first classroom where I began collecting data was Mary's Classroom A, starting from lesson 1. Structured field notes (see Appendix) were taken in most of the lessons. There were lessons I could not take field notes such as lesson 6, where students were introduced to Scratch and started creating their Scratch programs, because I was interviewing students about their house models in those lessons. In Mary's Classroom A, students began constructing their house models during lesson 4, and these models were tested in a simulation during lesson 5. Immediately following the simulation, I started interviewing students about their house models and the results of the simulation. This was because simulations played a significant role in students’ house models and in what kind of revision they were going to propose. During the days I was conducting these interviews, I couldn't take field observations because I was focused on the interviews. Once the house model interviews were completed, I resumed my field observations for the remaining lessons. Later, as students finished creating their Scratch programs, I conducted another set of interviews with the same students about their Scratch projects if they were present. The procedure in Mary’s Classroom B mirrored this process closely. In Kimberly’s classroom, I followed a similar data collection approach. I took field notes in the lessons I attended and interviewed students about their house models after they had been simulated by the teacher. Then, as students completed their Scratch programs, I conducted interviews about those projects as well.

I would like to clarify why I could not take field notes on some days (Please refer to Table 4 above for details). No field notes were taken on days 8 and 9 Mary’s Classroom A because I was conducting interviews with students about their house models. In the same classroom, field notes were also not collected on days 10, 11, and 13 due to interviews with students about their Scratch programming projects. On these days, the students were engaged in working on their Scratch projects. Lesson observations were not conducted on days 11, 12, and 13 in Mary's Classroom B . This was again due to interviews being conducted about Scratch projects. During these sessions, the students were in the process of creating their Scratch programs. I was unable to attend the first four lessons in Kimberly's classroom due to scheduling conflicts, which resulted in no field observations being collected for those initial sessions. In summary, the collection of field notes was impeded on specific days due to the need to conduct interviews with students regarding their projects, and in one instance, scheduling conflicts prevented the observation of early lessons in a different classroom. In the following subheadings, I will discuss the data collection method, such as what an artifact interview is followed by interview protocols.

Artifact Interviews

The data collection method of the present study is artifact interviews. This method is consistent with tenets of constructionism. Brennan and Resnick (2012) define artifact-based interviews as interviewing one about an artifact such as a computational artifact in Scratch or a tangible artifact such as a house model that one created. The aim is to delve into the student's conceptual understanding of a topic, their thought processes, and the decisions they made during the creation of the artifact. The participants built house models and created computational artifacts in Scratch. Second graders were interviewed about house models and Scratch programs which they created during the integration of the module.

Ginsburg (1997) argues that clinical interviews are a powerful technique for gaining insight into a child's way of thinking. Ginsburg writes that during a clinical interview the interviewer asks open-ended questions such as “how did you do it”? or “why” [...] (p. 34). In comparison to clinical interviews and artifact interviews, both are geared towards understanding the student's comprehension of a specific topic or concept. In both types of interviews, open-ended questions are posed. These questions are not meant to elicit simple 'yes' or 'no' responses, but rather encourage students to expand upon their answers, providing insight into their thinking processes. The primary distinction between the two lies in the use of artifacts in artifact interviews. In an artifact interview, physical or digital items (artifacts) that the student has created are used as the basis or focal point of the conversation. In this approach, the interviewer engages in conversation with the child about their computational products and practices, using work samples to guide the conversation. These artifacts can be assignments, projects, drawings, or any other product of the student's work such as house models and Scratch programs. They serve as tangible evidence of the student's understanding and thinking. By discussing these artifacts, the interviewer can probe deeper into the student's thought processes, decisions, and the rationale behind the creation of the artifact. On the other hand, clinical interviews do not specifically rely on student-made artifacts to guide the conversation. In summary, while both clinical and artifact interviews aim to delve into a student's understanding and thinking processes, artifact interviews have the added dimension of analyzing and discussing actual work products made by the student. This can offer additional insights into their knowledge and skills of a topic. The artifact interview method was convenient for this study because most of the activities students were involved with in the module were meant to create artifacts.

Interview Protocols

In this study, I employed two distinct interview protocols: one for house model interviews and another for Scratch interviews (Appendix). Both protocols are semi-structured and include a set of probing and follow-up questions tailored to each specific type of interview. In the following two paragraphs, I will discuss how I developed the questions for each protocol and how they were targeted to particular elements of CT practices in house model interview protocol and of CT concepts and practices in Scratch interview protocol.

For the house model interview protocol, I designed specific questions to target CT practices. For example, with the question of *What did you notice when your teacher tested/simulated your house model?* I aimed to expose students' understanding of CT practice of testing. To reveal students' understanding of CT practice of debugging, I asked *What material(s) would you change in your model and why?*

In the interview protocol designed for Scratch projects, I formulated specific questions to uncover students' comprehension of computational thinking (CT) concepts and practices. With regards to CT concepts, to assess their understanding of the CT concept termed “events”, I posed the question, “*Can you show me how to start your project?*” When focusing on the 'sequences' aspect of CT, I asked, “Can you tell me why you put these coding blocks in this order?” As for CT practices in Scratch interview protocol, the question “*Can you tell me how you created this script?*” was crafted to delve into their grasp of the incremental and iterative development process. By asking “*Was there a time when your project didn’t run as you wanted?*” I aimed to gauge their experiences and understanding of testing and debugging. Lastly, with “*Why did you choose these sprites?*”I sought insights into their knowledge of abstraction and modularization.

Data Analysis

The main data sources of the current dissertation are house model interviews and Scratch interviews. After transcribing the 30 interviews, I was left with a substantial text-based dataset. To analyze these student responses, I employed Brennan and Resnick's 2012 three-dimensional Computational Thinking (CT) framework. This framework focuses on CT concepts, CT practices, and CT perspectives. All these three dimensions were applied to Scratch interviews data while only the CT practices dimension was applied to the house model interviews data. Before discussing how I analyzed the house model interviews and Scratch interviews, please refer to Table 1 in chapter two, which is the Brennan and Resnick's (2012) CT framework I adopted in this dissertation study to analyze the data.

Please note that a statement in the interviews could get multiple codes and only utterances related to the codebook received a code, meaning that parts of interviews that did not reflect CT were not coded. And, an average of 20% of the total number of codes in each house model interview and each Scratch interview was not coded, as those did not reflect CT. In the following subheading, I will discuss the analysis of house model interviews.

Analysis of House Model Interviews

In analyzing the house model interviews, only CT practices were employed as coding criteria from the CT framework as well as a data driven code. However, both CT concepts and CT practices were used to analyze the Scratch interviews. It's noteworthy that CT perspectives were not applied in the analysis of either interview type. This omission stems from the module's design which did not offer opportunities for students to be engaged in CT perspectives. For example, neither the house models nor the Scratch programs provided avenues for students to showcase self-expression. However, all three dimensions and their components mentioned above constituted the codebook in data coding. In the following paragraph, I will discuss how I applied both a theory-driven approach, which is the Brennan and Resnick (2012) CT framework, and a data-driven approach to code the house model interviews and Scratch interviews.

In a nursing-focused article, Fereday and Muir-Cochrane (2006) employed a dual method of thematic analysis, combining inductive (data-driven coding) and deductive (theory-driven coding) techniques. This dissertation similarly adopted that combined analytical approach for house model interviews, as it used both theory-driven codes and data-driven codes. For the house model interviews, I initially used a mix of "data-driven" and "theory-driven" codes. The theory-driven codes for the house model interviews included "simulation" related to the CT practice of testing and "proposing a revision" linked to debugging. Data-driven codes addressed students' understanding of natural disasters and material choices. However, after reviewing a few of initial coded interviews and consulting a committee member, I realized two of these data-driven codes could also be categorized as theory-driven, aligning well with the existing framework. For example, material choice fit into "being incremental and iterative", as they built their house models in an iterative and step-by-step fashion, and planning resonated with "decomposition", as students broke the activities in the module into three manageable components such as planning a house model, testing it, and proposing a revision to it. Reusing and remixing is the only CT practice from the CT framework that wasn't employed in the data analysis. This practice is specifically tailored to the Scratch environment, rendering it unlikely to emerge in house model construction. As for students’ understanding of natural disasters, it correlated with data collection as a practice. Data collection, a CT practice not mentioned in the framework, is included as a data-driven code, and I elaborate on implications in the discussion chapter. As a result, I used both theory-driven codes, which are CT practices mentioned above, drawn from Brennan and Resnick's (2012) CT framework and a data-driven code, which is data collection, for analyzing the house model interviews.

For the analysis of house model interviews, the average number of codes for a group of participants was around 87 per interview after subtracting the codes that did not reflect CT. This means that each theme (five themes in total in house model interviews) for a group of participants was an average of 17 codes (87 divided by 5). At this point, I would like to state that for a code to be a theme in house model data, I followed a rule that the average number of codes per a group, which is 17, needed to be seen across the majority of the participants, which is 8 groups in this case. These figures were a bit lower for the theme of “Abstraction and Decomposition in Planning a House Model”, as it did not emerge as much as other themes. This could be because the CT practice of decomposition was found based on poster boards data mostly, not house model interviews. In other words, students were not inquired enough about their poster boards, which is a methodological limitation and it will be discussed in limitations. In the following two paragraphs, I will discuss how I coded house model interviews by using CT practices defined by Brennan and Resnick (2012).

During the interviews, especially house model interviews, students often discussed their material choices they made in creating their house models and proposing a revision because materials were at the center of participants’ discussions. This is understandable in a context where participants planned what materials to use for their house model, they built their houses with those materials, that most of the problems with their houses during simulation stemmed from materials such as that their roof or walls dislodged or carried away during simulation, and that they proposed a revision to the problems that materials caused during simulation.

Consequently, materials assumed a central role in the activity that the participants were engaged with. In the following paragraph, I will provide an example to show how I coded students' material choice-related statements as the CT practice of being incremental and iterative, their simulation-related statements by using the CT practice of testing, and their revision-related statements as debugging.

When one pair, Iris and Travon, discussed their observations of what happened to their house model during the simulation and how they proposed a revision to their house model, they stated the following:

> Our house jumped because the wall [index cards] was [were] not sturdy. We used index cards. And the index cards are not the strongest materials. Since it's weak and there's a huge amount of wind, and there's a lot of water, the water could dampen and then it caved in [...] We learned that paper is definitely not the strongest one it could get wet. The next time we could put tape all around the base and then see if it works. We'll definitely check if there's any holes. I'm not going to ignore the holes cause that's what I did. Actually I'll put more glue, more tape, make sure it's stable. Give myself a mini test. [...] We would use legos, cardboard, playdough, and popsicle sticks because they are very strong. Instead of paper for the walls, we would use cardboard because cardboard is strong material. And paper was more likely to fall over.
>
> (House model interview on 3/25/2022 and Scratch interview on 4/3/2022)

When coding these statements, I coded their simulation-related statements using the CT practice of testing such as "our house jumped" or "there's a huge amount of wind, and there's a lot of water" because they are referring to what happened to their house model during the simulation. I coded their material choice-related statements such as "We used index cards. And the index cards are not the strongest materials" or "We learned that paper is definitely not the strongest one it could get wet." as being incremental and iterative, as the materials they selected did not work as they expected and they noticed this and decided to change the materials they used in their first trial. Iris and Travon's experience during the hurricane simulation highlighted the need for stronger materials and careful building, showing the value of learning through repeated trials, which suggests an iterative process. As for debugging, I coded their revision-related statements such as “The next time we could put tape all around the base and then see if it works” or “Instead of paper for the walls, we would use cardboard because cardboard is strong material” as debugging, as those statements refer to revisions they proposed, which I thought corresponds to the CT practice of debugging in the framework. As a result, the CT practices of being incremental and iterative, testing, and debugging were suitable to code students house model interviews. Please note that I initially considered revision-related statements as benign incremental and iterative. However, I noticed that debugging is a powerful, stand-alone CT practice so I decided to code revision-related statements as debugging for the sake of precise analysis. As discussed before, collecting data was a practice students developed before building their house models but it was not part of the chosen framework so I decided to use a data-driven approach to code students’ natural disasters research-related statements. In the house model interviews, when asked about their research, students provided substantial input. Statements like “Hurricanes can be caused in Florida mostly. Yeah, mostly in Florida. So they can cause a lots of damage to cars. Such as trees, cars and houses.” or “A hurricane causes people to be homeless and have no houses.” were coded under “data collection” to reflect their research-driven insights.

Analysis of Scratch Interviews

As opposed to using only CT practices of the framework to code the house models interviews, I used all components of the framework to code the Scratch interviews: CT concepts, CT practices, and CT perspectives. As mentioned before, CT perspectives did not emerge from the Scratch data, as the module's design did not offer opportunities for students to be engaged in CT perspectives. Although the codebook encompassed all CT concepts and CT practices, certain CT concepts like loops, parallelism, and operators weren't evident in the data. This gap can be attributed to the module's design; students were directed to design a basic Scratch program, which didn't necessitate the use of these specific concepts. Regarding CT practices, three prominent practices - abstraction and modularization, adopting an incremental and iterative approach, and testing and debugging - were discernible from the Scratch data. The lesson's structure focused on having students create simple Scratch programs using basic elements, thereby creating a context where reusing and remixing were not necessary components. The CT practice of reusing and remixing did not emerge from the Scratch data.

Before coding the data, I converted the theory-driven codes into a codebook. This codebook included all CT concepts in Brennan and Resnick (2012) CT framework like sequences and data, all CT practices such as abstraction and decomposition, and all CT perspectives like connecting. In the first column of the codebook, I listed the code names like “CT concepts”. The second column contained definitions for each of these CT concepts, practices, and perspectives defined by Brennan and Resnick (2012). I also included a couple of examples and non-examples for each. After completing the codebook, I applied only CT practices to house model interviews and all three CT dimensions in the framework (CT concepts, CT practices, and CT perspectives) to Scratch interviews.

In the analysis of Scratch data, there were around 800 codes in total in Scratch interviews. This figure was divided by the total number of seven themes (four CT concepts include sequences, events, conditionals, data, and three CT practices include abstraction and modularization, being incremental and iterative, and testing and debugging). In other words, 800 was divided by 7 themes and it was found a total number 110 codes per theme on average. These 110 codes were divided by the number of groups, that is, 110 was divided by 15 groups. Each group took a total of around 7 codes per theme. At this point, I would like to state that to qualify a theme as official, it needed to meet a specific criterion: this average of 7 codes had to be consistently present across the majority of the participant groups. Specifically, a theme would be considered official only if it appeared in at least eight out of the fifteen house model interviews, maintaining this average of 7 codes for each group of participants. In the following paragraph, I will discuss how CT concepts in Brennan and Resnick's CT framework were applied to Scratch interviews as an example.

Students used different types of blocks in their Scratch programs and they programmed their projects in such a way that they paid attention to the order of blocks used. Statements like "the event is supposed to go on the top, and then if you're doing a recording, the recording is supposed to go in the middle. And then the final thing, whatever you wanna do, sensing, control, sound, looks, whatever you wanna do, you can put that at the bottom because it might work for you." were coded under the CT concept of sequences. Remarks like "When you press a certain button [key] on the keyboard, it plays that certain thing" or "That's [the *when right arrow key pressed* block] what starts the Makey Makey [event block]" were coded under the CT concept of events. Statements such as "[...] if we do [code] all left arrows then it will all play at the same time." or "If we pressed [had] the same button [for all three sprites], all of them would be going on." were coded under the CT concept of conditionals. Statements like "We used cardboard, popsicle sticks, tinfoil, cotton balls, paper towel, pipe cleaners, tape, and index cards. We used these materials because they are strong", which were entered into the sound blocks, were coded under the CT concept of data. This means that when students played one of their sound blocks in their Scratch program, the CT concept was applied, as the sound blocks were the blocks into which students recorded their data such as what materials used, simulation results, and revisions.

In the following paragraph, I will discuss how I analyzed the Scratch data by using the CT practices as an example.

With regard to coding the Scratch interviews using the codes of CT practices, during the Scratch interviews, when asked about using three sprites for their project, some examples of students' responses were as follows: "Because we would get mixed up. Which one's one, which one's two, which one's three. [...] That's why we separate" or "We used three different sprites because there were three boxes in the poster. So we only do three [sprites]" or "We will probably need three blocks to record everything." These kinds of statements were coded under the CT practice of abstraction and modularization, as those suggest that students knew they needed to decompose their Scratch program so they could mirror each data set on each column on their poster board. Remarks such as "[...] we pressed this [My Blocks], and then we went to the Makey Makey and then pressed this [a Makey Makey event block]. Then we got these [Makey Makey blocks came up on Scratch and they added one of the Makey Makey event blocks]" or "We recorded our data into a sound block. So we dragged a Makey Makey first. And then changed it to the [one of the arrow options under Makey Makey event block" were coded under the CT practice of being incremental and iterative, as those indicate that students built their Scratch programs in an iterative and step-by-step manner. Statements like "We click space. But also if we did that [space key] for every single one [script], then it would all talk over each other. That's why you need the [different] Makey Makey [blocks for each script]", if they [event blocks] are not different, they would talk and talk over each other", "If we pressed [had] the same button [for all three sprites], all of them would be going on" were coded under the CT practice of testing and debugging because those remarks suggest that students were aware that assigning the same button to multiple scripts would result in concurrent activation, disrupting the intended sequential flow and individual functionality of each script within the program.

One final word is that the sequence of activities in this study, beginning with the tangible house model activity and progressing to the more abstract Scratch work, was a critical factor in enhancing the learning experience for second graders. Starting with the tangible, unplugged task of building a house model provided a concrete foundation, making it easier for young students to grasp basic concepts. This approach aligns perfectly with constructionist principles, which emphasize learning through direct, experiential engagement. Transitioning from this physical activity to the digital realm of Scratch, a relatively more complex and abstract platform, allowed students to build on their initial understanding and apply it in a new context. This progression from simpler, hands-on tasks to more challenging, abstract ones not only facilitated a smoother learning curve but also likely deepened students' comprehension and retention of key concepts. Therefore, I think that the deliberate structuring of these activities, moving from the tangible to the abstract, played a pivotal role in supporting the cognitive development of second graders in this study.

Analysis of Field Notes

In the analysis of field notes, I considered all components of the framework to code the field notes data: CT concepts, CT practices, and CT perspectives. CT perspectives were not applied to the field notes. This omission stems from the module's design which did not offer opportunities for students to be engaged in CT perspectives. In the following paragraph, I will provide an example of how I analyzed the field notes by using a student’s tornado presentation with three chairs in the classroom where I think they emphasized the CT practices of being iterative and testing.

During the first lesson in Classroom B, Mary covered the topics of modeling and simulation, discussing their use by scientists for data collection. After presenting these ideas, Mary sought feedback on the class's understanding. Charlotte then stepped up to show her grasp of the concepts. Using three chairs from the room, she gave a presentation. In it, she depicted a tornado's appearance, differentiated between strong and weak buildings, and explained how scientists gather information from simulations.

> Charlotte pretended that three chairs in the classroom represented three houses: one made of hay, another of wood, and the third of brick or cement. She started acting like a tornado by opening her arms, spinning around, and moving towards the three chairs/”houses”. She first hit the wooden house with her arms and tipped it over. Then she hit the hay house and knocked it down. Lastly, she hit the brick one and pretended that she could not tip it over. At the end, she said that we needed to write down that a brick or cement house is strong against tornadoes. (FN 05.16.2022)

Charlotte skillfully used three chairs to symbolize houses constructed from hay, wood, and brick or cement, effectively showing the effects of a tornado on each type of structure. She enacted the scenario, demonstrating vividly the response of each house type to the simulated tornado. Her performance showed that houses made of wood and hay were easily toppled and damaged by the tornado, highlighting their vulnerability. Conversely, she indicated that the brick or cement house remained unshaken, portraying its resilience and ability to withstand the tornado's force. Through this demonstration, Charlotte effectively communicated her comprehension of the significance of testing and experimenting, two CT practices in the framework, to determine robust building materials for safeguarding against natural calamities.

Please note that it's important to recognize the invaluable role of field notes taken during lesson enactments. The field notes serve as a robust medium for capturing these interactions, detailing students’ dialogues with their teachers and peers. These notes are not merely supplementary but are essential in unveiling how students’ CT knowledge emerged during the lesson.

Trustworthiness

Trustworthiness or credibility is concerned with whether the results of a study are reliable. Merriam and Tisdell (2015) state all research is done to produce valid and reliable knowledge ethically. But how can we know whether results of a research are trustworthy? In the following two paragraphs, I will discuss how I established trustworthiness through two ways: Peer review and triangulation respectively.

Merriam and Tisdell (2015) highlight the role of peer review in establishing the credibility of a study. According to them, peer review involves discussions with colleagues about the study process, data, and preliminary interpretations. I ensured peer review in two ways to bolster the credibility of my work. First, I regularly consulted with my dissertation advisor during the coding process. In our meetings, we discussed coding examples until we reached a consensus. Second, I took part in a research group led by my advisor. This group, comprised mainly of graduate students, allowed me to clarify ambiguous coding instances. I even presented to the group on several occasions, during which we collectively coded my data.

To ensure the trustworthiness and validity of this study, I employed triangulation as recommended by Merriam and Tisdell (2015), who assert, "Probably the best-known strategy to shore up the internal validity of a study is what is known as triangulation" (p. 245). In line with Denzin's (1978) four types of triangulation—multiple methods, multiple data sources, multiple investigators, and multiple theories—I incorporated multiple sources of data and multiple investigators. Specifically, the research group and my advisor assisted with coding, thus serving as multiple investigators. The data was further triangulated using diverse sources, including students' utterances during artifact interviews, screenshots of those artifacts, and field notes taken during the module implementations. This approach aims to confirm the emerging findings and enhance the study's credibility. In the following chapter, dissertation results will be discussed.

CHAPTER IV

RESULTS

The results chapter is organized in the following ways: First, I address each of the research questions (RQ1 and its sub-questions to RQ2) sequentially. Second, to represent all classrooms studied, I chose examples strategically from the three different classrooms—Mary's Classroom A, Mary's Classroom B, and Kimberly's classroom—wherever possible, to exemplify how CT practices came to life in the context of the lesson. Third, the primary data sources for answering the research questions are interviews about the house models and Scratch programs. Additional materials include field notes, photographs of classroom artifacts such as house models and poster boards. One exception to this structure is in answering the part of RQ1 related to modeling (Please refer to Table 3 in the methods chapter to see what data sources were used to address which research question). In investigating RQ1, which delves into CT practices in students' comprehension of modeling and simulation, it's crucial to recognize the invaluable role of field notes taken during lesson enactments because the design of the curriculum itself facilitated this exploration. Each lesson within the module was crafted to emphasize a distinct concept, with modeling being the focal point of lesson 1. Consequently, students' engagement with the idea of modeling during this lesson was anticipated. The field notes serve as a robust medium for capturing these interactions, detailing students' dialogues with their teachers and peers. These notes are not merely supplementary but are essential in unveiling how the practice of modeling emerged during the lesson. This is why the field notes were foundational in addressing the modeling aspect of RQ1.

Research questions include:

- RQ1: Which CT practices did second graders develop through the engagement with the CT integrated science and engineering module while planning, building, testing, and proposing a revision to their house models, and how did they emerge?
    - RQ1a: How do 2nd graders describe their understanding of modeling and simulation?
    - RQ1b: What CT practices did 2nd graders develop in planning their house models?
    - RQ1c: To what extent did participants' initial house models survive from the simulation/test?
    - RQ1d: Which CT practices did second graders engage in describing and understanding their material choices they made in creating their house models and proposing a revision?
- RQ2: How do they describe in Scratch the process they undertook?

In this dissertation, I explore how second graders engaged with Computational Thinking (CT) while planning, building, testing, and proposing a revision to house models for a natural disaster scenario and creating Scratch programs.

Based on house model interviews and Scratch interviews, classroom observations, and the artifacts created during lessons, I found that second-grade students developed various computational thinking CT concepts and practices. While planning and constructing models of houses, as well as discussing concepts like modeling and simulation, the students demonstrated a range of computational thinking practices. Additionally, they developed both computational thinking concepts and practices when creating their Scratch programs. In the following paragraph, I will briefly state which CT practices they developed while planning, building, simulating, and proposing a revision to their house models. CT concepts and CT practices developed through Scratch will be discussed later.

Students used abstraction and decomposition to break big ideas into smaller parts. They worked step-by-step and made changes when needed, which is called being incremental and iterative. And they were engaged in testing and debugging/simulating, which involved evaluating their designs and identifying areas of improvement. The following sections illuminate these practices in greater detail, drawing from the rich conversations during the house model interviews, from field notes and artifacts such as Scratch programs, poster board, and from Scratch interviews, in which participants occasionally discussed their house model processes.

Computational Thinking Practices in Planning, Building, Testing, and Proposing a Revision to Students' House Models (RQ1)

Below is Table 5 showing frequency of CT practices emerged from house model data, which is house model interviews, field notes and photographs of classroom artifacts such as poster boards. CT practices is the broadest theme of this data set. Specifically, the table shows group number and participants in the first two columns on the left, and CT practices in the first row. Under CT practices are data collection, being incremental and iterative, abstraction and decomposition, testing, and debugging. As for figures in the cells, those represent the frequency of sub themes for a group of students that emerged from the data. In the last two rows are the total number of codes by theme and their percentages.

Table 5: Frequency table of CT house model interviews and Scratch interviews

		CT Practices in House Model Data				
Group number	Participants	Data Collection on Natural Disasters	Being Incremental and Iterative in Material Choices	Abstraction and Decomposition in Planning a House Model	Simulation/Testing	Proposing a Revision/Debugging
1	Iris & Travon	**20**	**47**	4	**23**	**17**
2	Alicia & Gabriela	12	**40**	4	**17**	8
3	Nico & Rafael	**22**	**52**	10	14	**18**
4	Tristan & Mariana	13	**50**	5	**19**	**17**
5	Violet & Javier	**22**	**52**	6	**25**	10
6	Luis & Ian	**28**	**55**	7	13	**19**
7	Charlotte & Sofia	**21**	**76**	13	**35**	**25**
8	Roberto & Sara	**30**	**49**	11	16	8
9	Amelia & Selena & Ana	**24**	**25**	8	**27**	**17**
10	Diego & Caleb	**21**	**81**	10	**45**	**22**
11	Alice	8	**33**	9	**18**	9
12	Carlos & Eduardo	14	**40**	14	**24**	**19**
13	Julia & Jorge	12	**24**	12	**19**	10
14	Nathan	**20**	**54**	8	**18**	**25**
15	Samuel	11	**43**	7	**17**	**17**
	Total number of codes by theme	**278**	**721**	128	**330**	**241**
	Percentages of total codes	**16%**	**42%**	8%	**20%**	**14%**

As mentioned in Chapter 3, house models frequently were mentioned in Scratch interviews. All house model-related codes during Scratch interviews and total number of codes from house model interviews were put together in Table 5 above.

In Table 5, figures bolded represent the number of codes higher than the average, which is 17 per theme for each group of participants (Please refer to the methods chapter for details), and the regular ones (no bold) represent the number of codes lower than the average. Additionally, total numbers in the bold in the last row are the official themes that emerged from the data. The regular one, which is abstraction and decomposition, is still considered an official theme and will be discussed below. From this data, I can deduce that the CT practices of data collection on natural disasters, incremental and iterative approaches in material selection, simulation/testing emerged, and debugging/proposing a revision as prominent themes. However, the theme “Abstraction and Decomposition in Planning a House Model” had fewer codes than the average, indicating its lesser prominence. This might be due to the reliance on poster board data mostly, rather than house model interviews, for identifying decomposition as a CT practice in house model interview data, representing a methodological limitation, as there was insufficient inquiry about the poster boards during the house model interviews, a point that will be explored further in the limitations section. Furthermore, the CT practice of being incremental and iterative stood out significantly, largely because of the frequent discussions about materials in the house model interviews. This emphasis on materials aligns with their crucial role in the project, as participants chose materials, faced challenges during simulations, and proposed solutions to these material-based issues. Consequently, this made the theme of material discussion the most recurrent one in the data.

Below is Table 6 showing example codes for each CT practice emerged from the house model interview data, types of codes such as theory-driven or data-driven, code names.

Table 6: Sample codes for CT Practices emerged from house model interviews

CT Practices	**Code Types**	**Codes**	**Example Codes**
Computational Thinking Practices in House Models	Theory-Driven Codes	Being Incremental and Iterative	Int: What happened to your first model that you needed to start it over? Charlotte and Sofia [at the same time]: The Cardboard walls wouldn't stay. Sofia: The glue wasn't dry. And we didn't have enough time, like we start[ed] the walls. So we just started on a brand new house. Charlotte: And they [cardboards] kept falling. [...] The first one was a total disaster. Int: So you said that you couldn't glue [the cardboard to] the construction paper. Charlotte: Yeah. Cause the cardboard kept falling apart. Sofia: And Charlotte kept saying that the tape isn’t gonna be strong enough Charlotte: And neither was the glue. <laugh> I think we kind of tried both.

		Abstraction and Decomposition	Int: Why do you have three sections on your poster? You're looking at your poster? Roberto: Yeah Sara: Cause there's three sections [on our poster]. Int: Three sections? What do you mean by that? Sara: Because they show what we use, one thing we noticed, one thing we would change.
		Testing/Simulating	Int: How would you define simulation? Alice:I would define it as a test of how strong we did. How strong would the house be if a hurricane hit and a tornado hit. — Int: what type of simulation did your teacher do? Eduardo: She did a fan and two fans, one big fan, one small fan. Int: Mm-hmm Eduardo: [...] she was testing to see whose [house model] stand[s] up the most.
		Debugging/Revising	Int: What would you change in your house model? Caleb: Using the data from the simulation, I would change the roof to construction paper. — Int: What would you do differently in your house model? Nathan: I would make a couple of these [popsicle sticks-the triangle shape structure he made before] around the house. Int: To do what? Nathan: To make it extra protected. Int: What would you do the same? What would you not change? Nathan: I would not change the alarm system. Int: Why? Nathan:Because it would be good for natural disasters because, well hurricanes you could see in the sky, but most other disasters you can't see. So this would be good. It would alarm all the people.
	A Data-Driven Code	Data Collection	Int: What natural disaster did you research? Iris: Well, we researched floods first. Int: Mm-hmm. Iris: And then we started researching hurricanes. Int: What did you learn about hurricanes? Iris: What I learned about hurricanes that they can cause an entire house to break on the ground cuz they have very, very strong ones. Some winds can go up to 40 miles per hour. — Int: What did you learn about hurricanes? Julia: We learned that it break[s] things.

			Int: What did you learn about hurricanes? Jorge: Hurricanes can be caused in Florida mostly. Yeah, mostly in Florida. So they can cause a lots of damage to cars. Such as trees, cars and houses. Int: What are the consequences of a hurricane? Julia: A hurricane causes people to be homeless and have no houses.

Next, I'll outline the CT practices participants engaged in while planning, constructing, testing, and proposing a revision to their house models.

Computational Thinking Practices in Students' Understanding of Modeling and Simulation

RQ1a asks, how 2nd graders describe their understanding of modeling and simulation. During the course of the study, I took field notes during lesson observations in three classrooms: two in Mary's school and one in Kimberly's school, and conducted interviews (both house model interviews and Scratch interviews). The concepts of modeling and simulation are central to the module they were studying. This focus aligns with two key module standards: 1) Describe how models represent a real-life system (e.g., globe, map, solar system, digital elevation model, weather map); and 2) Define simulation and identify the concepts illustrated by a simple simulation (e.g., growth and health, butterfly life cycle). Below, I discuss the structure of the following section where I answer RQ1 and why I structured it in that way.

I included examples of modeling and simulation from students in the three classrooms such as examples of abstraction and decomposition practices from students in Mary's classroom A, Mary's classroom B, and Kimberly's classroom. In investigating RQ1, the field notes serve as a robust tool for capturing these interactions, detailing students' dialogues with their teachers and peers. This is why the field notes were foundational in addressing the modeling aspect of RQ1.

Simulation part of RQ1 was addressed through house model interviews, Scratch interviews, and field notes (Please refer to Table 3 in the methods section to see what data sources were used to address which research question). Since students discussed the revision they proposed and even discussed the concept of simulation during their Scratch interviews, readers could find me referring to Scratch interviews while discussing their proposed revisions or simulations. This means that data of Scratch interviews are not only about students' Scratch programs but also about their house models. Subsequent sections will detail participants' insights on modeling and then on simulation with regard to CT practices from the three different classrooms through conversations in house model interviews and Scratch interviews as well as field notes I took during lessons.

CT Practice of Abstraction in Students' Understanding of Modeling. Abstracting and modularizing involve creating larger systems or projects by assembling a series of smaller, more manageable components (Brennan and Resnick, 2012). I will be drawing upon my field notes from Lesson 1, 4, and 5 to address this heading because these lessons are the lessons in which teachers not only taught about but also reviewed the concept of modeling, and dialogues between the teachers and students reflect CT practices in students' understanding of modeling.

Mary, the teacher who implemented the module in her classrooms A and B, brought a globe to her classroom A in lesson 1. She asked the class if the globe is real. Gabriela responded "The globe represents the Earth. It is a model of the Earth" (Field Notes 03/16/22). Upon this, Mary [nodding her head as a form of approval] asked if we could see the real globe. Rafael replied that "We cannot see it but we can draw one." (FN 03/16/22). Gabriela's response about the globe representing the Earth directly connects with the CT practice of abstraction. She's simplifying the real Earth into a manageable representation, a key aspect of abstraction in CT.

Additionally, Rafael's response, "We cannot see it but we can draw one," (FN 03/16/22) can also be linked to the abstraction CT practice. This is because he understands that while the real globe isn't physically visible, it can still be represented through drawing, showing his ability to work with abstract concepts and represent them in a different form. In other words, his response highlights the connection between the abstract idea of the globe and its representation through drawing.

During lesson 4 in Mary’s classroom A, she was reviewing the concepts such as modeling she taught in previous lessons. She asked why we make models. Working with Alicia during the project, Gabriela responded “to show people what the real one looks like. For example, a toy car is a model of the real car. We make a model of a real car to show what the real one looks like.” (FN 3/21/22) Upon this, Mary replied "I could not make a better explanation than that.” (FN 3/21/22). Right after Gabriela, Tristan also responded that “The clock [in the classroom] is a model that represents the time” (FN 3/21/22). Gabriela's idea of models as representations directly connects with the practice of abstraction in CT. She compares toy cars to real ones, just like abstraction simplifies complex ideas. Tristan's mention of the classroom clock as a model for time showcases an understanding of how models simplify real concepts. To conclude, both Gabriela's and Tristan's insights highlight the second graders' innate understanding of the CT practice of abstraction, illustrating their ability to simplify and represent complex systems through models.

In lesson 1 in Mary’s classroom B, after defining models, Mary asked the class why they think scientists make models. Sofia said “to show an example of what they are studying.” (FN 05/16/22) Sara responded “to show us what things look like.” (FN 05/16/22) Diego’s response was “to show us the color and shape of real things.” (FN 05/16/22). After those replies, Mary approved all those responses and added that “Scientists create models for various purposes, and the ones you've shared are just a few examples among many.” (FN 05/16/22) The students' responses to why scientists make models align closely with the abstraction CT practice. Sofia's response reflects her understanding that models are abstract representations that exemplify scientific concepts. Sara' response demonstrates her grasp of abstraction by recognizing that models visually represent real objects or phenomena. Diego's reply emphasizes abstraction as he identifies how models capture specific attributes of real objects. The students' responses underscore their grasp of models as tools of abstraction to represent real-world objects and phenomena. Mary's follow-up comment reinforces this understanding, emphasizing that scientists use models for diverse purposes, further highlighting the broader application of the abstraction CT practice in scientific exploration. In the following paragraph, students’ understanding of modeling and how it is linked to abstraction and decomposition will be discussed through the field notes that I took in Kimberly’s classroom during lesson 1.

Kimberly, the other teacher who implemented the module in one of her classrooms at a different school, also brought a globe to the classroom to review the concepts of modeling and simulation in lesson 5. After letting kids know that they would start building their house models in this lesson, Kimberly asked the class what a model is. Julia responded “How you show something” (FN 04/01/22). After Julia, Carlos said “You do math and science with a model” (FN 04/01/22). Samuel said: “You show stuff through a model” (FN 04/01/22). The teacher reacted “these are great responses" (FN 04/01/22) and then showed and read aloud the definition of a model on the screen, “A model is a copy of a real life item or system that people use to get information or learn from” (FN 04/01/22). After that, she was holding the globe in her hands and asked what it was. Jorge said “the globe” (FN 04/01/22). Nathan replied “a model” (FN

04/01/22). The teacher asked, “a model of what?”(FN 04/01/22) Eduardo said “the Earth” (FN 04/01/22). And then, the teacher asked the class the following: “What kind of information would you get from this model?” (FN 04/01/22) Carlos said “places to go” (FN 04/01/22). Jorge said “Florida” (FN 04/01/22). The teacher asked “What do colors on the globe represent?” (FN 04/01/22) Julia replied “continents” (FN 04/01/22) and Eduardo said “countries” (FN 04/01/22). After this interaction, this time the teacher was holding a bigger globe, a globe that was designed to show the mountains especially. She asked the class “what do these bumps represent?” (FN 04/01/22) Eduardo responded “mountains” (FN 04/01/22). The teacher asked “Do these globes give us information?” (FN 04/01/22) Nathan said “it shows us the north pole and the south pole”. (FN 04/01/22). In the conversation above, the teacher effectively integrates both the abstraction and decomposition CT practices with the science and engineering lesson. When Kimberly asked the class what a model is, Julia, Carlos, and Samuel's responses reflect abstraction, as they simplify the concept of a model to "how you show something" or "showing stuff through a model." This highlights the practice of abstraction in breaking down complex ideas into simplified representations. Furthermore, when Jorge and Nathan identify the globe as both "the globe" and "a model," they demonstrate the decomposition practice by recognizing the globe as a representation of a real object. The teacher's use of a larger globe to emphasize mountains and asking about the bumps representing mountains aligns with the decomposition practice. Eduardo's response of "mountains" illustrates the practice of breaking down a complex system (the globe) into its specific components (mountains). Moreover, when the teacher asks about the information the globe provides, Carlos, Jorge, Julia, and Eduardo's responses showcase the concept of abstraction as they identify "places to go," "Florida," "continents," and "countries" respectively, as information gleaned from the model.

Across the three classrooms, students consistently exhibited a strong grasp of abstraction and decomposition when interacting with models. Their intuitive understanding, reinforced by Mary and Kimberly's teachings, illuminates second graders' capacity for computational thinking within the context of modeling.

CT Practice of Testing in Students’ Understanding of Simulation. CT practice of testing involves running the artifact through various scenarios and using specific test cases to identify any errors, bugs, or unexpected outcomes (Brennan and Resnick, 2012). I will be drawing upon data from field notes as well as interviews to construct a description of CT practice of testing in this part (Please refer to Table 3 in the methods chapter to see what data sources were used to address which research question).

Students’ grasp of simulation corresponds to the CT practice of testing. In Lesson 1 in Mary’s classroom B, she, Mary, explained the concepts of modeling and simulation and how scientists use simulations to gather data. Following her definitions, she prompted the class for comprehension feedback, at which point Charlotte volunteered to demonstrate her understanding through a presentation using three chairs in the classroom. In her presentation, she illustrated what a tornado looks like, which types of buildings are strong or weak, and how scientists collect data from simulations.

> Charlotte pretended that three chairs in the classroom represented three houses: one made of hay, another of wood, and the third of brick or cement. She started acting like a tornado by opening her arms, spinning around, and moving towards the three chairs/”houses”. She first hit the wooden house with her arms and tipped it over. Then she hit the hay house and knocked it down. Lastly, she hit the brick one and pretended that she could not tip it over. At the end, she said that we needed to write down that a brick or cement house is strong against tornadoes. (FN 05.16.2022)

Charlotte effectively demonstrated the impact of a tornado on different types of houses by using three chairs to represent houses made of hay, wood, and brick/cement. By physically acting out the scenario, she vividly illustrated how each type of house reacted when hit by her "tornado" representation. Through her demonstration, Charlotte conveyed that wood and hay houses were vulnerable to tornadoes as they were easily tipped over and knocked down. On the other hand, she pretended that the brick or cement house could not be tipped over, suggesting that it was more resilient and stood strong against the force of the tornado.

During their house model interview, Charlotte and Sofia made similar statements to Charlotte’s statements she made during her hurricane illustration discussed above.

> Interviewer: How can we protect ourselves against natural disasters?
> Charlotte: I think we could maybe test out different things [materials] to try to build the strongest house. Like we did [built a house model and tested it] and whichever one [material] is the strongest, we should probably build a lot of [it].
> Sofia: So I kind of had the same thinking as Charlotte, kind of like our house, we had to test it and our house stood the tornado [simulation]. So that would be a good house to build there. There hasn't been a real house that's actually made out of cardboard and paper and things.
>
> (Group interview 06/02/22)

Charlotte and Sofia's responses during the house model interview demonstrate their understanding of the importance of testing and experimentation to identify strong building materials for protection against natural disasters. By learning from simulations and acknowledging real-world limitations, they show a balanced understanding of the complexities involved. Their preference for practical materials underscores the need for feasibility in construction choices. Their responses reveal a thoughtful consideration of safety and a desire to use effective strategies for safeguarding against natural disasters. This suggests that the simulation activity not only engaged the students but also effectively illustrated the criticality of testing various materials, a key aspect of computational thinking, demonstrating how such testing can lead to the construction of stronger, more disaster-resistant houses.

In Kimberly’s classroom, Eduardo worked with Carlos throughout the project. In their house model interview, Eduardo discussed why their teacher simulated the house models and used a spray bottle and fans during the simulation.

> She was testing to see whose [house model] stood up the most. And some people won. Some people lost. And what happened was she sprayed water on our house models. Because that was the water from a hurricane. And the wind was the wind from a hurricane because it has a really windy thing.
>
> (Group interview 04/06/22)

In the house model interview, Eduardo explained that their teacher conducted a simulation to test the strength of their house models against hurricanes. The teacher utilized a spray bottle to represent hurricane water and fans to imitate powerful hurricane winds. As Eduardo stated some models withstood the test, while others did not, allowing Eduardo to understand the real-world implications of testing house models in hurricane scenarios, which corresponds to the CT practice of testing.

In Mary’s classroom A, Alicia, who worked with Mariana throughout the project, also made statements on the concept of simulation during their interview."Oh it means doing a test. it's something the house would just fall over in real life. But a test to see if that was a strong house. If it would stand or not." (Group interview 04/03/22). Based on the conversation between the interviewer and Alicia, it is evident that Alicia demonstrates a clear understanding of a simulation as a test to evaluate a house's strength and its ability to withstand real-life conditions. Her explanation reflects practical knowledge of simulation's role in assessing building resilience, which corresponds to the testing practice of the CT framework used in the current study.

In Mary’s classroom B, when Amelia was asked what fans and spray bottles represent in their house model interview, Amelia said that “It represents how stable the house is. So it's like winds with one hit, it could knock over or it could stand still like shake a little bit” (Group interview 06/01/22). Alice, again in Mary’s classroom B, also made similar statements to Amelia during her interview:

> Interviewer: During the simulation, what did your teacher use?
> Alice: She used the wind from the fans and the water from the spray bottles and the pouring to act like and the water from the ocean and the wind.
> Interviewer: So why do you think she used those?
> Alice: Because they act basically like a hurricane. Because a hurricane is just filled with full wind and water.
> Interviewer: How about what do you think simulation is? How would you define simulation?
> Alice: I would define it as a test of how strong we did. How strong would the house be if a hurricane hit and a tornado hit.
>
> (House model interview on 06/02/22)

In the interview, Alice describes the simulation's purpose and components. The use of fans and spray bottles to mimic wind and water accurately represents a hurricane's impact. By simulating these conditions, Alice recognizes the importance of testing house models to assess their strength against real-life disasters like hurricanes and tornadoes. The concept of simulation, as described by Alice, is seen as a crucial tool for evaluating the resilience of structures under extreme conditions. This understanding emphasizes the significance of simulations in preparing and building resilient structures that can withstand the forces of natural disasters. As a result, throughout the students' interactions and demonstrations from three different classrooms, it is evident that they recognize simulations as a critical means of testing resilience and efficacy, which corresponds to CT practice of testing, underscoring the importance of simulation in understanding and preparing for real-world challenges.

Computational Thinking Practices in House Model Planning

In the subsequent sections, data collection practice in planning will be discussed followed by decomposition practice in planning. This section is where I address RQ1b: What CT practices did 2nd graders develop in planning their house models? I will draw upon the following data to address this RQ: house model interviews, Scratch interviews, field notes, and photographs of classroom artifacts such as house models and poster boards.

Data Collection Practice in Planning. It is important to note that data collection is not a part of Brennan and Resnick’s framework as a practice, about which I further comment in implications in the discussion Chapter. Data collection refers to the systematic gathering of information relevant to a particular problem or situation. In this case, participants conducted research on natural disasters, especially hurricanes, to collect information through different sources such as doing research on different natural disasters on Get.epic, which is a website on which there are many, various online books on different disciplines and topics, watching videos, observing their simulations, and engaging in class discussions. During implementation of lesson 4, the teachers even had students watch a specific video on how scientists collect and record data through a wind power experiment. Below is an example of the CT practice of data collection on natural disasters from a group of two in Mary’s classroom A.

> Interviewer: What natural disaster did you research?
> Nico: The natural disaster that we researched was a hurricane.
> Interviewer: [...] What did you learn about hurricanes?
> Nico: Well, we learned that hurricanes are really strong. They have really hard winds. They can blow away big stuff. And the skyscraper basically makes it help cause they basically twirled it around for the wind [...].
> Interviewer: Do you (Rafael) wanna say anything?
> Rafael: And if you twist it, then people will be safe because the wind doesn't knock it over.
> Interviewer: Got it. What else did you learn about hurricanes?
> Rafael: I also learned [...] that they can be really strong and I thought we [...] should build the barrier because it'll block the wind.
>
> (Group Interview 03/25/22)

From the conversation, it's evident that Nico and Rafael engaged deeply in their research about hurricanes. Their insights about the strength of hurricanes, the effects of wind on structures, and the idea of designing barriers showcase their active engagement in data collection and interpretation regarding natural disasters. Their ability to draw conclusions and suggest protective measures further emphasizes their thoughtful approach to understanding these catastrophic events. Furthermore, during my classroom observations, Nico shared the following with me after reading about hurricanes online. “I noticed from the picture in the book that all the houses made out of wood are gone. But, houses made out of bricks are still standing good. So, we should make our building from bricks against hurricanes” (FN 3/17/22). Based on Nico's observation, it's evident that he effectively engaged in the CT practice of data collection. By reading about hurricanes and examining photographic evidence, Nico identified a pattern that wooden houses were more likely to be destroyed, while brick houses remained intact. This led him to a logical conclusion – choosing bricks as a more resilient material for building in hurricane-prone areas. His approach showcases how even young learners like second graders can gather and analyze data to make informed decisions, a key aspect of computational thinking.

Another exampler of the CT practice of data collection is from Diego and Caleb from Mary’s classroom B. Below is our conversation about their research/data collection on natural disasters, specifically on hurricanes and tornadoes.

> Interviewer: What natural disaster did you research?
> Diego: Hurricane.
> Interviewer: how about you, Caleb?
> Caleb: Hurricanes. And tornadoes.
> Interviewer: What did you learn about hurricanes?
> Caleb: You have to go in a small room with no windows.
> Interviewer: Okay. How about, Diego, what did you learn about hurricanes?
> Diego: What I learned about tornadoes, that if you're driving in a car and you see a tornado foreman, you have to find shelter where probably the tornado won't hit so that it can be safe. Because if you keep driving near the tornado, you can get spun into it. Yeah. And then you cannot get out.
> Interviewer: Got it. What do you think causes hurricanes?

> Diego: When I think it was wind, water, firming and, well that makes the tornado and then it goes into water, which makes it a hurricane cuz it's a tornado and water trying to, they're all destroying stuff.
> Caleb: Yeah.
> Interviewer: Okay. What are the consequences of a hurricane and a tornado?
> Diego: Damage to houses. Maybe even an animal store or any place that's like you can go into and do stuff you can. So it just destroys everything. The road. Maybe even make a hole in it and damage houses and other shelters.
> Interviewer: How can we protect ourselves from natural disasters such as hurricane?
> Caleb: If you don't have a small room with no windows, you could board up your windows maybe.
> Diego: You could build a structure or build a house with air conditioning, but no windows so you can get air at the same time. And when a hurricane comes, you have all the rooms have no windows.
>
> (Group Interview 06/02/22)

Diego and Caleb's responses illustrate their engagement in the CT practice of data collection. They've gathered nuanced insights on hurricanes and tornadoes, offering practical suggestions like seeking shelter in rooms without windows. Their active data collection not only enhances academic understanding but also empowers them to propose real-world solutions, highlighting the importance of this CT practice of data collection. Additionally, I engaged in a discussion with Diego about a hurricane book that his mother had recently acquired for them. Here are my field notes on this: “His mom got a hurricane book for him yesterday and I asked him what he learned from it. He said when a hurricane happens, we need to go hide in a basement. He also shared that the picture in the book his mom got for him, in which all cars and houses were floating in the water after the hurricane happened, was the same as the picture in the book he read on get.epic for their research.” (FN 5/17/22). Diego's interaction with the hurricane book, both from their mom and on get.epic, illustrates his engagement in data collection, a core component of computational thinking in this dissertation. By correlating images from both sources, he not only reinforced his understanding of hurricane impacts but also derived a practical safety measure, which is seeking shelter in a basement during a hurricane. This reflects his ability to gather information, recognize patterns, and apply his findings to real-world scenarios, even as a young learner.

Last example for this CT practice is from Julia and Jorge in Kimberly's classroom. Below is a conversation between the interviewer and the students.

> Interviewer: What natural disaster did you two research?
> Julia: So we researched that a lots of rain cause the of flood and that can endanger your house and it can take stuff and that's all I know. And, we learned that it break[s] things.
> Interviewer: How about you, Jorge. What did you learn from your research?
> Jorge: Hurricanes can be caused in Florida mostly. Yeah, mostly in Florida. So they can cause a lots of damage to cars. Such as trees, cars and houses. [...] I know it Because my dad had to a lot of hurricanes. [...] He lives in Florida.
> Interviewer: Let's talk about the consequences of hurricanes.
> Julia: A hurricane causes people to be homeless and have no houses. [...] It takes things away from people.
> Interviewer: How can we protect ourselves from hurricanes?
> Jorge: You can go to a basement, you can go inside of a little bunker or something or things like that.
> Interviewer: Based on the data you collected, the research you did, how can we build a house that is strong enough to sustain hurricanes?
> Jorge: You could make, so you don't really have to build it. It's like when you get built at a house, if it has a basement, you don't really have to build anything. You just have to go into the basement. You have to stay in there and make sure the doors are locked or else.
>
> (Group Interview 04/06/22)

Julia and Jorge's detailed observations and recommendations underscore their active involvement in data collection, a key Computational Thinking practice. Their findings not only show a depth of understanding about the consequences and safety measures concerning hurricanes but also contribute to practical strategies for building resilient houses. In summary, students Rafael, Nico, Caleb, Diego, Jorge, and Julia, like many other participants of the current study, engaged in research and data collection to understand the impact and causes of natural disasters like hurricanes. Their insights highlight that they considered different aspects, such as the strength of winds, the need for safe spaces like basements, and the consequential damage, to inform their designs for more resilient structures. This shows that data collection is a key element in their problem-solving process, which shows that gathering information helps them make better choices.

Decomposition Practice in Planning through a Poster Board.

Decomposition/modularization involves creating systems or projects by assembling a series of smaller, more manageable components (Brennan and Resnick, 2012).I will be drawing upon the following data sources in addressing this section: house model interviews, scratch interviews, field notes, photographs of classroom artifacts such as poster boards.

The lesson design strategically incorporated the use of poster boards for each group, serving as a critical tool in the planning and documentation of their projects. These boards were divided into three distinct columns to facilitate a structured approach: the first column for material planning and house model drawing with labels, the second for recording simulation data and illustrating results, and the third for proposing revisions. This deliberate organization of the poster boards was essential, not just for recording data, but as a pivotal anchor throughout the construction process, steering the students' planning, analysis, and iterative thinking in line with the objectives of the activity. As shown in the accompanying photograph, the poster board from a group of three students serves as a representative example. It's important to note that all groups across the three classrooms adhered to the same structural format for their poster boards, which was captured through photographs.

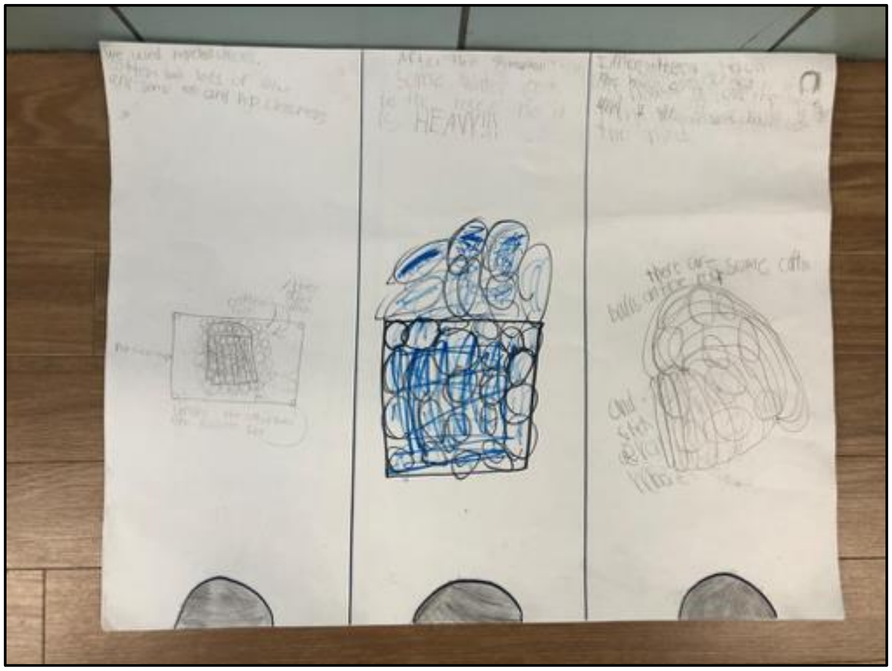


Figure 3: A picture of one group of students' poster board

Dividing the poster board into three columns for planning, recording simulation data, and proposing a revision(s) allows students to break down the complex task of designing and testing house models into manageable steps.

In the Scratch interviews, students elaborated on their decision to divide their projects into three separate sprites, mirroring the three distinct columns on their physical poster boards. For instance, when queried about the reasoning behind their three Scratch sprites, Travon from Mary's Classroom A mentioned, "The teacher said that we have to do it [in] three things," (Group interview 04/04/22) referring to the three categories outlined on their poster board. Similarly, Charlotte from Mary's Classroom B explained, "Because, of course, Mary [the teacher] told us to do [it], also we can answer just three different questions on our poster." (Group interview 06/03/22). In both instances, students linked the organization of their Scratch sprites back to the three questions or sections on their physical poster boards. This was also echoed by Samuel from

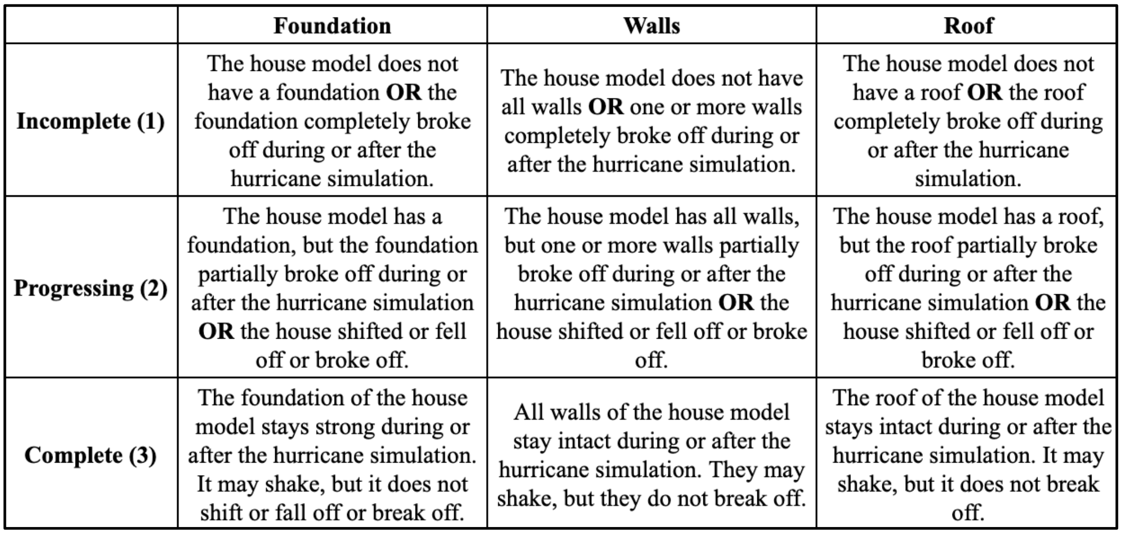
Kimberly's classroom, who affirmed that the three sprites corresponded to "three numbers or three things on the poster" (Scratch interview 04/08/22). It is important to note that the purpose of the three columns on the poster significantly differs from the use of three sprites in Scratch. While students utilized the quantitative data from the poster in relation to the number of sprites, it's clear from their comments that the poster's three-column structure was not designed to symbolize three characters. Instead, it served a distinct function in the project's organization. In conclusion, this practice of segmenting their poster boards and Scratch projects into three distinct categories represents the Computational Thinking (CT) practice of decomposition. Students were essentially breaking down the complex task of presenting their projects into smaller, more manageable components, thereby demonstrating their grasp of this key CT practice.

Evaluating Simulation Results (Testing Practice)

CT practice of testing involves running the artifact through various scenarios and using specific test cases to identify any errors, bugs, or unexpected outcomes (Brennan and Resnick, 2012). Students' comprehension of simulation, which was discussed above as part of RQ1a, is different from the actual simulation or test that was performed by the two teachers to measure the resilience of students' house models. Therefore, it is important to discuss the performance of students' house models during simulations. Please note that the main point with this research question is to show that students engaged in the CT practice of testing through simulation. To address RQ1c, to what extent did participants' initial house models survive from the simulation/test, I developed a rubric with three criteria: the foundation, walls, and roof of the house models. These criteria were chosen because students frequently discussed these parts of their house models in interviews. During interviews, I asked students if they thought their house models were strong. Even when parts of their models failed in simulations, some students still believed their houses stayed strong. Supporting this observation, teacher Mary noted in a field note that “When we were actually running it through the simulation, there was that sense of joy as we were spraying them. And of course that sense of sadness when we were destroying them.” (FN 05/30/2022) Students displayed emotional reactions during the simulations — they exhibited joy while their models were being tested and sadness when the models failed. This emotional engagement indicates that the students were indeed emotionally invested in the outcomes of their models.

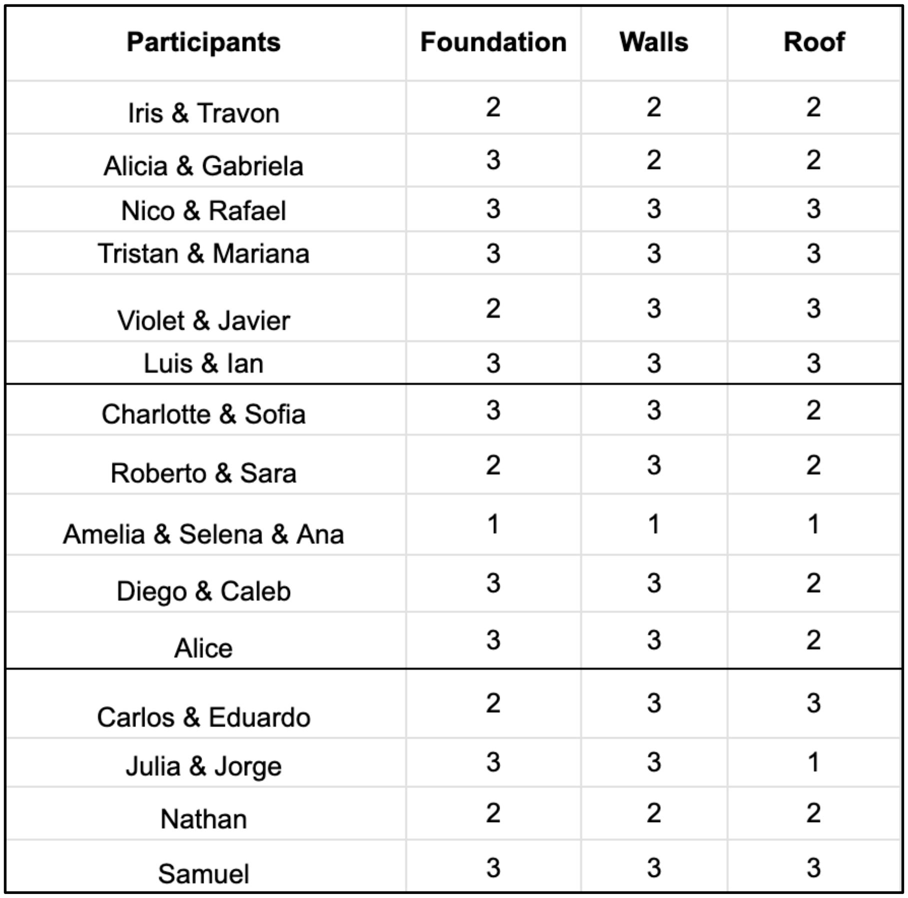
Given these emotional responses, including joyful reactions even when parts of the house model became compromised in the simulation, I developed a rubric to evaluate the house model construction together with student reactions. In developing this rubric, I took into account the students’ comments made during the interviews, ensuring the evaluation criteria aligned with the elements they consistently emphasized: the roof, walls, and foundation. This approach was intended to give students due recognition for their work and to reflect the components they regarded as most critical to their house models. Table 7 below is the developed rubric to evaluate students’ house models through a hurricane simulation. Description of Table 7 is below.

Table 7: A rubric to evaluate participants’ house models through a hurricane simulation/test



	Foundation	**Walls**	**Roof**
Incomplete (1)	The house model does not have a foundation **OR** the foundation completely broke off during or after the hurricane simulation.	The house model does not have all walls **OR** one or more walls completely broke off during or after the hurricane simulation.	The house model does not have a roof **OR** the roof completely broke off during or after the hurricane simulation.
Progressing (2)	The house model has a foundation, but the foundation partially broke off during or after the hurricane simulation **OR** the house shifted or fell off or broke off.	The house model has all walls, but one or more walls partially broke off during or after the hurricane simulation **OR** the house shifted or fell off or broke off.	The house model has a roof, but the roof partially broke off during or after the hurricane simulation **OR** the house shifted or fell off or broke off.
Complete (3)	The foundation of the house model stays strong during or after the hurricane simulation. It may shake, but it does not shift or fall off or break off.	All walls of the house model stay intact during or after the hurricane simulation. They may shake, but they do not break off.	The roof of the house model stays intact during or after the hurricane simulation. It may shake, but it does not break off.

For a house model to be incomplete, it does not have one of the following: roof, foundation, or walls (main parts of a house), or even if it has the main parts, it completely broke off during or after the hurricane simulation. For any main part of a house model to be progressing such as for foundation, the main parts are complete but that part partially broke off during or after the simulation, or the house model shifted or fell off during the simulation. For a main part of a house model such as roof to be complete, it should stay strong during hurricane simulation. It could shake but as long as it does not shift or fall off, it is considered complete. Based on this rubric, statistics of each house model can be found below.



Participants	**Foundation**	**Walls**	**Roof**
Iris & Travon	2	2	2
Alicia & Gabriela	3	2	2
Nico & Rafael	3	3	3
Tristan & Mariana	3	3	3
Violet & Javier	2	3	3
Luis & Ian	3	3	3
Charlotte & Sofia	3	3	2
Roberto & Sara	2	3	2
Amelia & Selena & Ana	1	1	1
Diego & Caleb	3	3	2
Alice	3	3	2
Carlos & Eduardo	2	3	3
Julia & Jorge	3	3	1
Nathan	2	2	2
Samuel	3	3	3

Figure 4: Scores of participants’ house models based on the rubric

In this rubric, complete house models took 3 points while progressing ones took 2 points and incomplete house models received 1 point. Based on the rubric, participants' foundation levels are as follows: While 60% of participants' foundation was evaluated as "complete", 33% of foundations were "progressing" and 7% of participants' foundations was incomplete. As for walls, while 73% of house models' walls were complete, 20% of their house models' walls were evaluated as "progressing" and 7% of house models' walls were evaluated as "incomplete". With respect to roof, while only 40% of participants' house models' roof was evaluated as "complete" roof, 47% of the house model roofs were still "progressing" and 13% of roofs was evaluated as "incomplete". In the following paragraph, I will discuss the structure of how I answered RQ1c followed by strategically chosen examples from the three classrooms to exemplify the three levels, which are complete, progressing, incomplete, students' house models fall into based on the rubric, why they did so.

To clarify the approach I took to answer RQ1c, I selected one example of simulation results from each of the three classrooms I observed. The first example is from Mary's Classroom A, the second from Mary's Classroom B, and the third from Kimberly's classroom. And, each example from each classroom was one at each of the three levels, which are complete, progressing, and incomplete. The primary sources of data for answering RQ1c were house model interviews, one of the main sources of the current dissertation, as well as photographs of students' house models, used for triangulation purposes (Please refer to Table 3 in the methods section to see what data sources were used to address which research question). In the following sections, I will detail the features of a few house models from these classrooms to demonstrate how the rubric was applied.

One example of a house that got “complete” for each level was Tristan and Mariana’s house model from Mary’s classroom A, which is below.

Figure 5: A screenshot of Tristan and Mariana’s house model before hurricane simulation

Their model had a strong foundation that was made out of a couple of layers of popsicle sticks and playdough and they used cardboard at the very bottom to strengthen their foundation. The model had four walls made from index cards that were taped to their roof, which was made out of cardboard. During simulation, their house model stayed so strong that none of the main parts of the model broke off or shifted. During their house model interview, when I asked them if their house model passed the simulation, Mariana replied that “I think it passed” (Group Interview 03/28/22). Tristan agreed with her and added that “I think it passed because it didn't move and get [got] wet” (Group Interview 03/28/22). After this short conversation, I asked them to read their simulation data in their poster board, which included three columns to take notes about materials planned to use, simulation result, and proposing a revision. Tristan replied: “Yeah. Yeah. Using the data, I would add a Play-Doh on the cardboard [Mariana and Tristan are reading it together] to make the house, the house not get wet. Only the play-doh get [got] wet. Playdoh Playdoh get [got] wet and I would make the walls with play-doh so it would be more stable. More stable." (Group Interview 03/28/22). The foundation of their house model was so strong because they used multiple layers of playdough and popsicle sticks. When I asked them why their house model didn't move during hurricane simulation, Tristan replied: "Probably because the Play-Doh and the popsicle stick[s] like kept them." (Group Interview 03/28/22). As a result, since this group had each main part in their house model and each of those parts was strong during simulation, each level got "complete" criterion.

Roberto, Sara, and their partner from Mary's classroom B worked together during the module. Their house model was evaluated as "progressing" for foundation and roof, and "complete" for walls (Figure X). Its foundation and roof was made from construction paper while the walls were cardboard. Although it is not seen in the picture above, they added their foundation afterwards.

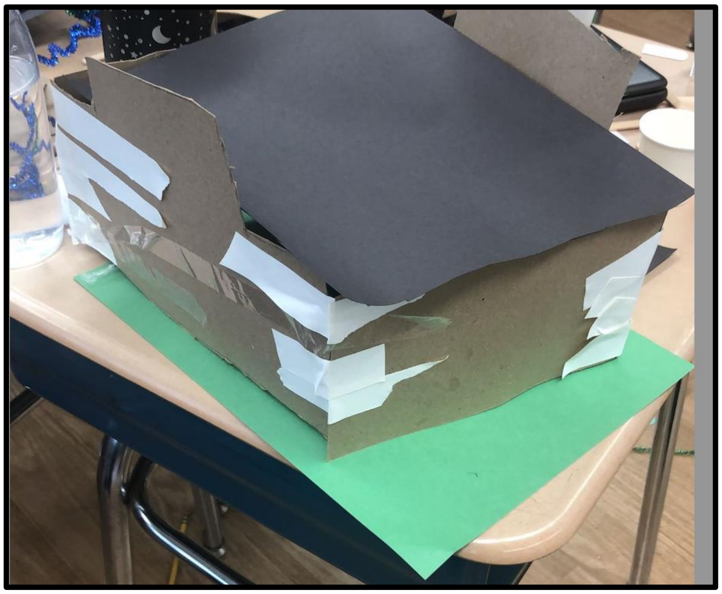


Figure 6: A screenshot of Roberto, Sara, and their partner's house model before simulation

In their house model interview, Roberto described what happened to their house model during the simulation: "What happened to our house model was it got kind of drenched in water, but it survived, even though the roof started caving in." Group interview 06/02/22). Although the group did not talk about their foundation, it is clearly seen from the picture above taken right after the simulation (Figure 7) that their base partially fell apart. As Roberto said, their roof caved in due to the water that was poured and sprayed. But, their walls survived in the simulation. For the above mentioned reasons, their roof and foundation were evaluated as "progressing" and their walls were evaluated as "complete" based on the rubric.

Below is a screenshot of their house model right after the simulation.

Figure 7: A screenshot of Roberto, Sara, and their partner's soaked, caved-in roof after simulation

Selena, Ana, and Amelia from Mary's classroom B were partners in the project. Their house model that got "incomplete" for each level was a house model made out of mainly popsicle sticks but covered up with a lot of cotton balls (Figure 8). They built their house in a way that the model had all three main parts such as roof. However, to close the gap between the roof and the wall on the right top in Figure 8 below and so that the water from hurricane simulation does not get in, they covered the house with a lot of cotton balls this time (Figure 9). The model collapsed a couple of days after hurricane simulation because it became so heavy due to the fact that cotton balls absorbed the water that was poured and sprayed during simulation. Amelia explained this during their group interview. "[...] because I saw you guys [teacher and the researcher] testing [simulating] all the other kids and I knew you guys were going to put a lot of water, so we just put a lot of cotton balls on there. But we never knew that it could get really heavy." (Group interview 06/01/22). In the simulation, their entire house model including the roof, the foundation, and the walls completely collapsed so their entire model was considered "incomplete" for each level (Figure 9).

Figure 8: A screenshot of Selena, Ana, and Amelia' house model before hurricane simulation

Below is a picture of their house model after simulation.

Figure 9: A screenshot of Selena, Ana, and Amelia’ house model after hurricane simulation

The above three examples were meant to explain not only how the rubric was applied while evaluating the participants’ house models in a hurricane simulation scenario but also to what extent the participants’ house models survived during a hurricane simulation, which is directly related to testing practice in testing and debugging practice. Debugging, which corresponds to proposing a revision to their new house model plan, will be discussed under RQ3, as it is closely connected to CT practices regarding students’ material choices. Therefore, testing and debugging were separated in the current dissertation study although they are usually thought together as in Brennan and Resnick’s (2012) framework.

CT Practices regarding Students’ Material Choices

This is where I address RQ1d, which CT practices did second graders engage in describing and understanding their material choices they made in creating their house models and proposing a revision? I found two CT practices regarding students’ material choices they discussed during their interviews: CT practice of being incremental and iterative, and debugging. Here are the definitions of those two CT practices from the framework. Being incremental and iterative refers to the adaptive process of project design, where planning and implementation are not strictly sequential but evolve through small, progressive steps that allow for continuous refinement and adjustment in response to emerging solutions. Debugging involves developing strategies to anticipate and resolve issues that arise during the design process, as the outcome seldom aligns perfectly with the initial conception. Materials were participants' most frequent coded in house model interviews because materials were at the center of participants' discussions during the interviews, especially house model interviews. All of this is consistent with tenets of constructionism. This is understandable in a context where participants planned what materials to use for their house model, they built their houses with those materials, that most of the problems with their houses during simulation stemmed from materials such as that their roof or walls dislodged or carried away during simulation, and that they proposed a revision to the problems that materials caused during simulation. Consequently, materials assumed a central role in the activity that the participants were engaged with. After analyzing the interviews as well as field notes, students' discussions with their house models suggest that they built the house models in a gradual and iterative manner and proposed revisions of the materials that they thought failed during simulation, which corresponds to debugging practice in the framework of the current dissertation study. This will be discussed in the subsequent sections where I will explore participants' explanations for their material selections during the house model interviews regarding CT practices and discuss any proposed changes they made to their original designs. Three instances below are from the three classrooms to ensure data representation for each classroom. I draw upon interviews data (both house model interviews and Scratch interviews) as well as pictures of students' models (Please refer to Table 3 in the methods section to see what data sources were used to address which research question).

Instance of Charlotte and Sofia. Charlotte and Sofia, from Mary's classroom B, worked together throughout the project. The materials they used in their house model were construction paper, popsicle sticks, tape, cotton balls, cardboard, and tin foil. Before their final house model, they started to build a house model the walls of which were made out of cardboard and glue. However, they failed to glue cardboard in their first house model attempt so they started on a new house model. During their house model interview, I asked them what happened to their first house model so they needed to start over.

> Charlotte and Sofia [at the same time]: The Cardboard walls wouldn't stay.
> Sofia: The glue wasn't dry. And we didn't have enough time, like we start[ed] the walls. So we just started on a brand new house.
> Charlotte: And they [cardboards] kept falling. [...] The first one was a total disaster.
> Interviewer: So you said that you couldn't glue [the cardboard to] the construction paper.
> Charlotte: Yeah. Cause the cardboard kept falling apart.
> Sofia: And Charlotte kept saying that the tape isn't gonna be strong enough
> Charlotte: And neither was the glue. <laugh> I think we kind of tried both.
> (Group interview 06/02/22)

Charlotte and Sofia encountered difficulties in adhering cardboard to construction paper, which served as the foundation for their initial house model walls. As Charlotte acknowledged, the first attempt resulted in a catastrophic outcome. This signifies their inclination towards an iterative approach in developing their house model, prompting them to initiate the construction of a new house. Furthermore, after this short conversation on the matter of glue, I asked them what they used to make the walls of their new model:

> Sofia: We used pops sticks. Like we were planning. And then we decided to change it [glue].
> Interviewer: Change it [glue] to what?
> Sofia: Like we were thinking of using tape and then we both like, well, the tape isn't as strong as glue. I wish we had hot glue to make it dry, faster and stronger.
> Interviewer: Okay.

> Sofia: But we didn't. So we had to have normal glue. And also then I'm just, I just remembered, I once went to a science book when glue hits water again and turns back to liquid. So I wasn't thinking glue was too much of a good idea.
> Interviewer: Mm-hmm.
> Sofia: We taped and glued that way. Glue would also help like keep 'em together, but the tape would keep the glue from getting wet.
> Interviewer: Yeah, that makes sense.
> Charlotte: Yeah. At first we tried glue and it wouldn't work.
>
> (Group interview 06/02/22)

Because glue didn't work well to stick the cardboard together, they decided to use popsicle sticks along with cardboard for support. They also decided to use tape along with glue, even though Sofia thought that tape isn't as strong as glue. But the group agreed to use both glue and tape because Sofia had read in a science book that glue can melt when it gets wet. So they glued the popsicle sticks to the cardboard and taped them to protect the glue from getting wet and melting during the simulation. As a result, they decided to use popsicle sticks, cardboard, tape, and glue to build the walls of their house model as seen in Figure 10 below. The iterative and incremental nature of their decision-making process is evidenced by the modifications undertaken in constructing the walls of their novel house model, indicating their exploration to identify the optimal material choice from the available options. In the following paragraphs, I will discuss how Charlotte and Sofia proposed a revision based on the data they collected from the simulation of their house model.

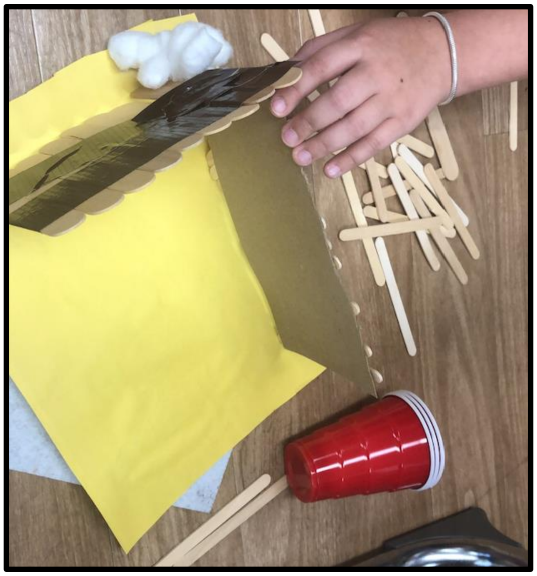


Figure 10: A picture of the walls of their new brand house model

After completing their new house model, their teacher with the help of the researcher of the current dissertation tested/simulated their new house model with spray bottles and fans, which represent a hurricane in this case. Below is a picture of their new house model before simulation.

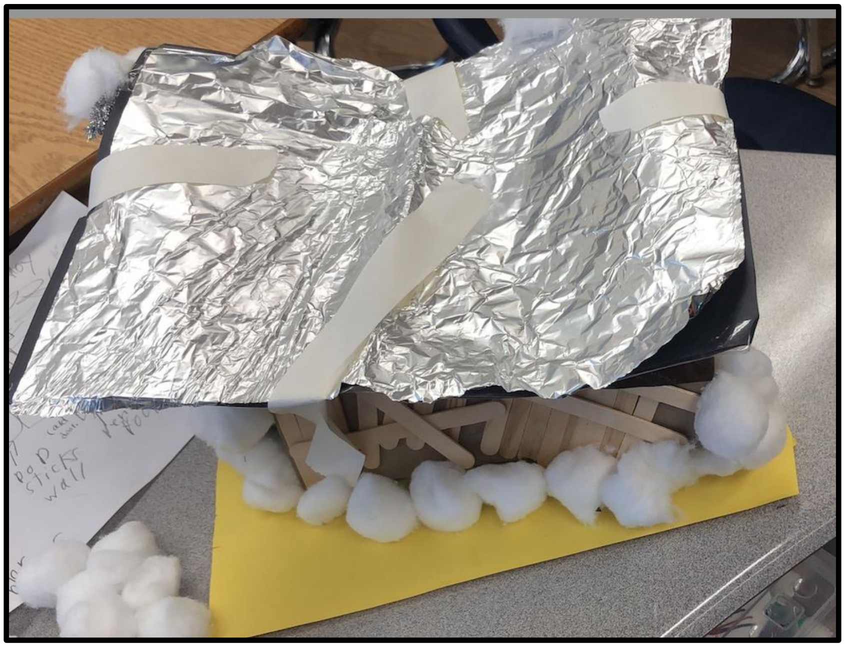


Figure 11: A picture of Charlotte and Sofia's new house model before simulation with no puddles on the roof

Charlotte and Sofia used tin foil that they supported with construction paper, which they taped to the tin foil, to make the roof of their house model. After simulation, their roof caved in due to water sprayed and poured on their model. Below is a picture of their house model after simulation.

Figure 12: A picture of Charlotte and Sofia's new house model after simulation with puddles on the roof

In their house model interview, I asked them what they noticed during the simulation of their house model. Sofia replied that "In the hurricane, so the house didn't really break. Puddles were on the roof" (Group Interview 06/02/22). As Sofia said, their house model had puddles on the roof but neither the roof nor any other part of the house broke. During the house model interview Charlotte proposed a revision to change the roof as the following:

> Charlotte: And also we do need to change something. We probably do need to change…
> Sofia: The roof.
> Interviewer: The roof?
> Charlotte: Yeah
> Interviewer: Why?
> Charlotte: "Because, um, it was too easy for the, um, it was too easy for the tin foil to get puddles in it so we couldn't use tin foil, even though we did use, um, construction paper under it. That was in trouble. I think next time we have to use cardboard."
> Interviewer: Cardboard for the roof? Not tin foil?
> Charlotte: Yeah
>
> (Group interview 06/02/22)

Due to the occurrence of water puddles during the simulation, the tin foil material employed for the roof of their house model caved in, prompting Charlotte and Sofia to propose a revision in their approach. They determined that substituting tin foil with cardboard for the roof would be more suitable. Notably, participants predominantly discussed their proposed revisions through the Scratch platform during their Scratch interviews. When specifically inquiring about their observations during the simulation in a Scratch interview, Charlotte responded by utilizing one of their programmed sprites, stating, "One thing we noticed from the simulation is that the tin foil roof helped, but it also made a few puddles on top. Using the data from the simulation, we now know how we should change, that's by changing the roof, adding more tape." (Group interview 06/03/22). By examining the dialogue from their house model interview and the subsequent Scratch interview, it becomes evident that although the tin foil roof offered some benefits during the simulation, the presence of puddles necessitated a revision so they proposed the adoption of a cardboard roof while also implementing additional tape to enhance the structural integrity. Consequently, the integration of incremental and iterative approaches, as well as the implementation of debugging practices, assumed pivotal roles in Charlotte and Sofia's decision-making process regarding materials. These practices served as the primary means through which they sought to enhance the resilience of their house model when subjected to the hurricane simulation.

Instance of Iris and Travon. Iris and Travon, from Mary’s classroom A, used index cards and tape for the walls, popsicle sticks, glue, and tape for the base, and cardboard and tape for the roof to make their house model. Below is a picture of their model before the simulation.

Figure 13: A picture of Iris and Travon’s house model before simulation

During their house model interview, the group discussed the importance of having a strong foundation in a house.

> Iris: Yeah. So we created a good base. That's important that you need to have [a] very strong base because wind could get in [from the bottom corners because the tape there came off after they built it]
> Travon: [...] For the base down here, I put glue and she put tape on it.
> Interviewer: Why did you use both?
> Iris: To make it stronger. [...] You can see it's unbreakable (The base is made from popsicle sticks]. Here in this part [the hole between popsicle sticks and paper wall at the bottom corners], it's not working.
>
> (Group Interview 03/25/22)

Knowing the importance of having a strong base in a house in case wind could get in the house through the holes at the bottom corners, Iris and Travon built a strong base for the model using popsicle sticks that were taped and glued. These suggest that they built their foundation in an incremental way. Moreover, when they were asked why they utilized both tape and glue, Iris responded as to make the base stronger. During the house model interview, Iris even gave a mini test to their foundation by holding the base in her hands and bending it a couple of times to show me that it is unbreakable [strong]. This shows that she tested their house model, which corresponds to testing practice, although that testing was not the actual simulation.

Iris and Travon suggested a strong base for the house because during a simulation, wind from fans entered through bottom holes, causing the house to move.

Figure 14: A picture to show holes at the bottom and top of Iris and Travon’s house model

This was because they were not able to tape well not only the walls to each other but also the walls to the base, which led the house to have holes. In their Scratch interview, Iris mentioned this as follows: “[Be]Cause you need a lot of tape and we didn't use that much. And then there would end up being holes” (Group Interview 04/03/22). Iris also described what happened to their house during the simulation as the following: “Our house jumped because the wall [index cards] was [were] not sturdy. We used index cards. And the index cards are not the strongest materials" (Group Interview 04/03/22). When asked why their house shifted during the simulation, Iris pointed her finger at the holes at the bottom and said “it’s holes” (Group Interview 04/03/22). She added that “It basically flipped off cause the wind was so strong it eventually just pushed it like a giant pond” (Group Interview 04/03/22). These all indicate that their house were not sturdy enough against a hurricane simulation. Additionally, during the simulation, not only did their house get wind through the holes but also it got wet due to water sprayed. Travon stated this during their Scratch interview as follows: “[...] since it's weak and there's a huge amount of wind, and there's a lot of water, the water could dampen and then it caved in” (Group Interview 04/03/22). Based on the data they collected from the simulation, the group proposed some revisions/solutions on their Scratch project. Here is what they recorded in one of their sprites in Scratch. "We would use legos, cardboard, playdough, and popsicle sticks because they are very strong. Instead of paper for the walls, we would use cardboard because cardboard is strong material. And paper was more likely to fall over" (Group Interview 04/03/22). As a result, after noticing what happened to their house model during the simulation as described above, they decided to change most of the materials they used in their first trial. Iris and Travon's experience during the hurricane simulation highlighted the need for stronger materials and careful building, showing the value of learning through repeated trials, which suggests an iterative process. I will discuss each of their revisions and their reasoning for those revisions in the following two paragraphs.

First, when I asked them about their use of index cards, Iris and Travon discussed this during their house model interview.

> Iris: “We learned that paper is definitely not the strongest one it could get wet.”
> Interviewer: I see.
> Travon: [...] the next time we could put tape all around the base and then see if it works.

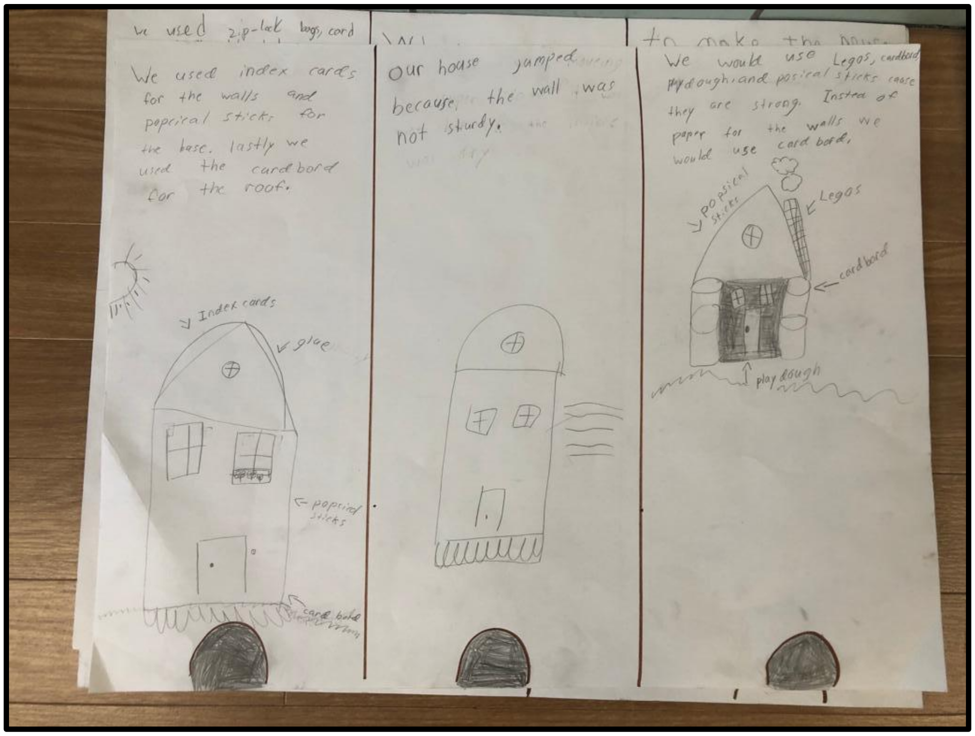
> Iris: Yeah, we'll definitely check if there's any holes. I'm not going to ignore the holes cause that's what I did. Actually I'll put more glue, more tape, make sure it's stable. Give myself a mini test.
>
> (Group Interview 03/25/22)

Travon and Iris agreed not to use index cards and to tape the base [to the walls] better in their new house model proposal. They opted to utilize cardboard for the walls, given that their walls not only became damp but also experienced infiltration of air, which caused the house to move during the simulation. Travon explained this during their Scratch interview as "I think the index cards didn't work because it got a little bit wet. And it got wet and soft" (Group interview 04/03/22). Upon this, I asked them what they would change the index cards to. Travon replied "To Cardboard" (Group interview 04/03/22). These observations highlight the ongoing evolution of their design choices, underscoring the essence of iteration in perfecting a house model.

Second, again during their Scratch interview, when I asked them what else they would change in their new house design, Iris and I had a conversation as follows:

> Interviewer: So in your new design, you would use what for the base?
>
> Iris: play-doh for the base.
>
> Interviewer: Why is it?
>
> Iris: Because it's strong and as I said, it would harden it [the base]. It gets really hard and adds two layers. So it would be very thick and then the house would be anchored down to the ground.
>
> Interviewer: Got it.
>
> Iris: It would be very hard to move. (Group interview 04/03/22)

The group decided to propose a revision to their design in a way that they would use playdough for the base instead of popsicle sticks because Iris thought that playdough is strong and it hardens and makes the base stronger. The above-mentioned revisions can be also seen in the third column of their poster board that they made use of throughout the project, which is below:



we used zip-lock bags, card

to make the

We used index cards for the walls and popcical sticks for the base. lastly we used the cardbord for the roof.

our house jumped because the wall was not sturdy.

We would use Legos, cardbord playdough, and posical sticks cause they are strong. Instea of paper for the walls we would use card bord.

Figure 15: A picture of Iris and Travon’s poster board

The poster board was divided into three columns. In the first column, they wrote down what materials they planned to use before they started building their house model. In the second column, they recorded their observations about what happened to their house model during simulation. And, in the third column, they wrote down the proposal of their revision(s). The revisions of index cards to cardboard and of popsicle sticks to playdough can be seen in the poster board above. Iris and Travon's decision to modify their original design materials, as reflected in their poster board, emphasizes the continuous and adaptive nature of the design process, showcasing the significance of iteration in refining a house model.

Instance of Nathan and his partner. Nathan worked with his partner during their house model project. His partner was not allowed to be interviewed so I interviewed Nathan only. Below is a picture of the house model they built.

Figure 16: A picture of Nathan and his partner’s house model before the simulation

Nathan and his partner used a unique combination of materials for their house model project: cups, popsicle sticks, tape, and newspaper. At a cursory glance, their choices and design could appear complex. Firstly, Nathan explained the double-cup feature, saying, “[...] you see these two cup[s], and this one [the one that is outside] down here was supposed to be the basement and these [popsicle sticks] almost protecting a part of the basement.” (House model interview 04/06/22). The outer cup symbolizes the foundation, supported by popsicle sticks, designed especially "for floods and stuff like that" (House model interview 04/06/22). Secondly, another standout feature is a third cup, serving as an alarm system. When asked about its function, Nathan said it’s against “natural disasters” (House model interview 04/06/22). He elaborated, “Okay. So this is the little alarm system... So it can echo” (House model interview 04/06/22). Below is a picture of the alarm system on their house model.

Figure 17: A picture of Nathan and his partner's alarm system on their model

Probing further into the echo function, he justified, "To make it loud because this is going to be the mayor's house and the mayor is going to warn everybody about natural disasters" (House model interview 04/06/22) The alarm's design, with a popsicle stick inside, works like a bell, as demonstrated by Nathan: "It was supposed to shake [shaking the popsicle stick in the cup to show how it makes sound] around and smack up and down so it can make noise" (House model interview 04/06/22). Lastly, Nathan and his partner used newspapers as the roof for their house model. Each material choice by Nathan and his partner had a specific, intentional use, and further discussions will delve into the CT practices they employed during their project.

When Nathan and his partner built the mayor's house, they went through a step-by-step process. During the house model interview, I asked Nathan how they built the model, he discussed the process they underwent: "So first we took the sticks and then we lined them up like a triangle.

And we taped. And then we took two sticks sometimes and we wrapped it. Then we snapped a stick in half, we snapped in half but we taped it" (House model interview 04/06/22). Nathan's statements demonstrate their step-by-step manner while building the mayor's house. In the following paragraphs, I will discuss what happened to their house model during and after the simulation and how the group proposed revisions to improve their house model.

First, during the simulation, since the teacher sprayed a lot of water on the models as one of the representations of a hurricane, the newspapers, which represent the roof of their house model, soaked up water and its sides bent down after the simulation, as shown in the picture below.

Figure 18: A picture of Nathan and his partner's house model after the simulation

When I asked Nathan about what happened to the roof, he reacted "Because it was like this [the sides of the roof were up before the simulation], we taped it like this, but probably when we were spraying it on the table and [the sides] probably fell off" (House model interview 04/06/22). At the moment of the interview, Nathan lifted the sides of the roof up to show that the sides of the roof were up before the simulation but “the roof went down” (House model interview 04/06/22) after the simulation due to the water from spray bottles. In the house model interview, when asked what material he used before but he would not use again. Nathan replied as “newspapers” (House model interview 04/06/22) and I asked why. He responded back, “Because then it's going to get soggy again, and then it's going to make it more wet” (House model interview 04/06/22). Since their roof got damp and the sides went down, Nathan decided not to use newspapers anymore. However, during his Scratch interview, Nathan changed his mind about using newspapers for the roof. When I reminded him of his statements on their roof during his house model interview, he replied that “I would use a newspaper. I'm going to tape Popsicle sticks on top of it. [...] so it could be strong and the newspapers won't get as wet” (Scratch interview 04/11/22). Although Nathan had planned not to use newspapers for the roof anymore, it seemed that he changed his mind and decided to strengthen the newspaper roof by supporting it with popsicle sticks. As a result, after the group noticed that the sides of their roof bent down after the simulation, they agreed to improve their roof by adding popsicle sticks on top of the newspaper so the newspaper will not get damp due to popsicle sticks’ protection and therefore do not bend. These all suggest that Nathan and his partner changed their material choice after the simulation and proposed a revision, which suggests an iterative process and a debugging proposal.

Second, a couple of revisions that the group proposed were for their foundation due to the fact that their house model fell on the popsicle sticks at the bottom of their model. During the house model interview, he stated that “it was moving and it was like this [tipping the house on the popsicle sticks at the bottom to show what happened during the simulation], but these two popsicle sticks [at the bottom] kept it up” (House model interview 04/06/22). Since their house fell on the popsicle sticks at the bottom, Nathan proposed a revision for their base during his Scratch interview and stated that "I would make a bigger base" because "[...] I saw the house moved" (Scratch interview 04/11/22). Nathan also made statements on their base and its revision during their house model interview as the following:

> Interviewer: If you had time to make this house model again, what would you do differently to make it stronger?
> Nathan: [...] this time I would add a bigger thing [base]. [...] And then add cups around the base.
> Interviewer: Why?
> Nathan: So wildfires and floods that can't get in.
> Interviewer: Do you mean you kind of surround your house with cups?
> Nathan: Mm-hmm.
> Interviewer: I see. What else would you do differently?
> Nathan: I would make a couple of these [popsicle sticks they made in their original house] around the house.
> Interviewer: To do what?
> Nathan: To make it extra protected.
> Interviewer: Where did you get that idea from?
> Nathan: Because I saw my other friend put Popsicle sticks on the bottom. I mean, it was strong.
>
> (House model interview 04/06/22)

Due to the result of simulation described above where their house shifted, Nathan proposed a solution where he said that he would make a bigger foundation by supporting it with popsicle sticks, which was an idea they got from another group in the classroom during the modeling and simulation project, and surround the house with cups against natural disasters to protect the base. These all suggest that Nathan went through an iterative process by reusing ideas from others and proposing revisions to their foundation.

Last, in the house model interview, I asked Nathan what he would not change if they had a chance to rebuild their house. His statements are as follows:

> Interviewer: What would you not change?
> Nathan: I would not change the alarm system.
> Interviewer: Why would you not change the alarm system?

> Nathan: Because it would be good for natural disasters because, well hurricanes you could see in the sky, but most other disasters you can't see. So this would be good.
> Interviewer: Oh, it alarms all the people. How does it work?
> Nathan: It was supposed to shake [shaking it in the cup] around and smack up and down. [...] And so it can make noise.
> Interviewer: Where did you get that idea from?
> Nathan: From a show that I watched because in a show that I watched and they had an alarm system, and then it was going up and down.
>
> (House model interview 04/06/22)

Nathan stated that he would keep the alarm system because he thought that most natural disasters are invisible as opposed to hurricanes. He added that he used the idea of an alarm system from a show he watched. Nathan statements demonstrate that he reused the idea of an alarm system from a show and would not revise the alarm system, which he thinks is a critical part of a house for safety reasons. In the following section, I will answer RQ2.

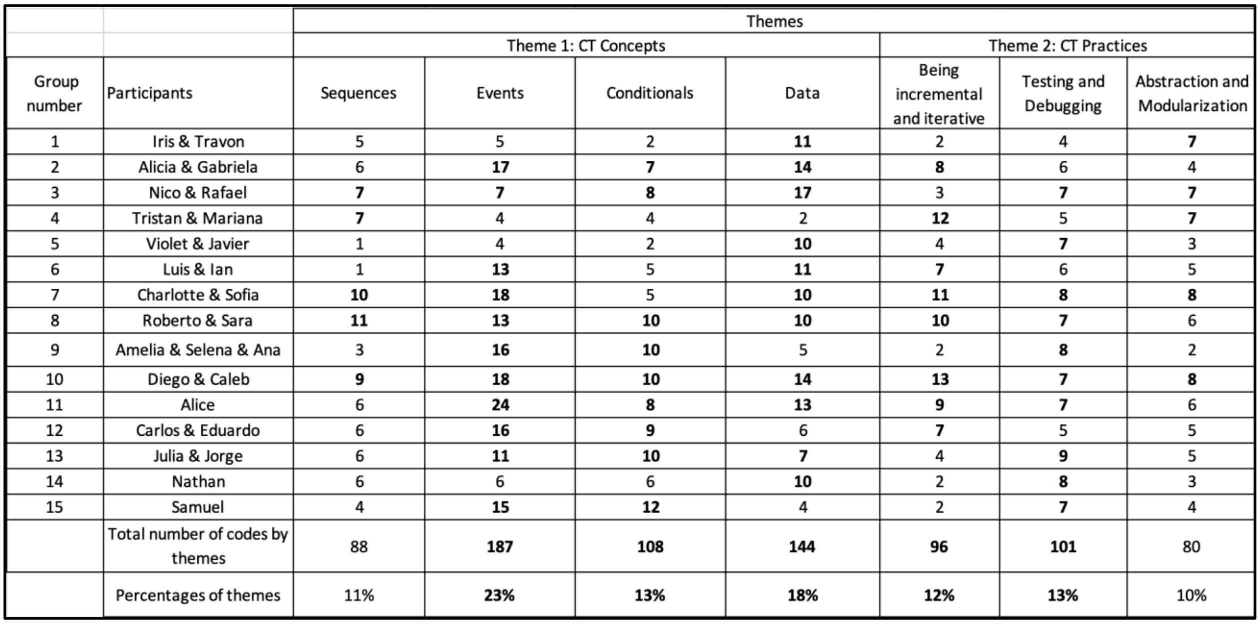
Computational Thinking Concepts and Practices through Scratch (RQ2)

In this section, RQ2 will be addressed, which is how students describe in Scratch the process they undertook. I found two broad themes to answer this RQ: CT concepts and CT practices. I will discuss the CT concepts and CT practices drawn from Brennan and Resnick's CT framework that participants developed based on their Scratch interviews and Scratch programs. The CT concepts they engaged in were sequences, events, conditionals, and data. The CT practices participants engaged in were being incremental and iterative, testing and debugging, and abstraction and modularization.

Table 8 below shows frequency of CT concepts and CT practices emerged from Scratch data: Scratch interviews, Scratch programs, and field notes (Please refer to Table 3 in the methods chapter to see what data sources were used to address which research question). CT concepts and CT practices are the two broadest themes of this data set. Specifically, the table shows group number and participants in the first two columns on the left, and CT concepts and

CT practices in the first top row. As for figures in the cells, those represent the frequency of sub themes for a group of students that emerged from the data. In the last two rows are the total number of codes by theme and their percentages.

Table 8: Frequency table of CT Concepts and Practices themes in Scratch Data



		Themes						
		Theme 1: CT Concepts				Theme 2: CT Practices		
Group number	Participants	Sequences	Events	Conditionals	Data	Being incremental and iterative	Testing and Debugging	Abstraction and Modularization
1	Iris & Travon	5	5	2	**11**	2	4	**7**
2	Alicia & Gabriela	6	**17**	**7**	**14**	**8**	6	4
3	Nico & Rafael	**7**	**7**	**8**	**17**	3	**7**	**7**
4	Tristan & Mariana	**7**	4	4	2	**12**	5	**7**
5	Violet & Javier	1	4	2	**10**	4	**7**	3
6	Luis & Ian	1	**13**	5	**11**	**7**	6	5
7	Charlotte & Sofia	**10**	**18**	5	**10**	**11**	**8**	**8**
8	Roberto & Sara	**11**	**13**	**10**	**10**	**10**	**7**	6
9	Amelia & Selena & Ana	3	**16**	**10**	5	2	**8**	2
10	Diego & Caleb	**9**	**18**	**10**	**14**	**13**	**7**	**8**
11	Alice	6	**24**	**8**	**13**	**9**	**7**	6
12	Carlos & Eduardo	6	**16**	**9**	6	**7**	5	5
13	Julia & Jorge	6	**11**	**10**	**7**	4	**9**	5
14	Nathan	6	6	6	**10**	2	**8**	3
15	Samuel	4	**15**	**12**	4	2	**7**	4
	Total number of codes by themes	88	**187**	**108**	**144**	**96**	**101**	80
	Percentages of themes	11%	**23%**	**13%**	**18%**	**12%**	**13%**	10%

In Table 8, figures bolded represent the number of codes higher than the average, which is 7 per theme for each group of participants (Please refer to the methods chapter for details), and the regular ones (no bold) represent the number of codes lower than the average. Additionally, total numbers in the bold in the last row are the official themes that emerged from the data. From this data in Table 8, I can deduce that the CT concepts of events, conditionals, and data are the official themes in CT concepts, and the CT practices of being incremental and iterative, testing and debugging are the official themes in CT practices based on the analysis of Scratch interviews and Scratch programs. However, the theme of the CT concept of sequences and the theme of the CT practice of abstraction and modularization was found as official themes because their finding was based on not only students' statements they made during their Scratch interviews but also their Scratch programs, which is part of Scratch data. Furthermore, the CT concepts of events and data came out higher than other themes. I assume this because almost all participants mainly used event blocks in order to trigger their scripts and sound blocks to record their data. Therefore, their discussions around these two blocks naturally more than other aspects of their Scratch programming. The higher frequency of these two findings were expected before analyzing the data due to the above-mentioned reasons.

Table 9 below shows CT concepts and CT practices, theory-driven codes utilized in data coding, codes, and example codes.

Table 9: Example codes for CT Concepts and Practices emerged from Scratch interviews

CT Items in Scratch Interviews	**Code Types**	**Codes**	**Example Codes**
CT Concepts in Scratch Interviews	Theory - Driven Codes	Sequences	Int: Here, you have a Makey Makey block (an event block) and a recording block? Which one should come first? Tristan: We should put this one [Makey Makey block] first. Int: What if I change the order of those two blocks? Can I put this one [recording block] first [on the top of the script]? Mariana: No. Tristan: No. Int: Why not? Tristan: Because there's no thing where it goes down a little bit [meaning that event block should come first].
		Events	Int: I see that you put a *when left arrow key pressed* block (an event block). What does it do there? Caleb: It plays the sound. — Nico: What we used to create that [script of Sprite 1] is we used the Makey Makey for the left arrow.
		Conditionals	Sofia: We clicked space. But also if we did [used] that [space key] for every single one [script], then it would all talk over each other.
		Data	Gabriela: Number three [sound block] was what we're going to use something different because it didn't work or stuff. —

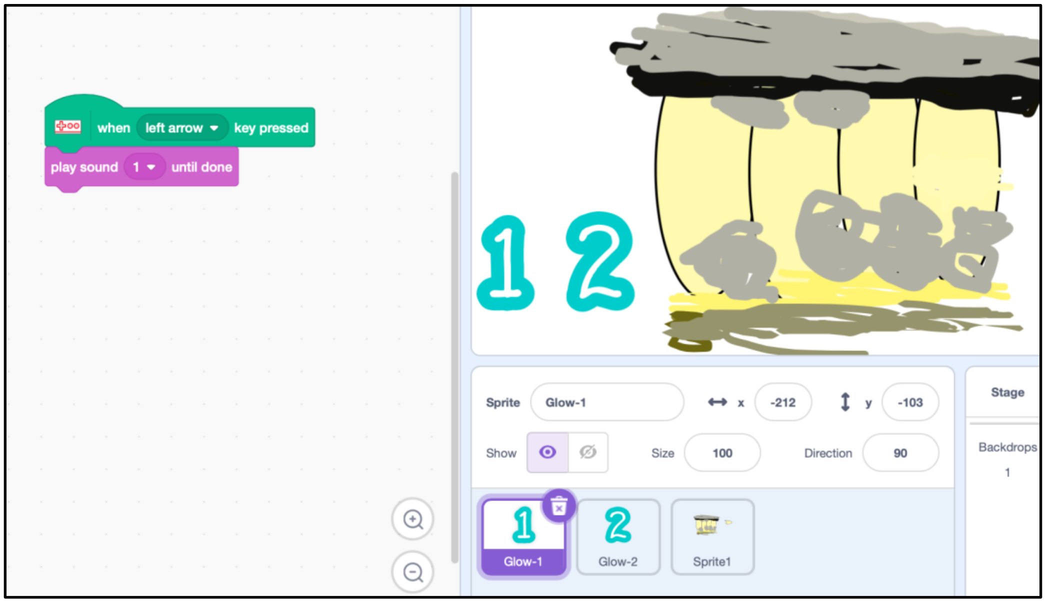
			Iris and Travon’s Scratch program: "We used index cards for the walls and Popsicle sticks for the base. Lastly, we used the cardboard for the roof."
CT Practices in Scratch Interviews	Theory-Driven Codes	Being Incremental and Iterative	Int: Can you tell me how you created the blocks of sprite 1? Mariana: Yeah. So we pressed this [My Blocks], and then we went to the Makey Makey and then pressed this [a Makey Makey event block]. Then we got these [Makey Makey blocks came up on Scratch and they added one of the Makey Makey event blocks] Int: Ok. And then what did you do? Mariana: We recorded our data into a sound block. So we dragged a Makey Makey first. And then changed it to the [one of the arrow options under Makey Makey event block]
		Testing and Debugging	Int: What happens if you use the left arrow for both? Jorge: It'll trigger them both at the same time. — Int: What did you click on to play it? Sofia: We click space. But also if we did that [space key] for every single one [script], then it would all talk over each other. That's why you need the [different] Makey Makey [blocks for each script].
		Abstraction and Modularization	Int: Could you have had just one Sprite [to code all in it]? Nico: No Rafael: No Int: What happened if you would do that? Nico: Because we would get mixed up. Which one's one, which one's two, which one's three [talking about sprites]. Int: Ok. Nico: That's why we separate. So that's why we got three Sprites. And then when number four, we just added that. — Int: Why do you have three different sprites? Alice: We have three because in the poster there were three boxes. So we have to make sure to get all the information from that. — Int: Why do you need three sprites in your project? Tristan: We will probably need three blocks to record everything.

Below, I will discuss the CT concepts followed by CT practices and provide examples from students’ Scratch interviews from the three classrooms where possible. While answering

RQ2, I mostly used participants' Scratch interviews data, their Scratch programs, and their poster board.

CT Concepts through Scratch

CT Concept of Sequences. Brennan and Resnick (2012) define the CT concept of sequencing as "a particular activity or task is expressed as a series of individual steps or instructions that can be executed by the computer" (p. 3) Most of the participants created three sprites and used two blocks in each script as seen in a screenshot below. Sprites in Scratch are characters that can be programmed to do an action such as moving. Blocks are the basic units of code in Scratch such as sound blocks. Below is Figure 19 to discuss this CT concept. One block was an event block, which triggers their script to play, and the event block they used was a Makey Makey block in most cases. The other was a sound block in which they recorded their data on their poster. Sequencing manifests in various aspects within the dataset, and I will elaborate on each of these in the subsequent paragraphs. Before discussing those aspects or ways, please refer to Table 3 in the methods chapter to see what data sources were used to address which research question.



```
when left arrow key pressed
play sound 1 until done
```

Sprite Glow-1 x -212 y -103

Show Size 100 Direction 90

Figure 19: A screenshot of a group of students' Scratch project

First, participants consistently placed an event block as the first block in their scripts, followed by a sound block. This indicates that they were aware of the requirement to start a script with an event block in order to trigger the subsequent actions as seen in the screenshot above. In other words, they understood the importance of the correct order, or sequence, of blocks to make their scripts function as intended. This suggests that the participants have developed a grasp of sequencing, a key component of computational thinking.

Second, during the Scratch interviews, I asked students if they could change the order of an event block and a sound block in order to understand if they knew the right order of blocks in a script. Below is a conversation with Carlos about the script of his sprite 3.

> Interviewer: Okay. Let me ask another question. Carlos, Look at this [the script of sprite 3]. You used an event block and then [a] switch costume [block] and then [a] play sound recording block. Do you see this event block? Can I put it [event block] under these [the last two blocks] or can I change their orders?
> Carlos: Maybe?
> I: Would it work if I change it?
> C: I don't think so.
> I: Why do you think so?

> Carlos: Because the event is supposed to go on the top, and then if you're doing a recording, the recording is supposed to go on the middle and then no, the recording is supposed to go in the middle. And then the final thing, whatever you wanna do, sensing, control, sound, looks, whatever you wanna do, you can put that at the bottom because it might work for you.
>
> (Group Interview 04/11/22)

While asking this question, I removed the event block from the script and placed it at the very bottom of the script (Figure 20). Carlos stated that if the event block in the script of his sprite 3 was put at the very bottom of the script, the script would not work. He argued that the event block is supposed to be on the top of a script so the script works properly and the remaining blocks in a script could be placed under an event block. At the end of his statements above, Carlos placed the event block back on top of the script and snapped the two blocks under it. Carlos's understanding suggests that there must be an order/sequence in a script and the first block in a script must be an event block.

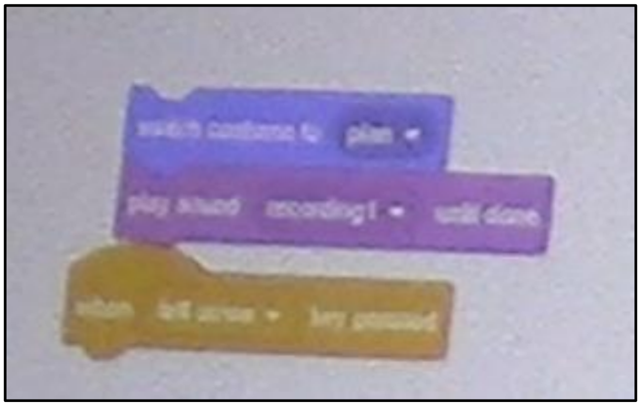
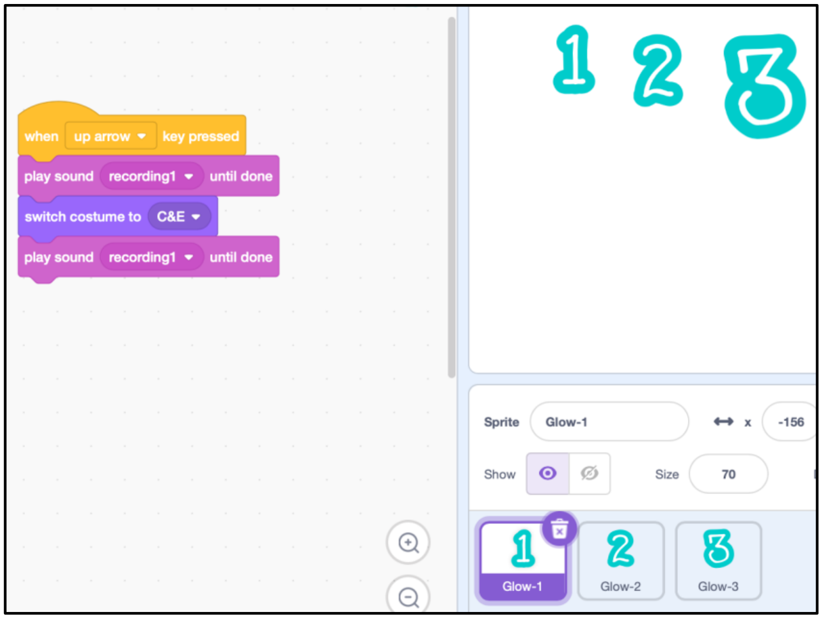


Figure 20: A screenshot of Carlos's script 3 in Scratch after changing the order of the blocks

Finally, students were given pre-created sprites that were numbered as 1, 2, and house (or 3) and they needed to at least have an event block and a sound block for each sprite so they could record the data in the three columns on their poster. They recorded the data in column 1 on their poster board (Figure 15) into sprite 1, which is the data on what materials to use and why the data on column 2 on the poster board into sprite 2, which is simulation data, and the data on column 3 on the poster board into sprite 3, which is the data on proposing a revision. Columns 1, 2, and 3 on the poster board show the right order of each activity. Considering the right order, when I asked students to play their Scratch project, most of the time they started playing their number 1 script, which discusses what materials they used and why, and then number 2 script, which is the simulation result, and number 3 script, which is their proposed revision. This suggests that there was an order/sequence that students followed while playing their Scratch project when asked. As a result, the three above-mentioned ways were related to the participants' understanding of the sequencing concept.

CT Concept of Events. I would like to state that the idea of sequences is different from events. While sequence is the idea of putting blocks in a script in a certain order, events are the blocks in Scratch that trigger a script or a program. Events are a CT concept defined as “one thing causing another thing to happen” (p. 4) by Brennan and Resnick (2012). In this case, event blocks that participants of the present study used are blocks that trigger a sound block. For example, for participants to play their sound blocks, in which they recorded their data on the poster board, they used different event blocks such as Makey Makey blocks. I would like to note that one finding I found is that almost all participants utilized an event block for each of their scripts, and events was the most frequent CT concept that participants discussed in their Scratch interviews. This could be because most participants used only two blocks in their scripts and event blocks were one of those two. Below is an example from Eduardo and Carlos’s Scratch project (Figure 21). Eduardo and Carlos used the when up arrow key pressed event block, a specific event block in Scratch, to control their sprite 1 in which they recorded the data of what materials they used and why and two more different event blocks for their other two scripts in which they recorded the data of simulation result and of how to improve their house design. In the following paragraphs, I will discuss another finding, which is participants’ understanding of functionality of event blocks they utilized in their Scratch project.



```
when up arrow key pressed
play sound recording1 until done
switch costume to C&E
play sound recording1 until done
```

Sprite Glow-1 ↔ x -156

Show Size 70

Figure 21: A screenshot of Eduardo and Carlos's script of their Sprite 1 on Scratch

During their Scratch project, I asked participants the function of the event blocks they used to see if they understood how event blocks they used work, as sometimes Scratch users copy others' projects without understanding how a certain block or script works. I found that most participants understood how the event blocks they used function. Therefore, it was through this question, their comprehension of cause and effect aspects of events were revealed. I would like to state that to represent all classrooms studied, I chose examples strategically from the three different classrooms—Mary's Classroom A, Mary's Classroom B, and Kimberly's classroom—wherever possible, to exemplify students' understanding of the CT concept of events. Below are a few examplars from the participants.

First, during their Scratch interview, when asked why Alicia and Gabriela had three different event blocks that are controlled by three different keys on the keyboard, Alicia replied "Because when you press a certain button [key] on the keyboard, it plays that certain thing" (Group interview 04/03/22). After her statement, I asked what are the function of the *when right* *arrow key pressed* block and the *when up arrow key pressed* block. She replied “Those are. Well those are the, so how I had to press the right arrow [on the keyboard]. Yeah, that controlled. That's what starts the Makey Makey [event block]” (Group interview 04/03/22). Alicia's responses reflect a clear understanding of event blocks and their functionality. She correctly associates each event block with a specific key on the keyboard, recognizing that when a particular key is pressed on the keyboard, the corresponding event block triggers a specific action. Her statements demonstrate her grasp of the cause-and-effect relationship between keyboard inputs and the event blocks' execution.

Second, the following conversation is from Alice about her understanding of event blocks.

> Interviewer: Okay let's go to the first one [Sprite 1] again. So I see in all of the three sprites, you have two blocks. One is the green block [Makey Makey event block] I see here and the other is a play sound block. What are those doing there? What are their functions?
> Alice: The Makey Makey. So it helps you, So when you start it, you don't just have to press the green flag, you press the arrows or the space or a letter [on keyboard].
> Interviewer: But if you had the green flag, so I mean would it make a difference in the project in terms of starting it?
> Alice: Not so much.
> Interviewer: Why?
> Alice: Because it wouldn't be so different because it's starting the project and it doesn't matter how you start the project. You just have to play it.
> Interviewer: What else you could have used to do the same thing?
> Alice: We could have used the space button or a letter [on the keyboard].
> (Group Interview 06/07/22)

Alice's understanding of event blocks in her Scratch project reflects a recognition of their functional versatility. She acknowledges that the Makey Makey event block allows for project initiation through multiple keyboard inputs rather than solely relying on the green flag, both a feature and an event block in Scratch used to start or initiate scripts or sequences of code. Her insight indicates a clear grasp of the concept that event blocks, such as the green flag or keyboard inputs, serve as triggers for starting the project and that the choice among them does not significantly impact the project's outcome. This showcases Alice's awareness of the interchangeable nature of these event triggers in achieving the same result within her Scratch project.

Finally, Luis and Ian also made statements on their understanding of event blocks.

> Interviewer: [...] I have another question. Why did you put that makey makey block there? What's its function?
>
> Luis: It plays when you press a button.
>
> Ian: It does something. It will do something when you put a piece of code on it. Like when space key pressed [he dragged a Makey Makey block to the coding area]. And if you put this [he also dragged a sound block] then it will actually play the sound.
>
> (Group Interview 04/03/22)

In this conversation, it's evident that Luis and Ian have a grasp of the Makey Makey block's function in their Scratch project. Luis associates the Makey Makey block with playing sounds upon button press, while Ian emphasizes its role as an activator for executing specific code. Ian's explanation further demonstrates that the Makey Makey block's integration with other blocks, like the sound block, produces desired outcomes, indicating a clear understanding of its functional interaction within the coding environment. In summary, Alicia, Alice, Luis, Ian and several other participants demonstrated varying levels of understanding of event blocks in their Scratch projects, ranging from recognizing their role in project initiation to comprehending their function in executing specific actions through code.

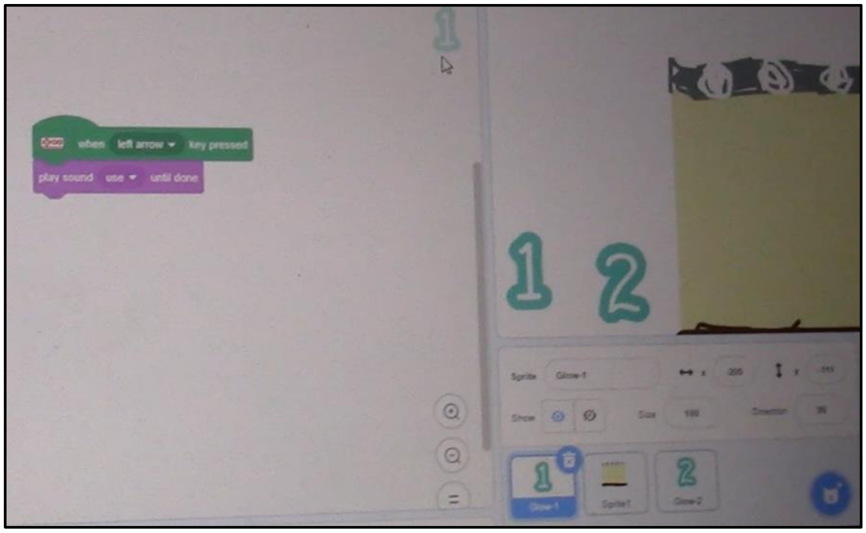
CT Concept of Conditionals. Conditional thinking, a CT concept explored in this study, was prominently observed among the participants. It is defined by Brennan and Resnick as “the ability to make decisions based on certain conditions” (p. 5) When discussing conditionals in their framework, Brennan and Resnick use an example of the *if* block or *if-then* block that their participants used, which are called conditional blocks. However, students in the current study did not use any conditional blocks in their Scratch programs but they were engaged with conditional thinking and I identified two different examples in participants’ conditional thinking statements they made during their Scratch interviews, which are implicit conditional thinking statements and explicit conditional thinking statements. Those two will be discussed below.

Implicit conditional thinking statements encompass instances where participants conveyed conditional thinking without explicitly using "if" statements. When prompted to identify the key that activates a designated script, nearly all participants demonstrated the correct keyboard input to initiate the intended action and expressed what key initiates that action. This implies their recognition of the connection between pressing a specific key and triggering a corresponding script, indicative of their grasp of conditional thinking. For instance, Gabriela noted, "Yeah, you have to press the up arrow" (Group interview 04/03/22) when instructed to play the script in sprite 3. This interpretation suggests an implicit if-then understanding: pressing the up arrow initiates the action. Similarly, Roberto's response offered insight. As he played their project, he remarked, "Yes. So this one [sprite 1] is just going to say the [what] materials were used," (Group interview 06/08/22) pressing the up arrow. Subsequently, he clarified that pressing the up arrow was the key for activation. Roberto's implication is that pressing the up arrow is a requisite condition to execute sprite 1. These implicit statements, made by Roberto, Gabriela, and the implicit conditional thinking statements that almost all other participants made, implicitly denote the conditions under which specific scripts are triggered.

Explicit conditional thinking statements arise when participants directly employ "if" statements. An illustration of this is evident in Charlotte and Sofia's Scratch interview, where they utilized the space key to activate a script. When queried about the key Sofia pressed, she explicitly remarked, "We clicked space. But also if we did [used] that [space key] for every single one [script], then it would all talk over each other. That's why you need the Makey Makey" (Group interview 06/03/22). Sofia's statement overtly demonstrates conditional thinking by suggesting that if the space key were employed for all scripts, undesired interference would occur, hence necessitating the use of the Makey Makey to avert this issue, as Makey Makey event blocks provide the capability of customizing various keys for distinct scripts. When prompted why they have three separate sprites, almost the same statement made by Sofia was made by many other participants including Amelia: “Because, if we do all left arrows then it will all play at the same time” (Group interview 06/06/22), Mariana: “If we pressed [had] the same button [for all three sprites], all of them would be going on” (Group interview 04/04/22), and Diego: “Because we don't wanna make it all the same. Because if we put space again, it'd probably play them together” (Group interview 06/06/22). These instances of participants, such as Amelia, Mariana, and Diego, explaining why they chose different keys for their sprites reveal their explicit conditional thinking. Their statements suggest that they recognized the potential consequence of using the same key for all sprites – triggering all scripts simultaneously. This understanding demonstrates their explicit consideration of conditions in programming, where the action of pressing a key serves as the condition for the specific script associated with that key to execute.

CT Concept of Data. “Data involves storing, retrieving, and updating values” (p. 6). All participants were engaged in various data activities through the engagement of the module, such as data collection, data recording, and data presentation. Data collection was done for the purpose of planning their house models so it was discussed under “CT practices in planning”. In other words, data collection as a CT practice is different from the CT concept of data, which I discuss here. By data I mean the data students recorded into sound blocks in their Scratch programs where they talked about what materials they used in one sprite, what happened to their house models during simulation in another sprite, and the proposing a revision/improvement to their house models in another sprite. This idea will be discussed under data recording below.

Data Recording. As discussed before, students used a poster board to record their data from the house model activity. As mentioned before, a poster board was divided into three columns: Column 1 was for which materials to use, column 2 was for simulation results, column 3 was for revision data. After recording this data on their poster, students mirrored those columns into three sound blocks under three different sprites in Scratch. Below is an example of the data on which materials to use from Alice and her partner in the Scratch program (Figure 22). The data that Alice and her partner entered into the sound block was from column 1 on their poster board. When asked to play that sound block during their Scratch interview, they played it: “We used cardboard, popsicle sticks, tinfoil, cotton balls, paper towel, pipe cleaners, tape, and index cards. We used these materials because they are strong” (Scratch interview 06/07/22). Alice and her partner, like almost all other participants in groups in the study, entered their different data sets into three separate sound blocks under three distinct sprites. Please note that a couple of students deleted their recordings mistakenly so they could not play their recordings during their Scratch interviews.



```
when left arrow key pressed
play sound use until done
```

Figure 22: Alice and her partner’s script of sprite 1 in her Scratch program

Data Presentation/Sharing. This subcategory of the CT concept of data concerns students' data presentation through Makey Makey. By data it means the entire Scratch program. To enhance their data presentation, students employed Makey Makey technology alongside their Scratch projects. By connecting a Makey Makey to their poster board, they transformed it into an interactive display, allowing them to present their Scratch programs in a captivating manner. Below is an example of Makey Makey connected to each column on their poster board and Scratch program from Violet and Javier (Figure 23).

Students shared their Scratch programs with the classroom in an interactive showcase by using Makey Makey. This combination of Makey Makey technology and Scratch not only made their presentations more interactive but also enhanced the depth of understanding and engagement among their peers. Not only Violet and Javier but all groups used Makey Makey to present their Scratch work.

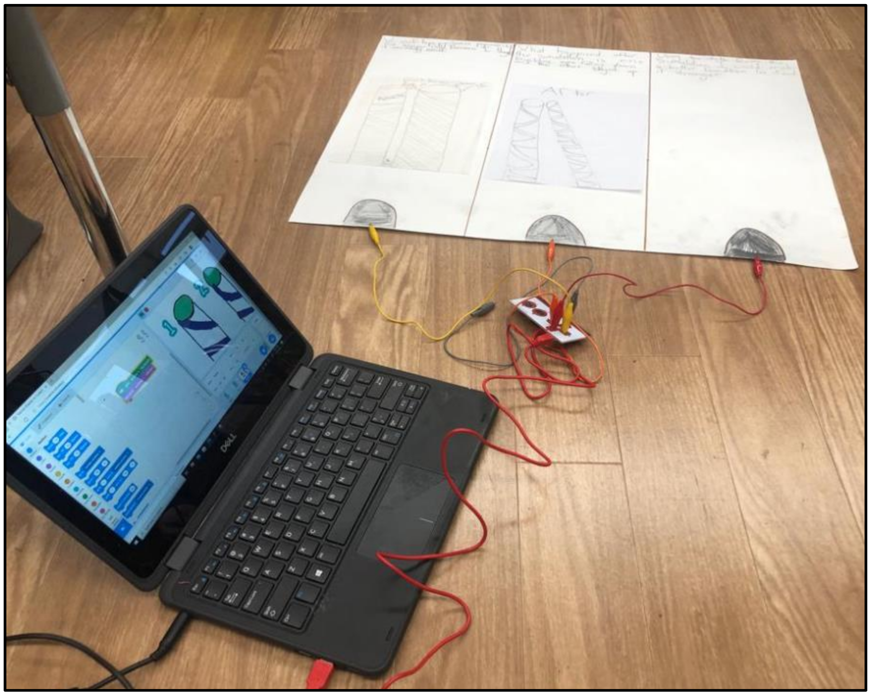


Figure 23: Violet and Javier's Makey Makey

Computational Thinking Practices through Scratch

The CT practices discussed here are the practices they developed in their Scratch programs based on their Scratch data. In the following paragraphs, the CT practices of being incremental and iterative (tinkering), testing and debugging, and abstraction and modularization will be discussed.

Being Incremental and Iterative. Brennan and Resnick (2012) define this CT practice as follows:

> Designing a project is not a clean, sequential process of first identifying a concept for a project, then developing a plan for the design, and then implementing the design in code. It is an adaptive process, one in which the plan might change in response to approaching a solution in small steps." (p. 7)

Making a Scratch program or programming in general necessitates one to go through step-by-step and iterative fashion, which is also called tinkering. Participants in the current study were engaged in this CT practice not only when creating their house model but also when making their Scratch projects. In the following paragraph, I will provide a couple of examples from Scratch interviews illustrating how participants developed this CT practice during their Scratch interview.

Tristan and Mariana were a group from Mary's classroom A. When asked which of their three sprites they created first, Tristan and Mariana replied "the house [sprite 3]" (Group interview 04/04/22) and then Mariana said "And then sprite 1 and then sprite 2" (Group interview 04/04/22). After that, we started discussing how they created the blocks under sprite 1.

> Interviewer: Can you tell me how you created the blocks of sprite 1?
> Tristan: Yeah. So we pressed this [My Blocks], and then we went to the Makey Makey and then pressed this [a Makey Makey event block]. Then we got these [Makey Makey blocks came up on Scratch and they added one of the Makey Makey event blocks]
> Interviewer: Ok. And then what did you do?

> Tristan: We recorded our data into a sound block. So we dragged a Makey Makey first. And then changed it to the [one of the arrow options under Makey Makey event block]
>
> (Group interview 04/04/22)

In the conversation, Tristan explained to the interviewer the step-by-step process they used in creating their Scratch program. They began by detailing the sequence in which they created their sprites, starting with "the house" sprite. Tristan then elaborated on how they went on to develop the blocks under sprite 1, describing their use of the “My Blocks” function, integrating “Makey Makey” event blocks, and subsequently recording their data into a sound block. This discussion underscored the incremental and iterative nature of their work, as they navigated through the Scratch platform in a systematic manner. Such a methodical approach was not unique to just Tristan and Mariana; indeed, all participants in the study showcased their engagement in the CT practice of being incremental and iterative.

Similar to Tristan’s statements, when asked about how they coded the two blocks of a sprite to represent the data of materials used, Roberto gave the following response.

> [...] so Makey Makey, if we click onto that, it shows the Makey Makey. And then when we put [dragged a Makey Makey event block to the coding area at the time of the interview] that there, we just had to record [he went to “Sounds” in Scratch] and then go to sound and then drag that [recorded sound] here and then we do that.
>
> (Group Interview 06/08/22)

Based on Roberto's explanations, it's evident that he approached the task in a step-by-step manner, demonstrating the CT practice of being incremental. Each action built upon the previous one, from selecting the Makey Makey event block to recording and finally integrating the sound. This process underscores the iterative nature of their approach, refining and building as they progressed. Such a method is at the core of computational thinking, showcasing the value of incremental and iterative practices.

Testing and Debugging. Brennan and Resnick (2012) define this CT practice as follows: “Things rarely (if ever) work just as imagined; it is critical for designers to develop strategies for dealing with – and anticipating – problems.” (p. 7)

One common understanding among participants is the recognition of the need to employ distinct event blocks for each of their scripts in Scratch. Most participants understood that using different blocks is crucial to prevent the simultaneous activation of multiple scripts, which would occur if the same event block is assigned to different scripts. This understanding is crucial to avoid potential errors in their Scratch programs, ensuring that each script operates independently and is triggered by its designated event block. The participants in this study were engaged with this CT practice in a way that they avoided using the same event block to avoid triggering multiple scripts at the same time, which would happen if different scripts are linked to the same event block. In the following paragraphs, I will provide three examples of this mentioned way of avoiding a programming mistake from each classroom I collected data from.

First example is that Sofia, a student from Mary’s Classroom B, exhibited a clear understanding of this potential programming mistake during her Scratch interview. When she demonstrated playing one of their blocks, I inquired about which block she activated to initiate it. She explained, “We click space. But also if we did that [space key] for every single one [script], then it would all talk over each other. That's why you need the [different] Makey Makey [blocks for each script]” (Group interview 06/03/22). When the conversation turned to Sofia and Charlotte discussing which key triggers which script, I posed a question about the necessity of having distinct keys, to which Sofia responded, “Yes. Because if they are not different, they would talk and talk over each other” (Group interview 06/03/22). Sofia’s response showcased their grasp of the concept that utilizing varied event blocks for distinct scripts is imperative to prevent overlapping of scripts during execution.

Second example is that Mariana, a student from Mary's classroom A, also mentioned the same potential programming error during their Scratch interview when asked about the need of using different arrows for their different scripts. She said that "If we pressed [coded] the same button [for all three sprites], all of them would be going on" (Group interview 04/04/22). Mariana's observation distinctly illustrates another instance of students' awareness and comprehension of the programming logic within Scratch. That is, they clearly recognized that assigning the same button to multiple scripts would result in concurrent activation, disrupting the intended sequential flow and individual functionality of each script within the program.

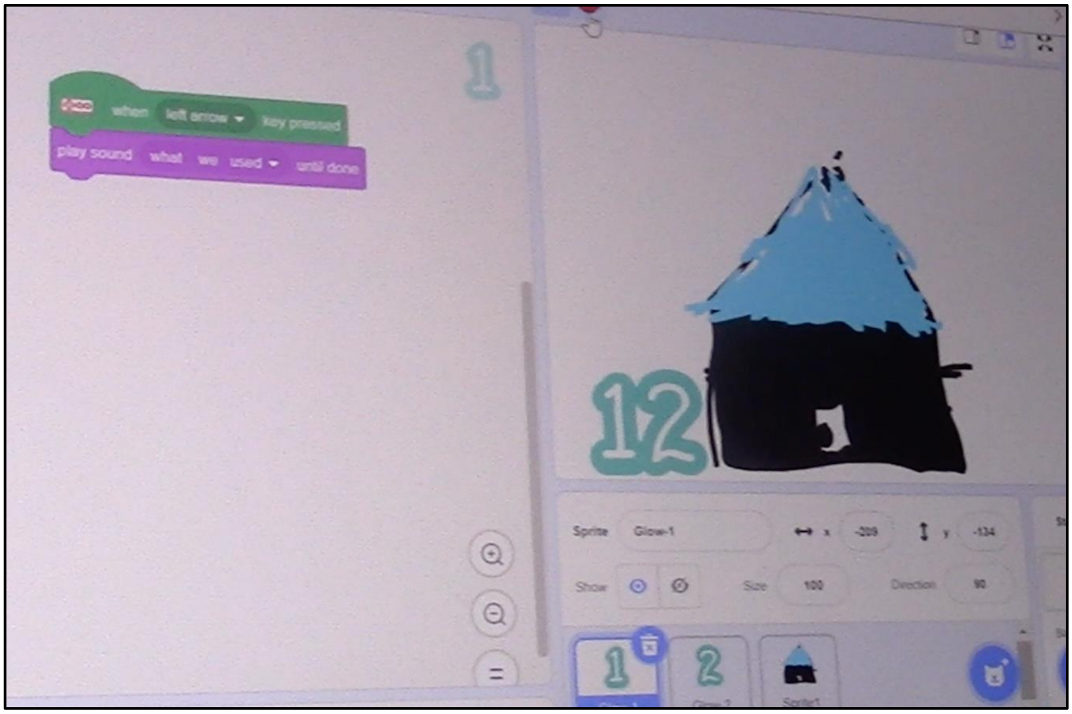
Final example of this said understanding is from Julia and Jorge, a group of two from Kimberly's classroom. Below is a conversation with them during their Scratch interview that showcases their understanding of the necessity of using different event blocks for different scripts to avoid simultaneous execution of multiple scripts.

> Interviewer: How about this one? You don't have any coding in this one?
> Julia: Because we didn't have time.
> Interviewer: If you had time, what would you use?
> Julia: We would put events again and we wouldn't do left [arrow] this time.
> Interviewer: Why would you not do a left arrow?
> Julia: Because [no response]
> Interviewer: Why do you think she wouldn't put the left arrow?
> Jorge: So then when you press [the] left arrow, you would activate it.
> Interviewer: Why would you not put [a] left arrow?
> Julia: Because we already did it for this [script of their sprite one].
> Interviewer: You already did it. What happens if you use [the] left arrow for both?
> Jorge: That arrow just trigger[s] both.
> Julia: Yes.
> Interviewer: What's that?
> Jorge: It'll trigger them both at the same time.
>
> (Group interview 04/08/22)

Sofia, Mariana, and Jorge and Julia’s responses underscore the students’ insightful grasp of the necessity to allocate distinct event blocks for separate scripts, a crucial understanding to preclude the simultaneous execution of multiple scripts, ensuring that each script operates independently and without interference, thus maintaining the integrity of their individual functions in the programming environment.

Abstraction and Modularization. This CT practice is defined by Brennan and Resnick as “building something large by putting together collections of smaller parts” (p. 9). As discussed before, groups were given a poster board to plan their project, document data from research and simulation, divided into three columns: material planning and house model drawing with labels (column 1), simulation data recording and results drawing (column 2), and proposing a revision(s) (column 3). This was discussed as the CT practice of decomposition in planning. Students were engaged in decomposition not only through the poster board but also in their Scratch program. They created three sprites on Scratch each of which corresponded to each step on the poster board. Although this practice was the least frequent practice among others (See the frequency table of Scratch interviews in Table 8) based on their Scratch interviews, almost all students created three sprites that mirrored each column on their poster board, which makes this practice a theme based on students’ Scratch programs. Below is a screenshot of a group of student’s Scratch project (Figure 24).

Second graders created a Scratch project using two different blocks: an event block and a sound block. They utilized three sprites to record their project components, by using the same quantity as the columns on their poster board. Sprite 1 documented the materials used and why, Sprite 2 recorded observations during the simulation of their house model, and Sprite 3 proposed revisions based on simulation results and depicted the proposed house model's picture. This highlights how students effectively decomposed their Scratch project into three parts by using three sprites, which are characters on Scratch one can program, aligning with their poster board columns. During the Scratch interview, when asked about using three sprites for their project, Nico replied, "Because we would get mixed up. Which one's one, which one's two, which one's three. [...] That's why we separate. So that's why we got three Sprites" (Group interview 04/03/22). Following the need to mirror each column of their poster board on the Scratch program, Nico and Rafael opted to create three separate sprites. Similarly, Tristan and Mariana echoed similar statements during their Scratch interview. Mariana: “We used three different sprites because there were three boxes in the poster. So we only do three [sprites] (Group interview 04/04/22). Tristan: “We will probably need three blocks to record everything.” (Group interview 04/04/22). In summary, the observations and participants' Scratch programs collectively indicate that participants strategically employed three distinct sprites within their Scratch projects to replicate each individual component detailed within the poster board. This practice underscores their deliberate engagement in the process of decomposition, effectively breaking the task down into manageable elements through the structure of their Scratch programs.



when left arrow key pressed
play sound what we used until done

Sprite Glow-1 x -209 y -134
Show Size 100 Direction 90

Figure 24: A screenshot of a group of students' Scratch project

In the following pages, the discussion chapter will be discussed followed by implications, limitations, and conclusion of the study.

CHAPTER V

DISCUSSION AND RECOMMENDATIONS FOR FUTURE RESEARCH

The purpose of this study was to investigate what CT concepts and practices 2nd graders developed in the creation of artifacts such as house models and Scratch programs through the engagement of a CS/CT integrated module into science and engineering. In this chapter, the synthesis of findings will be discussed, organized by research questions and in relation to the literature.

This study was grounded on Papert's (1990) constructionism and the Brennan and Resnick's (2012) CT framework. The participants of the present dissertation constructed house models and Scratch artifacts through engaging with a CS/CT module that was infused into science and engineering. I found that the participants developed the following CT practices while planning, building, testing, and proposing a revision to their house models: abstraction and modularization, being incremental and iterative, testing, debugging, and data collection. These findings, except for data collection, are consistent with Brennan and Resnick's (2012) study. Second graders also engaged in CT concepts and CT practices in their Scratch programs outlined in the Brennan and Resnick' CT framework: the CT concepts of sequences, events, conditional thinking, data, and the CT practices of being incremental and iterative, testing and debugging, and abstraction and modularization. In the following paragraphs, I will discuss each of these findings related to the literature and offer future recommendations starting with the findings regarding house model data.

CT Practices in House Model Data

The analysis of house model data revealed that participants developed the CT practices of abstraction and modularization/decomposition, being incremental and iterative, testing, debugging, and data collection. In the following paragraphs, each of these practices will be discussed organized by the research questions and in relation to the literature.

First, let me start by addressing RQ1a: Which CT practices 2nd graders engaged in describing their understanding of modeling and simulation. Second graders engaged in the CT practices of abstraction and decomposition and testing in describing their understanding of modeling and simulation. Abstraction and decomposition are key skills in computational thinking, as emphasized by several researchers (Kramer, 2007; Wing, 2008; Brennan and Resnick, 2012; Grover & Pea, 2013). Students showed their understanding of abstraction through their grasp of models. For instance, they understood that a toy car is a simplified representation of a real car, showing their ability to conceptualize models as representations of more complex real-world entities. This indicates that the students effectively engaged in computational thinking practices of abstraction and decomposition when they were engaged with the CT integrated science and engineering module. As for testing/simulation practice, which is part of RQ1a, students displayed comprehension and application of the CT practice of testing through simulations, actively experimenting with different materials to determine the sturdiest construction for house models. Some students even embodied this practice by physically simulating disaster scenarios, such as tornadoes, to assess the durability of their material choices, thereby internalizing the concept of simulation as a critical tool for evaluating the strength and feasibility of their engineered solutions. Testing students' house models with a physical simulation in the classroom provided an important opportunity for young students not only to see how strong their models were but also offered a hands-on approach to learning and an opportunity to reflect on the results of the simulation on their original designs. This finding aligns with current educational research that emphasizes experiential learning and the application of theoretical concepts such as testing/simulation in practical scenarios (Yang et al., 2018; Ehsan et al., 2021; Cabrera et al., 2023). This reflects the growing recognition in the literature of the importance of hands-on, experiential learning in developing problem-solving skills in early education. It also underscores the value of incorporating simulation-based activities in the classroom to enhance students' understanding and application of the CT practice of testing.

Second, students' engagement in the CT practices of abstraction and decomposition regarding RQ1a was discussed above. However, second graders also engaged in these two practices in planning their house models, which addresses part of RQ1b: What CT practices 2nd graders developed in planning their house models. Second graders used a poster board divided into three columns for planning, simulation, and revision, which not only facilitated a structured approach but also allowed the students to conceptually understand models as simplified representations of real-world entities. This aligns with the literature emphasizing the importance of abstraction and decomposition in early computational thinking skills development (Wing, 2008; Brennan and Resnick, 2012; Grover & Pea, 2013; Sengupta et al., 2013). As a result, findings corroborate the notion that young children can engage in CT and illuminate conditions that enable this to happen. This finding is consistent with the literature (Kafai et al., 2014; Hynes et al., 2016; Yang et al., 2019; Yang et al., 2020) where unplugged activities in CT integrated science and engineering contexts, such as engineering design like a puppy house, designing paper baskets or designing toy box organizers, designing earthquake-resistant bridges, and an airplane design activity, enabled students to engage in the CT practices of abstraction and decomposition. This finding also suggests that educators can effectively integrate these CT practices into early childhood education by creating tangible activities such as house modeling and using tools like poster boards that visually and conceptually divide complex tasks. Future research should explore the effectiveness of modeling activities and various instructional tools and methods in teaching abstraction and decomposition to different age groups in CT integrated science and engineering contexts. Additionally, investigating how the understanding of abstraction evolves as children grow could provide valuable insights into the development of computational thinking skills over time.

Third, one CT practice second graders developed in planning their house models was abstraction and decomposition through their use of poster boards and this practice was discussed above. However, second graders also engaged in the CT practice of data collection as a computational thinking practice, which again addresses RQ1b: What CT practices 2nd graders developed in planning their house models. Although not recognized as a CT practice in the Brennan and Resnick's (2012) CT framework, the activity of researching natural disasters before building house models highlighted its relevance. The students' engagement in gathering and utilizing information as part of their project work underscores the importance of data collection in the context of CT. Therefore, this finding extends the current understanding of CT practices defined by Brennan and Resnick (2012). There are several studies in the CT integrated science and engineering literature that considered data collection a computational thinking practice based on the framework they used (Yang et al., 2019; Waterman et al., 2020). The results of data analysis suggest that activities that require research such as natural disasters research in this case could reveal the CT practice of data collection. Further research is needed to obtain more reliable results regarding this finding. It points towards the importance of integrating research-based activities in the curriculum, enabling students to engage in data collection as part of their learning process.

Fourth, in this paragraph, I will synthesize the findings of RQ1c: To what extent did participants' initial house models survive from the simulation/test? This question was meant to evaluate the degree of durability of second graders' house models through a designed rubric and what this looks like in varying levels of house models in simulations. The statistics according to the designed rubric indicate that the majority of participants successfully constructed robust foundations and walls for their house models, but encountered more challenges in creating sturdy roofs, with a significant proportion still in the process of development or deemed incomplete. The application of a specifically designed rubric to evaluate their house models during a hurricane simulation highlighted the varying levels of model durability and offered a practical assessment of their problem-solving skills. The implications of these findings emphasize the importance of hands-on, experiential learning in fostering computational thinking skills in young learners, particularly in understanding the cause and effect of their design choices.

Fifth, another research question was RQ1d: Which CT practices second graders engaged in describing and understanding their material choices they made in creating their house models and proposing a revision. The findings revealed that second graders developed CT practices characterized by incremental building and debugging during the creation and revision of their house models. This finding is consistent with the literature (Kafai et al. 2014; Yang et al., 2018; Litts et al., 2020). This iterative process of building, evaluating, and making improvements reflects key aspects of problem-solving and adaptability in computational thinking. The discussions and decisions about materials, and the subsequent identification and resolution of issues during simulations, are indicative of a deep engagement with the CT practices of iteration and the debugging process, two core components of computational thinking. These findings contribute to the existing body of literature on computational thinking by illustrating how iterative processes and debugging can manifest in young learners' activities, supporting the notion that these skills can be developed from an early age. The study suggests the effectiveness of incorporating iterative and debugging processes in educational activities, which could help foster computational thinking or broadly problem-solving skills in young students. Furthermore, while students suggested modifications to their house models after running simulations, it's crucial for future studies to allow students to actually implement these changes and then retest the models. This step is important not just to verify the effectiveness of their proposed revisions, but also to help them realize that the process of testing and debugging is iterative in nature. This experience would reinforce their understanding of the iterative aspect of building a house model.

To conclude, the findings discussed above regarding the entire process of house model construction were to answer RQ1, which is what CT practices did second graders develop through the engagement with the CT integrated science and engineering module while planning, building, testing, and proposing a revision to their house models, and how did they emerge? Those were CT practices of abstraction and decomposition, being incremental and iterative, testing, debugging, and data collection. At this point, I would like to discuss the idea of adapting the Brennan and Resnick's CT framework to the house modeling activity and what we learned from this application. The inclusion of an engineering project such as house modeling within the scope of computational thinking (CT) provides an unplugged context for students to apply CT practices in a tangible way. This approach aligns with constructionist learning theories, where knowledge is built through the act of making. By examining house modeling through a CT lens, we gain insight into how young learners approach engineering challenges, breaking down complex problems (like designing a house that can withstand natural disasters) into manageable components, and applying testing and being incremental and iterative practices. This hands-on project allows students to embody CT concepts without the need for digital technology, thus reinforcing the universality and adaptability of CT practices across different domains. What we learn here is multifaceted: not only do we observe the application of CT practices in a new, less traditional domain, but we also see the potential for CT to enhance understanding in a core STEM area like engineering. Acknowledging the role of CT in this context underscores its value as a cognitive tool that transcends its computational origins, becoming a powerful lens for analyzing and improving STEM education more broadly. In the subsequent paragraphs, I will synthesize the findings of RQ2, which is how participants describe in Scratch the process they undertook, in relation to the literature and the implications and future recommendations of these findings for the field.

CT Concepts and Practices in Scratch Data

The analysis of Scratch data revealed that participants developed the CT concepts of sequences, events, conditionals, and data, and the CT practices of being incremental and iterative (tinkering), testing and debugging, and abstraction and modularization/decomposition. In the following paragraphs, each of these findings of the CT concepts will be discussed followed by the CT practices in relation to the literature and the implications of these findings for the field.

CT Concepts through Scratch

The sequence of activities in this study, which transitioned from an unplugged house model activity to the Scratch environment, suggests that certain computational thinking (CT) concepts may become more comprehensible to students when introduced after hands-on exploration. This sequencing approach and its impact on CT concept accessibility merit further investigation in future research. In the upcoming paragraphs, I will explore these CT concepts in the context of existing literature and RQ2, which is How did second graders describe in Scratch the process they undertook, followed by the CT practices.

First, the sequencing data illustrates a clear understanding of sequencing among second graders, as evidenced through their use of Scratch. The consistent placement of an event block followed by a sound block in their scripts indicates a grasp of the correct order necessary for the scripts to function properly. This finding aligns with the literature (Brennan and Resnick, 2012; Lee et al., 2014; Arik & Topcu, 2022). The key takeaway is that young learners demonstrate a notable capacity for understanding and applying this fundamental programming concept.

Second, from the findings on the CT concept of events, it's evident that students had a strong understanding and application of event blocks in Scratch programming. They recognized the cause-and-effect dynamics in programming, such as how pressing a keyboard key triggers specific program responses like playing a sound. Students like Alicia, Alice, Luis, and Ian not only explained event blocks' functions but also understood the adaptability of triggers, like keyboard keys or the green flag. This demonstrates their deep comprehension of programming events, an important component of computational thinking, showing their capability to grasp and apply complex programming concepts effectively at a young age.

Third, the students effectively demonstrated conditional thinking in their Scratch programming. Although they didn't use traditional conditional blocks like "if" statements, as the lesson design did not require them to use “if-then” blocks, they showed an understanding of conditions through their responses and actions. For example, they recognized that pressing specific keys on the keyboard would activate certain scripts. This understanding of conditional relationships, where a particular action leads to a specific outcome, is a fundamental aspect of computational thinking. The students' ability to apply this principle, understanding the consequences of their choices in programming, reflects a solid grasp of decision-making and logical reasoning in a computational context. This aligns with the current educational research (Sengupta et al., 2013; Lee et al., 2014, Aksit and Wiebe, 2020; Luo et al., 2020).

Fourth, data was another CT concept that second graders engaged in. The key takeaway from this is the comprehensive engagement of students with data activities, encompassing data recording and presentation. Students effectively used Scratch to record data related to their house model projects, entering details in the three columns on their posters about materials used, simulation results, and proposed revisions into different sound blocks. This process highlights their understanding of data management and representation. This finding is consistent with the literature (Weintrop et al., 2016; Waterman et al., 2020; Cabrera et al., 2023). Additionally, the use of Makey Makey technology for presenting their Scratch programs transformed their poster boards into interactive displays, enhancing their presentations' interactivity and engagement. This indicates not only the students' ability to handle and communicate data creatively but also their understanding of integrating technology to share and discuss their findings, a vital aspect of computational thinking.

Finally, in the context of this study, it is noteworthy that three additional computational thinking (CT) concepts from Brennan and Resnick’s (2012) framework—loops, parallelism, and operators—did not manifest in the analyzed Scratch data. This observation can be attributed to the design of the lesson plan, which appears to have been intentionally structured to align with the developmental stage of second-grade students. The primary objective of the lessons was to facilitate students in effectively presenting their data on poster boards, a goal achieved through the utilization of straightforward Scratch blocks, specifically event and sound blocks. These blocks were selected for their simplicity and efficacy in meeting the module's objectives. In essence, the need for students to represent their data in Scratch was adequately fulfilled without recourse to more complex blocks like operators or looping blocks. The employment of sound blocks, in particular, was deemed highly suitable for second graders, offering an accessible means to accomplish the module's requirements. The absence of loops, parallelism, and operators in the students' Scratch projects is thus reflective of a pedagogical decision to tailor the computational tasks to the students' age and cognitive capabilities, ensuring both the clarity and the accessibility of the learning process. In the following paragraphs, the CT practices of being incremental and iterative (tinkering/experimenting), testing and debugging, and abstraction and modularization will be discussed in order.

CT Practices through Scratch

The Scratch environment, and resulting practices, followed unplugged activity involving the house model, suggesting the possibility that some CT practices may be accessible when positioned within a sequence of activity and coming after unplugged exploration, something future research ought to consider. In the upcoming paragraphs, I will explore those CT practices in the context of existing literature, which is still under RQ2: How did second graders describe in Scratch the process they undertook?

First, the findings on the CT practice of being incremental and iterative show that students effectively engaged in a step-by-step, adaptive process while creating their Scratch projects. This aligns with Maloney et al. (2010) and Resnick et al. (2009), who emphasize the 'tinkerability' of Scratch and its facilitation of a hands-on, bottom-up approach to learning. The students' methodical process of building their projects, exemplified by descriptions of many second graders such Tristan, Mariana, and Roberto, mirrors this approach. They not only assembled and tested small chunks of code but also refined and built upon them, embodying the essence of tinkering. There are several studies in the literature that explored this finding (Brennan and Resnick, 2012, Lee et al., 2014; Litts et al., 2020; Luo et al., 2020; Cabrera et al., 2023). This particular finding in the dissertation was derived from the second graders' descriptions of their Scratch projects during interviews, where they predominantly utilized only two types of blocks: event and sound blocks. It is posited that Scratch programs incorporating a broader range or more complex blocks might yield more conclusive insights into this CT practice. Additionally, alternative methodologies could provide a deeper understanding of this CT practice in younger students. These could include recording the complete process of students creating their Scratch programs or employing the 'think aloud' method, which is instrumental in elucidating cognitive processes during task execution. Consequently, further research employing these methods is recommended to enhance the reliability and depth of findings related to this CT practice.

Second, another CT practice that was found in second graders' Scratch data was testing and debugging. The key takeaway from this finding is that students demonstrated a critical understanding of avoiding potential programming errors in Scratch by using distinct event blocks for each script. This awareness, as shown by many students like Sofia, Mariana, Julia, and Jorge, is crucial in preventing the overlapping and simultaneous activation of scripts, ensuring that each script operates independently. Their grasp of this concept highlights an important aspect of computational thinking: anticipating and strategizing to avoid potential problems in programming. This finding is consistent with the educational research where students developed the CT practice of testing and debugging in their programming (Weintrop et al., 2016; Aksit and Wiebe, 2020; Litts et al., 2020). The implication of this finding is significant in the context of teaching young learners programming skills. It suggests that even at a young age, students are capable of understanding and applying fundamental programming principles, such as the importance of unique triggers for separate processes, which is essential for the development of computational thinking.

Third, the last CT practice that was found in second graders' Scratch data was abstraction and modularization. This CT practice was also found by other researchers in the literature (Hynes et al., 2016; Cateté et al., 2018; Cabrera et al., 2023). An important takeaway from this finding is the strategic use of modularization/decomposition by second graders in their Scratch projects. Students effectively broke down the complex task of their house model project into manageable parts by creating three distinct sprites in Scratch, each mirroring a specific column of their poster board. This approach reflects a sophisticated understanding of modularization, where they organized their project into smaller, distinct components for clarity and ease of management. The implications of this finding are significant for computational thinking education. It demonstrates that young learners are capable of applying advanced programming concepts, like abstraction and modularization, in a practical context. This indicates their potential to grasp and implement complex problem-solving strategies, a crucial skill in developing computational proficiency from an early age.

Fourth, in the context of this dissertation, it is important to note the absence of the CT practice of reusing and remixing, as outlined in Brennan and Resnick's (2012) framework, within the Scratch data analyzed. This absence is largely due to the specific design of the lesson plan, which did not incorporate elements that would encourage students to engage in reusing and remixing. The lesson's structure focused on having students create simple Scratch programs using basic elements like event and sound blocks, thereby creating a context where reusing and remixing were not necessary components. Furthermore, it is posited that developing the practice of reusing and remixing requires a certain level of familiarity with the tool in use, such as an in-depth understanding of the Scratch online community. This community is recognized by Brennan and Resnick (2012) as a platform that offers ample opportunities for reusing and remixing, facilitating the creation of more complex projects by building upon existing ideas and code. However, as the integration of the Scratch online community was not a component of the lesson plan, the students did not engage in the practice of reusing and remixing. Consequently, this CT practice did not emerge in the data for the reasons outlined above.

Implications

In this section, I will discuss the implications related to theory and curriculum/instruction in order.

First, in this dissertation, I applied the Brennan and Resnick's (2012) computational thinking framework to the data collected from second graders who were taught a CT integrated science and engineering module. I found that young students engaged in the following CT practices through the engagement with the CT integrated science and engineering module while planning, building, testing, and proposing a revision to their house models: abstraction and modularization/decomposition, being incremental and iterative, testing, debugging, and data collection. All those CT practices are present in the Brennan and Resnick's CT framework, except for the CT practice of data collection. While this CT practice is not identified as a CT practice in Brennan and Resnick's (2012) framework, its significance became evident in this study. The students' active involvement in collecting and applying information for their projects emphasizes the critical role of data collection within computational thinking. Hence, this discovery broadens the scope of CT practices as outlined by Brennan and Resnick (2012). Another point about the framework is that for such a framework to be useful, we need examples of what such thinking looks like among young children, which the findings of this dissertation contribute to.

Second, the second graders engaged in building a house model, an activity considered an unplugged computational thinking (CT) exercise in this dissertation. This tangible task incorporated the CT practices outlined in Brennan and Resnick’s 2012 framework. The findings from this activity indicate that CT practices are not confined solely to digital or coding environments like Scratch but also apply to more tangible, unplugged activities (Kafai et al., 2014; Hynes et al., 2016; Yang et al., 2018; Yang et al., 2019; Yang et al., 2020). This highlights the adaptability and applicability of computational thinking practices to a range of contexts, including the construction of physical models like the house model.

Third, constructionism emphasizes learning through the process of making. This study posits that the second graders engaged in meaningful learning by creating artifacts such as house models and Scratch programs. This approach aligns with Papert's (1980) concept of 'objects-to-think-with,' where tangible creations embody embedded knowledge and personal identification. The creation of these artifacts in this study, both physical (house models) and digital (Scratch projects), served as external manifestations of the students' computational thinking (CT) development. By actively constructing these artifacts, the students not only internalized CT concepts and practices but also externalized their understanding, a process central to Constructionism as described by Papert (1990). This externalization was not just a demonstration of learned concepts but also a medium for sharing and reflecting, as highlighted by Kafai and Resnick (2011). The integration of making into the learning process, as a foundational design element in this study, effectively facilitated the students' access to CT concepts and practices. Therefore, the use of Constructionism in this context not only reinforces the theory’s relevance in contemporary educational settings but also underscores its effectiveness in engaging young learners with complex computational ideas.

Last, the sequence of activities in this study, beginning with the concrete house model building and subsequently transitioning to the abstract Scratch programming, was essential in enhancing the learning experience for the second graders. Initiating the learning process with a tangible, concrete task like constructing a house model laid a solid foundation for the students, simplifying the initial grasp of fundamental concepts. This approach is in line with the constructionist theory (Papert, 1990), which advocates for learning through active, experiential involvement. Moving from this physical activity to the more abstract and intricate environment of Scratch programming provided an opportunity for the students to expand their initial understanding into a different, digital context. This gradual progression from simple, tactile activities to more complex, conceptual tasks not only eased the learning process but also likely enriched the students' understanding and retention of crucial concepts. Consequently, the intentional design of the activities, transitioning from concrete to abstract, was a key factor in fostering the cognitive development of the second graders in this dissertation. Building on the success of this approach, where a concrete-to-abstract activity sequence significantly bolstered second graders' cognitive development, it's important to consider the broader educational context. Specifically, considering that elementary teachers often do not specialize in Computational Thinking (CT) or Computer Science (CS), integrating CT into subjects like science and engineering, as demonstrated in this study, emerges as a promising strategy to expand CT accessibility among elementary students. In the following sections, opportunities and shortcomings of using CT as an analytic lens will be discussed followed by limitations of the study.

Opportunities and Shortcomings of Using CT as an Analytic Lens

The application of computational thinking (CT) as an analytic lens in this dissertation provides a rich opportunity to understand how second graders interact with and process complex problem-solving tasks. Through this lens, researchers can capture the nuances of students' computational thinking such as problem decomposition, iterative design, and debugging practices, which are crucial elements in the development of CT skills. Moreover, the application of a CT framework to an activity such as house modeling enriches the domain of CT research by demonstrating its relevance beyond computer-based environments. This approach broadens the scope of CT, illustrating its applicability to a variety of problem-solving contexts. Also, the engineering task, which is house modeling, in the curriculum serves as an invaluable avenue for introducing unplugged CT tasks, which can later transition into plugged activities like coding, effectively bridging tangible problem-solving with computational skills development. However, there are shortcomings to consider. Using CT as an analytic lens may inadvertently narrow the focus, potentially overlooking other analytical frameworks out there. It may also impose a computational framework on activities that do not inherently require computational methods, possibly skewing the interpretation of what constitutes learning in that context. By being mindful of these limitations, researchers can more accurately delineate where CT provides clear insights and where it may require supplementation with other analytical approaches.

Challenges and Opportunities in Integrating Computational Thinking within Science Education

Incorporating computational thinking (CT) into science education presents unique opportunities and challenges when compared to other disciplines like English Language Arts or mathematics. Science shares several crosscutting concepts with CT, such as the principle of testing and iteration, which is pivotal in both scientific inquiry and computational problem-solving. In science and engineering, as in CT, products and hypotheses are systematically tested and refined based on outcomes. Educators can harness these overlapping concepts to seamlessly weave CT into the science curriculum, enhancing students' understanding of both areas. However, one of the challenges in this integration is the already crowded science curriculum. Finding the space to include CT without displacing essential science content requires careful curriculum planning and professional development for teachers. Another challenge is ensuring that CT activities are aligned with scientific concepts in a meaningful way that support core principles of both disciplines, which, to some extent, requires the knowledge and skills of both disciplines. Additionally, there is the practical challenge of resources; not all science classrooms have access to the necessary technology to support computational activities.

Limitations of the study

This dissertation has several limitations that need to be acknowledged. First, the participants' artifact creation process (house models and Scratch programs) during lesson implementation was not recorded, although their discussions and screen activities were captured during artifact interviews. Consequently, the complete process of artifact creation may not be fully represented. Second, the study's participants were second graders, whose developing language skills might have limited their ability to thoroughly articulate their thoughts about their artifacts during interviews. Third, the discovery of the CT practice of abstraction and decomposition in the students' comprehension of modeling, as part of RQ1a, was primarily based on field notes. This focus was due to the specific approach of the study, where the concept of modeling was not a central theme in house model interviews nor prominently featured in other datasets for triangulation purposes. This methodological choice represents a limitation, not because any data set was less valuable, but because it reflects the study's selective emphasis on certain aspects of the learning process. Last, the theme of "Abstraction and Decomposition in Planning a House Model" had fewer codes compared to others in the house model data, possibly because the decomposition practice was primarily identified through poster board data and not extensively through house model interview data, indicating a methodological limitation of the study.

Concluding Remarks

This dissertation focuses on the second graders’ computational thinking through their engagement in a CT integrated science and engineering module. This research provided evidence that young students like second graders were able to develop computational thinking concepts and practices in two CT activities: constructing house models and making Scratch programs. Considering the nature of these two activities in the sense that while one is plugged, the other is unplugged, computational thinking is an idea rather than programming or coding in a tool like Scratch. Moreover, it is posited that the lesson progression from tangible house model building to abstract Scratch programming played a crucial role in enhancing the second graders' understanding and development of computational thinking. The insights gathered from house model and Scratch interviews, field notes, photographs of classroom artifacts like poster boards, and Scratch programs offer valuable perspectives on second graders' computational thinking. Although further research is essential to deepen our understanding of computational thinking in young students, both in house model construction and Scratch programming, the findings of this dissertation mark significant progress towards achieving this objective.

APPENDICES

A. Artifact Interview Protocols for Module Three: Natural Disasters

House Model Interview

- **What is a natural disaster? (Doing research on natural disasters)**
    - What natural disasters did you research?
        - What did you learn about that?
            - What causes that natural disaster?
            - What are the consequences of that natural disaster? What damage does it do to the houses or the environment? (e.g., damaging houses)
        - How can we protect ourselves from that natural disaster?
            - Based on the data you collected from books on getepic and videos your teachers showed, how do you build a house that is strong enough to sustain that natural disaster? (e.g., what kind of materials do we use to sustain that natural disaster?)
    - Why did you want to research that natural disaster?
        - Does it have to do with anything in your family or relative or friend?
- **Drawing a design of a house that is strong enough to sustain that natural disaster (Designing/Planning)**
    - Can you show me the design that you drew on Schoology or poster board and tell me how you drew it that way and why?
- **Modeling/Building the design**

    - Can you tell me how you built your house model?
        - What materials did you use?
        - Why did you use those materials?
        - Why did you build your housel model in the way you did? (e.g., building a house whose roof is steep so the roof does not hold rain.)
        - Is your house model connected to your life or the life of someone you know of? If yes, how?
- **Simulating/Testing the design**
    - What did you notice when your teacher tested/simulated your house model?
    - What happened to your house model?
        - Did it pass the test?
            - If yes or no, why?
- **Redesigning the model based on the test**
    - Based on all you learned and did, how would you change your design after you learned more, and why? OR, how would you do your model differently?
        - What material(s) would you change in your model?
        - What part(s) of your model would you change and why?

Scratch Interview

- **Please show me your project?**
    - Can you show me how to start your project?
- **Can you tell me about your project?**
    - Why did you choose these sprites?
    - Why did you choose this backdrop?

- **Can you tell me how you created this script? Or, Can you please explain to me each block in your script? (potential follow-up as below)**
    - How do you make your sprites move? Can you show me?
    - (If a student used recordings) How did you make those recordings? (telling about the changes they would make to their design for natural disasters)
    - Why do you want it to say that?
    - Can you tell me why you put these coding blocks in this order?
    - What if I move the blocks around, (give an example), what would happen?
- **What was hard for you creating this project?**
    - Why?
    - What happened when you got stuck? Follow up: How did you fix it? Or, why do you think you could not fix it?
    - Was there a time when your project didn't run as you wanted?
- **Did you get any help or ideas from anybody, such as your classmates or parents, or even Youtube?**
    - If yes, what kind of help was it?
- **For students who didn't finish their projects:**
    - If you had a chance to complete your project, what would you add or change?
- **For students who finished their projects:**
    - If you have the time, is there anything you would like to add to or change in your Scratch project?

B. A link to Natural Disasters Module (Module Three):

https://www.springfieldpublicschools.com/cms/One.aspx?portalId=494689&pageId=53831879

C. Structured Protocol of Field Notes

Lesson Steps	Teacher Actions	Student Actions	Researcher Notes
This column listed predefined steps of the lesson, such as "introducing models/modeling," which were originally created by the module's authors, whom we observed for the implementation of the module in their classrooms for the first time in 2021 for the purpose of a different research before this dissertation.	In this column, I noted what the teacher was doing during each lesson step, such as asking questions or giving a presentation.	This column captured student actions, like responding to questions from the teacher or peers, or interacting with each other.	Here, I made annotations or interpretations based on the Brennan and Resnick (2012) CT framework, which guided this study.

REFERENCES

Adrion, W. Richards, Katie Bevan, Paul Foster, Denise Matuszczak, Rachel Miller, Laura Rita, Florence R. Sullivan, Sneha Veeragoudar, Scott Wohlers, and Melissa Zeitz. "How a Research-Practice Partnership Refined its Strategy for Integrating CS/CT into K-5 Curricula: An Experience Report." In Proceedings of the 53rd ACM Technical Symposium on Computer Science Education, pp. 592-598. 2022.

Aksit, O., & Wiebe, E. N. (2020). Exploring force and motion concepts in middle grades using computational modeling: A classroom intervention study. *Journal of Science Education and Technology*, *29*(1), 65-82.

Arık, M., & Topçu, M. S. (2022). Computational Thinking Integration into Science Classrooms: Example of Digestive System. *Journal of Science Education and Technology*, *31*(1), 99-115.

Barr, V., & Stephenson, C. (2011). Bring computational thinking to K-12: What is involved and what is the role of the computer science education community? *ACM Transaction on Computational Logic, 2*(1), 48–54.

Brennan, K., & Resnick, M. (2012, April). New frameworks for studying and assessing the development of computational thinking. In *Proceedings of the 2012 annual meeting of the American Educational Research Association, Vancouver, Canada* (Vol. 1, p. 25).

Cabrera, L., Ketelhut, D. J., Mills, K., Killen, H., Coenraad, M., Byrne, V. L., & Plane, J. D. (2023). Designing a framework for teachers' integration of computational thinking into elementary science. *Journal of Research in Science Teaching*.

Cateté, V., Lytle, N., Dong, Y., Boulden, D., Akram, B., Houchins, J., ... & Boyer, K. (2018, October). Infusing computational thinking into middle grade science classrooms: lessons learned. In *Proceedings of the 13th Workshop in Primary and Secondary Computing Education* (pp. 1-6).

Computer Science Teachers Association. (2017). K–12 Computer Science Standards, Revised 2017. Retrieved from https://www.csteachers.org/page/standards.

Denzin, N. K. (1978). The research act: A theoretical introduction to sociological methods (2nd ed.). New York: McGraw-Hill.

Ehsan, H., Rehmat, A. P., & Cardella, M. E. (2021). Computational thinking embedded in engineering design: Capturing computational thinking of children in an informal engineering design activity. *International Journal of Technology and Design Education, 31*(3), 441-464.

Fereday, J., & Muir-Cochrane, E. (2006). Demonstrating rigor using thematic analysis: A hybrid approach of inductive and deductive coding and theme development. *International journal of qualitative methods*, *5*(1), 80-92.

Ginsburg, H. (1997). *Entering the child's mind: The clinical interview in psychological research and practice*. Cambridge University Press.

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

Gautam, A., Bortz, W. E. W., & Tatar, D. (2017, November). Case for integrating computational thinking and science in a low-resource setting. In *Proceedings of the Ninth International Conference on Information and Communication Technologies and Development* (pp. 1-4).

Harel, I., & Papert, S. (1990). Software design as a learning environment. *Interactive learning environments*, *1*(1), 1-32.

HAREL, I. (1993): Who is a Constructionist. International Logo Conference. July 1993, Melbourne, Australia.

Harvard University. (n.d.). Computational Thinking with Scratch. Retrieved from http://scratched.gse.harvard.edu/ct/assessing.html

Harvard University. (n.d.). Student Assessment Rubric. Retrieved from http://scratched.gse.harvard.edu/ct/files/Student_Assessment_Rubric.pdf

Hynes, M. M., Moore, T. J., Cardella, M. E., Tank, K. M., Purzer, S., Menekse, M., & Brophy, S. P. (2016, June). Inspiring Computational Thinking in Young Children's Engineering Design Activities (Fundamental). In *2016 ASEE Annual Conference & Exposition*.

Interviewing students about scratch programming experiences. (n.d.). Retrieved from http://scratched.gse.harvard.edu/ct/files/Student_Interview_Protocol.pdf

International Society for Technology in Education and the Computer Science Teachers Association (2011). Operational definition of computational thinking for k-12 education. Retrieved from https://cdn.iste.org/www-root/ct-documents/computational-thinking-operational-definition-flyer.pdf

Kafai, Y. B., & Resnick, M. (Eds.). (1996). *Constructionism in practice: Designing, thinking, and learning in a digital world*. Routledge.

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), 1-20.

Kalelioglu, F., Gulbahar, Y., & Kukul, V. (2016). A framework for computational thinking based on a systematic research review.

Kramer, J. (2007). Is abstraction the key to computing?. *Communications of the ACM*, *50*(4), 36-42.

Lee, I., Martin, F., & Apone, K. (2014). Integrating computational thinking across the K--8 curriculum. *Acm Inroads*, *5*(4), 64-71.

Litts, B. K., Lewis, W. E., & Mortensen, C. K. (2020). Engaging youth in computational thinking practices through designing place-based mobile games about local issues. *Interactive Learning Environments*, *28*(3), 302-315.

Luo, F., Antonenko, P. D., & Davis, E. C. (2020). Exploring the evolution of two girls' conceptions and practices in computational thinking in science. *Computers & Education*, *146*, 103759.

Maloney, J. H., Peppler, K., Kafai, Y., Resnick, M., & Rusk, N. (2008, March). Programming by choice: urban youth learning programming with scratch. In *Proceedings of the 39th SIGCSE technical symposium on Computer science education* (pp. 367-371).

Maloney, J., Resnick, M., Rusk, N., Silverman, B., & Eastmond, E. (2010). The scratch programming language and environment. *ACM Transactions on Computing Education (TOCE)*, *10*(4), 1-15.

Mannila, L., Dagiene, V., Demo, B., Grgurina, N., Mirolo, C., Rolandsson, L., & Settle, A. (2014, June). Computational thinking in K-9 education. In *Proceedings of the working group reports of the 2014 on innovation & technology in computer science education conference* (pp. 1-29). ACM.

Massachusetts Department of Elementary and Secondary Education. (2017). English Language Arts and Literacy Framework. Retrieved from https://www.doe.mass.edu/frameworks/ela/2017-06.pdf

Massachusetts Department of Elementary and Secondary Education. (2016.). Digital Literacy Computer Science Framework. Retrieved from https://www.doe.mass.edu/frameworks/current.html

Massachusetts Department of Elementary and Secondary Education. (2016.). Science and Technology Engineering Framework. Retrieved from https://www.doe.mass.edu/frameworks/current.html

Merriam, S. B., & Tisdell, E. J. (2015). *Qualitative research: A guide to design and implementation*. John Wiley & Sons.

NGSS Lead States. (2013). Next Generation Science Standards. Washington, DC: The National Academies Press.

Ogegbo, A. A., & Ramnarain, U. (2022). A systematic review of computational thinking in science classrooms. *Studies in Science Education*, *58*(2), 203-230.

Orton, K., Weintrop, D., Beheshti, E., Horn, M., Jona, K., & Wilensky, U. (2016). Bringing computa- tional thinking into high school mathematics and science classrooms. Paper presented at the Trans- forming Learning, Empowering Learners: The International Conference of the Learning Sciences (ICLS) 2016, Singapore.

Papavlasopoulou, S., Giannakos, M. N., & Jaccheri, L. (2019). Exploring children's learning experience in constructionism-based coding activities through design-based research. *Computers in Human Behavior*, *99*, 415-427.

Papert, S. (1980). " Mindstorms" Children. *Computers and powerful ideas*.

Papert, S., & Harel, I. (1991). Situating constructionism. *Constructionism*, *36*(2), 1-11.

Papert, S. (1993). *The children's machine: Rethinking school in the age of the computer*. BasicBooks, 10 East 53rd St., New York, NY 10022-5299.

Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., ... & Kafai, Y. (2009). Scratch: programming for all. *Communications of the ACM*, *52*(11), 60-67.

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.

Shute et al., V. J., Sun, C., & Asbell-Clarke, J. (2017). Demystifying computational thinking. Educational research review, 22, 142-158.

Wang, C., Shen, J., & Chao, J. (2021). Integrating Computational Thinking in STEM Education: A Literature Review. *International Journal of Science and Mathematics Education*, 1-24.

Waterman, K. P., Goldsmith, L., & Pasquale, M. (2020). Integrating computational thinking into elementary science curriculum: An examination of activities that support students' computational thinking in the service of disciplinary learning. *Journal of Science Education and Technology*, *29*(1), 53-64.

Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., & Wilensky, U. (2014). Defining computational thinking for science, technology, engineering, and math. Paper presented at the 2014 Annual Meeting of the American Educational Research Association, Philadelphia, Pennsylvania. https:// ccl. north weste rn. edu/ papers/ 2014/ CT- STEM_ AERA_ 2014. pdf.

Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., & Wilensky, U. (2016). Defining computational thinking for mathematics and science classrooms. *Journal of science education and technology*, *25*(1), 127-147.

Wilensky, U., Brady, C. E., & Horn, M. S. (2014). Fostering computational literacy in science classrooms. *Communications of the ACM*, *57*(8), 24-28.

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

Wing, J. M. (2008). Computational thinking and thinking about computing. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, *366*(1881), 3717-3725.

Yadav, A., Good, J., Voogt, J., & Fisser, P. (2017). Computational thinking as an emerging competence domain. In *Competence-based vocational and professional education*(pp. 1051-1067). Springer, Cham.

Yang, D., Swanson, S. R., Chittoori, B., & Baek, Y. (2018). Work in progress: Integrating computational thinking in STEM education through a project-based learning approach.

Yang, D., Baek, Y., Chittoori, B., & Stewart, W. H. (2019). Elementary students' computational thinking practice in a bridge design and building challenge (Fundamental).

Yang, D., Baek, Y., & Swanson, S. (2020, October). Developing computational thinking through project-based airplane design activities. In *2020 IEEE Frontiers in Education Conference (FIE)* (pp. 1-4). IEEE.