The Impact of an Integrated Literacy and Design Activity on Student Attitudes Toward Coding

Scott R. Bartholomew, Brigham Young University
Emerson Barnum, Brigham Young University
Jessica Allen, Brigham Young University
Kellie Wilcock, Brigham Young University

ABSTRACT

Coding is a growing and important area within Design and Technology Education and is also one of the arenas of education where the most significant effort is being given to increase diversity, equity, and inclusion. To introduce young learners to coding and engineering design, a pioneering curricular unit was designed for upper elementary schools, intertwining literacy within its framework. To reasonably fit in the already overcrowded standards for elementary schools in the United States, the integration of multiple subjects was a defining feature of this unit which we termed "Digital Storyboards." Digital Storyboards integrate engineering design, literacy, and coding into one unit which emphasizes students' ability to design, develop, and automate an illustration from a favorite story using a variety of electronic elements including LEDs, copper tape, and micro:bits. Students are intentionally taught core content from literacy (the elements of a story), engineering (design), and computational thinking (variables, loops, Booleans) while they create and program their own digital storyboards as part of a 10-week unit in class. While initial implementations of digital storyboards in one classroom positively impacted all students, a more significant impact was discovered with female students specifically – an important idea since females are traditionally underrepresented in coding. Following our pilot work, the digital storyboard project was expanded into 16 classrooms with more than 200 students. Our findings, as well as the practical implications for teachers engaged with elementary and secondary content related to literacy, engineering, design, and computer science, will be shared.

Key Words: Elementary Education, Coding Integration, Computer Science Education, Literacy Integration, Electrical Engineering Education

1. INTRODUCTION

In the predicted future, individuals who are deemed literate within society will require, at minimum, a foundational comprehension of programming skills. Undoubtedly, the need for this essential skill will significantly increase, particularly in the context of childhood education (Murphy, 2022). Prensky (2008), argues that as "programming becomes more important, it will leave the back room and become a key skill and attribute of our top intellectual and social classes, just as reading and writing did in the past".

As programming literacy increasingly emerges as a prerequisite for surviving the digital world, the imperative to develop programming skills in the current generation of elementary children becomes ever more relevant (Murphy, 2022). However, the integration of coding basics into schools in the US is a daunting task, particularly when considering the extensive teacher training required and the demanding curricular standards upheld by American educational institutions (Yadav, Gretter, Hambrysch, & Sands, 2016).

Extensive research has illuminated the reciprocal relationship between computer science and literacy, underscoring the advantages of integrating computer science lessons with literacy instruction (Jacob & Warschauer, 2018). Programming has the potential to support literacy skills by encompassing writing, reading, brainstorming, and much more (Murphy, 2022). By occasionally merging these territories, elementary schools may be able to effectively address the vital incorporation of computer science while also tackling the demanding requirements of educational standards and schedules.

1.1. Digital Storyboards

The Digital Storytelling Project (DSP) was devised to integrate coding skills, literacy, and engineering design. To facilitate implementation within the busy schedules of the elementary school teachers, undergraduate Design and Technology Education students were sent to team-teach with elementary school teachers in several classrooms. These visits occurred once a week for a duration of ten weeks. Research data were gathered through the administration of pre- and post-program surveys, as well as interviews conducted with both students and teachers following the conclusion of the project.

Throughout the course of the 10-week program, the students were guided in the exploration of fundamental concepts in literature, which were then enriched through the integration of coding and engineering principles. In the initial weeks, the students received instruction on the essentials of storytelling, practicing familiar vocabulary and retelling narratives as a preliminary review before delving into more advanced lessons. Subsequent weeks were dedicated to lessons on circuitry, computational thinking, and the main aspects of storytelling. This paved the way for the introduction of a Digital Storyboard, a project that aimed to merge engineering design, literacy, and coding. This activity offered a vast range of creative possibilities, allowing each student to select a board that features a scene from a favorite TV show or movie that they then personalized through coloring, coding, and programming of LED lights. The circuitry component proved challenging for many students, yet it emerged as a highlight of the DSP according to subsequent interviews. In the final weeks of the project, the students were further introduced to the capabilities of micro:bits, engaging in coding challenges and projects designed to encourage creativity, brainstorming, and problem-solving abilities.

The specific research question guiding our investigation was: What is the impact, if any, on student perceptions of coding, following participation in the Digital Storyboard Project?

2. LITERATURE REVIEW

The Digital Storytelling Project follows both the Utah (a rocky mountain state in the United States of America) state *Science with Engineering Education* (SEED) and *English Language Arts* (ELA) standards. While literacy, encompassing the skills of reading, writing, verbal expression, and critical thinking, has long been recognized as a fundamental pillar in the educational curriculum (Billman & Pearson, 2013), we noted that the instruction of engineering design and computational thinking can be effectively intertwined. Specifically, the context of engineering design presented a fitting opportunity for children to actively engage in computational thinking while they simultaneously engaged in this literacy learning experience (Ehsan, Rehmat, & Cardella, 2021).

2.1. Literacy

The early stages of childhood play a crucial role in equipping young learners with the essential skills for adult literacy (Hopkins, Brookes, & Green, 2013). Research shows that during this foundational phase, it is imperative to pose significant questions and emphasize literacy competencies such as phonological awareness, phonics, fluency, and reading comprehension (National Governors Association Center for Best Practices & Council of Chief State School Officers, 2010). Often, these literacy competencies can be taught through technological tools, supports, and experiences (Hopkins, Brookes, & Green, 2013).

The initial lessons of the DSP focused on the fundamental elements of storytelling embedded in the literacy standards; these included characters (CCSS.ELA-LITERACY.RI.4.6), plots (CCSS.ELA-LITERACY.RI.4.5), themes (CCSS.ELA-LITERACY.RI.4.2), and settings (CCSS.ELA-LITERACY.RI.4.3). Over the course of the ten-week program, students were encouraged to integrate their newfound skills with their storytelling abilities. The digital storyboards provided an opportunity for students to bring these stories "to life." Drawing upon their literary knowledge, they worked to build up their narratives to the climax of the story, where they then unveiled their boards to their classmates and used these boards as a means of telling their chosen story in a new and exciting way. At this pivotal moment, students illuminated their boards. Some incorporated personalized micro:bit codes to enhance the uniqueness of their storyboards. For instance, they programmed flashing lights to act in place of laser shots from their Star Wars ships, adding an extra layer of creativity. While literacy education represents an immensely large area of research, additional focus in not devoted to the particular elements used as engineering and computational thinking were the topics of main interest.

2.2. Engineering Design

Engineering is seen as beneficial for both student development and success along with workforce readiness; thus, leading to its integration across K-12 schools (Arik & Topcu, 2020). This has resulted in the development of numerous endeavors, research projects, and educational lessons and materials employing the "Engineering Design-Based Learning" method (Arik & Topcu, 2020). The incorporation of the engineering design process, encompassing the stages of asking, imagining, planning, creating, and improving (Syukri, Halim, Mohtar, & Soewarno, 2018), was emphasized throughout the lessons to equip the students with the proper tools to navigate the challenges associated with the DSP, particularly the micro:bit challenges. Research has shown engineering design can provide students with an opportunity to fail often and then succeed more quickly; thus, encouraging teachers to make the value of failure for learning and improving designs more evident in their lessons (Cunningham & Lachapelle, 2014). Engineering design activities can also encourage students to both reflect on their designs and ask new and better questions for improvement; therefore, encouraging students to embrace failure if they learn from those experiences and enhance their designs, learning, and ideas (Cunningham & Lachapelle, 2014). We intentionally combined literacy and computational thinking *through* an engineering design activity to leverage these opportunities for students.

2.3. Computational Thinking

Computational thinking describes the several mental procedures involved in framing problems and devising solutions in a way that these challenges can be efficiently executed with the assistance of technology (Wing, 2011). The key components of computational thinking include decomposition, pattern recognition, abstraction, algorithmic design, and evaluation. While computational thinking is widely practiced in the world of computer programming, it is an important component of computer science education that is gradually integrating into the K-12 curricula to promote problem-solving skills and critical thinking among students (Wing, 2011). Additionally, the state of Utah (located in the United States) recently passed legislation mandating all elementary school teachers include computational thinking in their curriculum – another key element in our desire to include computational thinking in the Digital Storyboard Activity.

Our past experiences demonstrated some frustration and struggles as students were introduced to computational thinking principles. These struggles did not end with students and elementary school teachers were also often hesitant to embrace these ideas which were often new and foreign to them. Therefore, as part of the DSP, we relied on literature around character introduction. Specifically, Wand, Lee, & Chu (2010) who demonstrated that the probability of children grasping challenging concepts increases through the utilization of sensory stimuli such as vivid characters; thus, establishing connections that can unlock the potential of the developing brain. To enhance the comprehension of computational thinking, our strategic approach involved the creation of four distinct characters, each associated with a specific concept. These "computational thinking friends" (see Figure 1) were introduced across participating classrooms as part of the Digital Storyboards Project (DSP). While computational thinking is an umbrella term that has been used to refer to a variety of fundamental concepts and reasoning methods that originated in computer science, these students were introduced to decomposition, pattern recognition, abstraction, and algorithmic thinking in the context of four distince characters with personality traits mirroring the computational thinking principle they were named for.

Deco the Zombie served as a guide, emphasizing the significance of decomposition - the skill of breaking down complex problems into manageable steps. Pat the Cat highlighted the importance of recognizing patterns, emphasizing the value of examining past problems and solutions to inform current challenges. Abs the Detective encouraged students to think creatively, urging them to visualize abstract ideas and explore unconventional solutions. Lastly, Algordo the Chef guided students in the art of following and generating precise instructions, particularly when engaging with coding exercises involving Micro:bits.

Figure 1.
Computational Thinking Friends: Deco, Pat, Abs, & Al Gordo

By merging these characters with the core lessons of literacy, engineering design, and computational thinking, students were encouraged to integrate their knowledge in both literacy and technology, enabling them to automate sections of their storyboards. This process provided students with the opportunity to witness the physical completion of their digital storyboards.

3. METHODS

To gauge the impact, if any, of the DSP on student perceptions of coding, a mixed-methods approach was utilized which encompassed a pre/post survey as well as semi-structured interviews completed at the conclusion of the DSP. This approach was utilized to explore both the *what* (quantitative findings from the ESCAS; pre/post) and the *why* (semi-structured interviews) of the students DSP experience. Specifically, quantitative data collection came as students completed the Elementary Student Coding Attitudes Survey (ESCAS; Mason & Rich, 2020), a 23-item instrument that assesses elementary students' coding attitudes and self-efficacy before and after engaging in the DSP activities. All items are measured using a six-point Likert scale, where selecting a 'one' represents strong agreement with the statement and selecting a 'six' represents strong disagreement. The ESCAS was specifically designed to assist educators, administrators, and researchers in their attempts to better understand which factors influence students' attitudes toward coding and confirmatory factor analysis using data from 6000 4-6 grade students identified five strong factors: coding confidence, coding interest, social value, perceptions of coders, and coding utility (Mason & Rich, 2020).

Qualitative data collection came through semi-structured interviews conducted at the conclusion of the project. Following the DSP, five students from each class were interviewed by a member of the research team. These five students were selected by the teachers of the class. Teachers were asked to choose two top, two low, and one middle-performing students for the interviews – without letting the research team member know which category each student was aligned with. This quota sampling was used to ensure that a variety of student experiences were included. Interviews were conducted using semi-structured interviewing techniques from Berg (2009) and averaged between 5-10 minutes in length. Each interview was initially recorded and then later transcribed (interview questions are included in the Appendix).

Following the transcription of the interviews, all student responses were collated for further analysis by members of the research team. The first step in this process was for each member of the research team to read over the students' responses and develop initial thoughts and ideas regarding themes. A discussion was held following this exercise and several potential categories were established. Following this process, each member of the research team independently coded several student interviews using the proposed themes and, following this coding, a comparison of results and discussion was undertaken. This process was repeated three times with a refinement/replacement of codes until a final coding scheme was developed. This process served as an inter-rater reliability check as differences in coding were discussed at each stage until agreement was reached among members of the research team. Further, the addition, removal, and refinement of thematic code wording at each step also assisted in improving the reliability between members through discussion and improving understanding for each coder.

When a final coding scheme was developed, with no discrepancies in coding theme assignment between research team members, each student response was independently coded by multiple members of the research team using the final themes (see Table 1).

Table 1.
Themes for the qualitative thematic coding of student interview responses

Theme	
1	Grit
2	Reference to computational thinking characters
3	Coding Inputs and Outputs
4	Following Explicit instructions vs Problem Solving
5	Physical components versus digital components
6	Specific reference to a task/challenge
7	Teamwork
8	Choice/Freedom
9	Complexity of task/directions
10	Mentor/Adult influences

The resulting counts–both total and by thematic code–were later used in conjunction with the quantitative findings to both triangulate and explore the findings from the DSP experience for students.

4. FINDINGS

Two-tailed paired t-tests were used to determine if any statistically significant shifts were detected between the pre-tests and post-tests in each of the five categories measured by the ESCAS: Confidence, Interest, Utility, Social Influence, and Perception of Coders. The mean shift was found to be significantly positive in all categories with a p-value below .01 (See Figure 2 and 3). Figure 2 presents the mean numerical equivalent of all students' responses combined. For example, the mean pre-test score for most categories was between 2 and 3 meaning that students generally disagreed with the provided statements. The mean post-test scores for most categories were between 4 and 5 meaning that students generally agreed with the provided statements.

Figure 2.
Pre to Post Change in Students' Attitude for Coding (Likert scale 1-6 questionnaire)

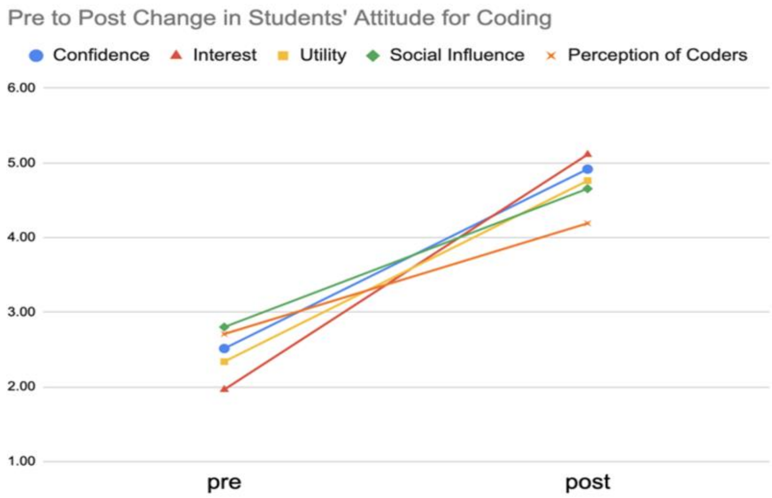
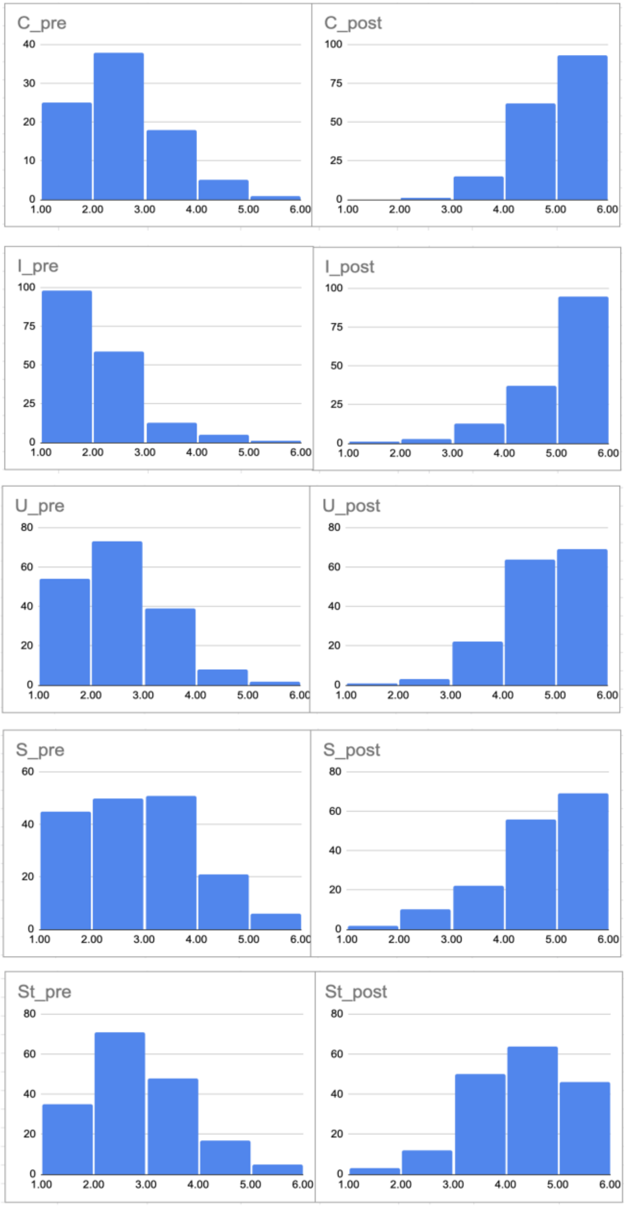


Figure 3 presents a histogram of each students' average response to each category on both the pre and post-tests. For example, in the category of confidence the most common average response was between 2 and 3. However, following students' participation in the DSP the most common average response in the same category was between 5 and 6. The right shifts of the histograms in each category show a shift towards greater agreement with the statements in the ESCAS (e.g., positive perceptions of and attitudes towards coding).

Figure 3.

Pre to Post Change in Student Attitude for Coding by Subarea



The independent coding of student interview responses by multiple members of the research team resulted in a total of 249 unique student statements being coded into one of the identified categories. The total counts, as well as the proportion for each theme, are included in Table 2.

Table 2.
Counts for thematic coding of student interview responses

	Theme	Count	Percentage
1	Grit	26	10%
2	Reference to computational thinking characters	22	9%
3	Coding Inputs and Outputs	2	1%
4	Following Explicit instructions vs Problem Solving	26	10%
5	Physical components versus digital components	26	10%
6	Specific reference to a task/challenge	92	37%
7	Teamwork	21	8%
8	Choice/Freedom	15	6%
9	Complexity of task/directions	7	3%
10	Mentor/Adult influences	12	5%

4.1. Student Interview Themes

While student responses to interview questions were spread relatively evenly across the identified thematic categories, it was noteworthy to the researchers that the most coded area was student comments about a specific task or challenge. Students often mentioned a challenge that was especially fun/exciting or difficult/frustrating. For example, one student referenced the traffic light challenge by saying:

> *Um I it was kind of hard because the um like it was kind of frustrating too because it like didn't turn on, (*) you get a little mad because like you've been working on it or maybe it like would turn on and off and on and off.*

We also noted many instances of student comments that demonstrated an ability to do hard things, to push through a challenge, or to work harder than they had on other tasks - all areas we labeled as "grit." This idea of grit was not something we had originally hypothesized as an area arising from the study, but several student comments highlighted this as a benefit of student participation; for example, students commented:

> "*It was so hard. And then we're like, this is too hard. I can't do it. And then we keep trying and trying and trying. And and then it's like, oh, it's finally working."*

> *"I liked how it was challenging because I had like no idea what I was doing, but it was fun when I was done because I thought I could maybe do that again."*

> *"·It was difficult. But later the as soon as I got towards the end it came together to me really easy. So now I can solve coding projects."*

"We got all the lights to turn on. I thought it was impossible. Because it was super hard for me but I got over it. That's what was most exciting."

Several students mentioned challenges, and successes, from both solving written instructions (e.g., those included in the tasks) and using open-ended problem-solving skills. Although we did not originally delineate between these types of problems, it was interesting to note student comments that alluded to these ideas, including:

"So I learned about thinking for like how to do it in ways of like if mine didn't work"

"Yeah, I learned that like ... if something doesn't work, you can rethink and um rethink what happened and see and change that. See if that works. ..."

"So the paper is telling me to do this but I have to like, think how am I gonna do it?"

Finally, we noted several instances where students specifically referenced either the physical (e.g., LEDs, wires, breadboards) or digital components (e.g., code, CT characters). This was an interesting delineation as we did not intentionally separate ideas, concepts, or challenges into these categories. Student comments included:

I really like the story card, because like, there was a little light that we had to figure out which side was negative and positive. And then also with their like, copper tape and stuff like that, and the drawing and the light and stuff, and the battery. And also the story. What I liked that a lot, because then we got to like, weave the lights through it and then hook a battery up to it.

"...[the challenge we were working on] was kind of hard, because we had to like, hook it up to the computer and then we had to change it and then download it and change it and change it and take it and then download it again. And then also like putting the battery in and then the battery pack into the microchip to make it make it sound."

5. CONCLUSION

Arguments have been made in favor of implementing computational thinking and coding into STEM curricula within K-12 (Sengupta, Kinnebrew, Basu, Biswas, & Clark, 2013). Given this perspective, it is not unreasonable to anticipate a steady incorporation of computer science concepts into future programs, areas, and educational institutions. In many instances, the teachers tasked with incorporating these ideas are not prepared to do so – whether they be elementary school teachers or middle/high school Design & Technology teachers. By starting to prepare teachers for these imminent changes, we can ensure a more smooth and successful transition toward a curriculum that welcomes and prepares students for the demands of the near future.

The Digital Storyboards Project employs a distinctive method aimed at introducing students to engineering design and coding by intertwining it with literacy components – something that we anticipate being attractive and reasonable for currently-practicing Design & Technology teachers given their background and the implicit connections to other commonly-taught content. Further, the connections between the DSP and literacy principles should make it both attractive and feasible for elementary school teachers to incorporate. Preliminary data analysis indicates that the DSP has demonstrated promising results, leading to a favorable shift in the majority of students' perspectives toward both computer science and coders. The successful utilization of our unique approach of combining elements from several silos (literacy, engineering design, coding) into a single project also lends credence to the potential for additional activities which synthesize several areas, ideas, and courses.

However, to ensure success throughout K-12 schools, it is important to explore potential avenues for improvement that does not involve the additional support of undergraduate students. Ongoing research is being conducted, including the expansion of more 4th-grade classrooms and teachers. The hope of this future research is to enhance the DSP's effectiveness and sustainability, thereby proving its value as an innovative, sustainable, and positive educational initiative. Findings from these ongoing efforts can be used to continually shape and improve the usefulness of the approach as well as inform future efforts centered on combining various subjects/topics/fields into a cohesive project for students.

An aspiration for the future of this program is to integrate the Digital Storytelling Project (DSP) into classrooms – both statewide and nationwide. Specifically, our next efforts are focused on eliminating the dependency on undergraduate student teachers. This expansion would enable greater exposure to computer science, engineering design, and literacy for elementary-aged students. Additionally, our review of student semi-structured interview comments and survey results is inspiring adjustments to further improve the project efficacy.

6. REFERENCES

Arık, M., & Topçu, M. S. (2020). Implementation of engineering design process in the K-12 science classrooms: Trends and issues. *Research in Science Education*, 1-23.

Billman, A., & Pearson, P. D. (2013). Literacy in the disciplines. *Literacy Learning: The Middle Years*, *21*(1), 25-33.

Cunningham, C. M., & Lachapelle, C. P. (2014). Designing engineering experiences to engage all students. *Engineering in pre-college settings: Synthesizing research, policy, and practices*, *21*(7), 117-142.

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.

Hopkins, L., Brookes, F., & Green, J. (2013). Books, bytes and brains: The implications of new knowledge for children's early literacy learning. *Australasian Journal of early childhood*, *38*(1), 23-28.

Jacob, S. R., & Warschauer, M. (2018). Computational thinking and literacy. *Journal of Computer Science Integration*, *1*(1).

Murphy, K. (2022). Computer Science and Literacy in K-5 Education: A Reciprocal Relationship.

Prensky, M. (2008). Programming is the new literacy. Edutopia magazine.

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*, 351-380.

Syukri, M., Halim, L., Mohtar, L. E., & Soewarno, S. (2018). The impact of engineering design process in teaching and learning to enhance students' science problem-solving skills. *Jurnal Pendidikan IPA Indonesia*, *7*(1), 66-75.

Wang, W. C., Lee, C. C., & Chu, Y. C. (2010). A brief review on developing creative thinking in young children by mind mapping. *International Business Research*, *3*(3), 233.

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

Wing, J. (2011). Research notebook: Computational thinking—What and why. *The link magazine*, *6*, 20-23.

Yadav, A., Gretter, S., Hambrusch, S., & Sands, P. (2016). Expanding computer science education in schools: understanding teacher experiences and challenges. *Computer science education*, *26*(4), 235-254.

7. APPENDIX

Tell me about your experience with the Digital Storytelling project

(i) What did you like, dislike, etc.?
(ii) What was hard, easy, fun, exciting, challenging?
(iii) What did you learn about Computational Thinking while working on this project? Did your experience surprise you? Can you see yourself using this information again, how?
(iv) Can you identify any instances of problem decomposition from the activity?
(v) Can you identify any instances of pattern recognition from the activity?
(vi) Can you identify any instances of abstraction from the activity?
(vii) Can you identify any instances of algorithm design from the activity?
(viii) Would you consider a career in Computational Thinking after an experience like this?
(ix) Anything else you'd like to share with me from this experience with the Digital Storytelling Project?