Journal of Museum Education

ISSN: 1059-8650 (Print) 2051-6169 (Online) Journal homepage: https://www.tandfonline.com/loi/rjme20

Building Capacity for Computational Thinking in Youth through Informal Education

Leigh Ann Mesiti, Alana Parkes, Sunewan C. Paneto & Clara Cahill

To cite this article: Leigh Ann Mesiti, Alana Parkes, Sunewan C. Paneto & Clara Cahill (2019) Building Capacity for Computational Thinking in Youth through Informal Education, Journal of Museum Education, 44:1, 108-121, DOI: 10.1080/10598650.2018.1558656

To link to this article: https://doi.org/10.1080/10598650.2018.1558656

Published online: 08 Feb 2019.

JOURNAL OF MUSEUM EDUCATION
2019, VOL. 44, NO. 1, 108–121
https://doi.org/10.1080/10598650.2018.1558656

Routledge
Taylor & Francis Group

WHAT THE RESEARCH SAYS

Check for updates

Building Capacity for Computational Thinking in Youth through Informal Education

Leigh Ann Mesiti, Alana Parkes, Sunewan C. Paneto and Clara Cahill

ABSTRACT
Building Computational Thinkers, a three-year research study, explored how educators and designers can most effectively support the development of computational thinking capacity, and how these learning experiences could be customized to meet the needs of different learners. This research study focused on three specific exhibit design approaches that conveyed problem decomposition content in *The Science Behind Pixar (Pixar)*, a 13,000 square foot traveling exhibition about the computer science, mathematics, and science behind Pixar's innovative films. Phase One investigated how novice learners could be supported to interact with exhibits and understand problem solving strategies that tackle complex, creative challenges in computer programming. Phase Two investigated the affordances of these exhibits to build capacity, feelings of efficacy, and interest in problem decomposition content in middle and high school youth. The findings in this paper describe the types of scaffolds that can be used to support computational thinking in novice youth, as well as how a combination of exhibit approaches were found to increase youth perceptions, understanding, and beliefs of computer programming. It will also discuss how two exhibit approaches worked particularly well for engaging girls in problem decomposition content.

ARTICLE HISTORY
Received 29 November 2018
Revised 10 December 2018
Accepted 10 December 2018

KEYWORDS
Computational thinking; problem decomposition; exhibit design; exhibition; informal science education; novice and expert; girls; Pixar

About the study and theoretical framing

As computing continues to play a larger role in daily life, there is a sense of urgency for formal and informal learning environments to introduce youth to computational thinking before they reach higher education. The question is, how can we support students to systematically engage in the process of solving complex problems? Jeannette Wing, a leading thinker in computer science, defines the construct of computational thinking as the "thought processes involved in formulating problems and their solutions so that the solutions are represented in a form that can be effectively carried out by an information-processing agent."[1] Computational thinking is becoming a new literacy for the twenty-first century as it encompasses problem-solving skills that can transfer to other domains.[2] Informal learning environments, such as museums, can support a variety of visitors by providing opportunities to develop computational thinking capacity and providing an introduction to a contextualized use of computational thinking that complements the work of formal K-12 settings.

Building Computational Thinkers through Informal Exhibit Experiences (Building Computational Thinkers)[3] was a three-year research study that began in 2013 to learn how to effectively support computational thinking practices, and increase visitors' capacities in these skills given the varying needs of visitors with novice to expert backgrounds. This study, conducted by the Museum of Science, Boston, explored how novices can be supported to engage in computational thinking practices during informal education experiences. It focused on exhibits from *The Science Behind Pixar (Pixar)* traveling exhibition that presented computer science, math, and science skills that go into creating animated films at Pixar.[4] The goals of this study were to (1) Develop exhibits that convey computational thinking content; (2) Explore and understand differences in how experts and novices think about exhibits conveying computational thinking; (3) Create scaffolds that mediate novice use and understanding of computational thinking tools and approaches in exhibits; and (4) Understand impacts of different exhibit design strategies for developing computational thinking capacity in youth related to aspects of learner identity, including self-efficacy and prior interest. This article will describe the exhibit approaches we used to convey computational thinking content, discuss the strategies that novices and experts used to successfully engage with these exhibits, unpack the strategies embedded into the exhibits to better support learning, and identify the impacts of these exhibits on novice learner.

Computational thinkers are skilled at figuring out how to break problems into discrete steps that can be communicated to and carried out by a computer. For the purposes of our research, we considered computational thinking to include three dimensions: concepts (i.e. algorithms, patterns, abstraction, and decomposition), practices (i.e. debugging, creating, collaborating), and perspectives (i.e. sense of identity, empowered to ask questions about technology, relationship to the technical world).[5] The exhibits in this study (and to a lesser extent the *Pixar* exhibition as a whole) focused on the concept of problem decomposition which we define as breaking down a problem into its functional elements in order to better understand the problem and craft an appropriate solution.

Part of this study focused on how to structure teaching and learning about problem decomposition by leveraging similarities and differences in how novices and experts approach computational problem solving in order to create scaffolds or embedded supports that would help novices to think more like experts.[6] Understanding a task requires knowledge about the nature, demands, and strategies of addressing the task.[7] Expertise consists of a web of skills and knowledge in a logical order.[8] Research has shown that it takes years of learning and practice to become an expert, and expertise is a continuum that differs by interest and subject domain.[9] Experts organize their knowledge of a subject around big ideas or overarching themes, and their knowledge often reflects contexts of applicability. Because of this, experts are skilled at selectively and appropriately applying facts and skills to a variety of situations or contexts.[10] This includes recognizing meaningful patterns and identifying relevant information, in addition to working faster and more accurately. Experts are often able to think metacognitively about their work in order to integrate new knowledge and reflect on how to make improvements to their own understanding or strategies.[11]

When confronted with an unfamiliar task or problem, novices tend to struggle with sense-making, process management, articulation of ideas, and reflection.[12] Novices may have some basic skills and knowledge, but often have not had opportunities to practice using them in an applicable setting.[13] They may struggle to organize prior knowledge in order to identify relevant information and may focus on superficial aspects of a problem because seeing the underlying task structure is difficult. They may employ naïve strategies, as their thinking is problem-specific and tied to knowing a formula or limited steps to use.[14] Often their learning is outcome oriented, focusing on the end product rather than the process.[15] As such, this study aimed to see where the greatest gains could be made for novices by identifying approaches that experts are using and novices are not. Through scaffolding, the exhibits aimed to assist novices in doing a task that might be otherwise difficult for them to do, while helping them draw from their own experiences to improve their process, skill, understanding, and self-efficacy.[16]

Exhibit design approaches

The structure or design of a software tool or exhibit can help support people's interactions with a task and mediate what they are able to accomplish by supporting intuitive strategies and providing the learner with levels of responsibility and control. In the *Pixar* exhibition, these exhibits used activities to structure a learning goal and guide visitors through key aspects while supporting their planning and performance.[17] These exhibits helped distribute cognition, where they could assist with the types of computation necessary to implement information in a usable form, such as restricting the problem space so visitors can focus on relevant goals and resources in more productive and targeted ways.[18]

This study focused on three specific exhibit design approaches to convey problem decomposition content in the *Pixar* exhibition, and each strategy facilitated different types of multimodal cognitive and behavioral engagement with computational thinking (Figure 1). The aim was to create exhibits with a "low floor, high ceiling." It should be easy for a novice to get started, but the tool should be powerful and extensive enough to be recognized as authentic to advanced programmers and allow for more in-depth exploration.[19] These exhibit approaches are built on common educational strategies so

Figure 1. Examples of three approaches to conveying computational thinking content (left to right): "Pixar's Modeling Challenge," a multimedia narrative (video); "Lighting Effects Basics," a solution exploration; and "Crowd Simulation Workstation," a creative design activity. Beth Malandain © Museum of Science, Boston.

they could be adapted to other informal and formal learning environments. We developed six exhibits to these design strategies (two of each), variations of which were also present throughout the exhibition.

- ***Multimedia narratives*** used storytelling and direct instruction approaches to describe specific examples of *Pixar* filmmakers using problem decomposition to solve specific creative challenges. Visitors heard from an expert about how a complex problem was broken down into functional parts that could be processed by a computer. Essential elements of these videos included an authentic story, personal connections, and a cohesive explanation of the problem and solution. These exhibits were "Pixar's Sets Challenge" where visitors learned how math is used to construct a natural landscape on a computer and "Pixar's Modeling Challenge" where visitors learned how animators address the challenge of making smooth surfaces for complex shapes.
- ***Solution explorations*** enabled visitors to experience the ways *Pixar* filmmakers decompose a creative challenge into discrete, adjustable, and computable functions. Visitors toggled functions on or off within the overall algorithm and adjusted function parameters to learn how an experts' solution to a complex problem entailed breaking it down or building it up from its functional elements. These exhibits provided guided exploration to the user and had well-defined tasks to accomplish. These exhibits were "Lighting Effects Basics" where visitors matched the lighting on three virtual spheres to a physical counterpart and "Surface Appearance Workstation" (Figure 2) where visitors adjusted the visual appearance of virtual sphere using different surface textures and effects to match one provided by the exhibit.

Figure 2. In "Surface Appearance Workstation," visitors adjusted appearance variables including color, texture, transparency and reflection to make their sphere match one provided. Michael Malyszko.

- ***Creative design activities*** were constructionist-based, interactive experiences that engaged visitors in adjusting, evaluating, and refining the logical flow of an algorithm. Visitors were able to design a creative solution to a complex problem by assembling an algorithm from functional parts. These exhibits have greater emphasis on exploration and the development of user-defined goals. These exhibits were "Programming Natural Variety," (Figure 3) where visitors manipulated variable parameters to create a field of grass and "Crowd Simulation Workstation" (Figure 4) where visitors created schooling behavior of a group of fish through adjusting three slider variables.

Methods

Project research was conducted in two phases. Phase One focused on characterizing and understanding behavioral, social, and cognitive strategies that experts and novices used to learn and approach complex problems, so that design scaffolds could reflect expert strategies and better support novice learning and engagement. Novices were middle and high school-aged youth (grades 6–12) and their visiting groups ($N = 13$ novice groups comprising 29 individuals). Experts were computer science professionals, graduate students, or educators with experiences in a computer animation or programming field ($N = 14$ expert groups comprising 31 individuals).[20] Upon arrival, novice and expert participants completed an interview and survey about their prior experience, awareness, attitudes,

Figure 3. In "Programming Natural Variety," visitors designed a field of grass by adjusting six attributes including number of blades, color, length, curvature, and how variable or uniform those characteristics were. Nicolaus Czarnecki.

Figure 4. In "Crowd Simulation Workstation," the problem of designing a school of fish is broken down into three parameters visitors can modify: distance between fish, number of groups, and match the direction of other fish. Nicolaus Czarnecki.

motivations, and interests toward computing. Groups then used two to four exhibit prototype versions of solution explorations and creative design activities. After each exhibit, the researcher interviewed participants to better understand their overall experience. Participants took part in retroactive think-alouds, where a video of their own exhibit interaction was played back to them so they could discuss thinking and sense-making strategies used at the exhibit. The protocol ended with a demographic survey.[21] During analysis, researchers identified the similarities and differences between expert and novice approaches to problem solving and learning.[22]

Phase Two focused on how museums can afford novice youth the opportunity to develop computational thinking capacity and become initiated into contextualized use of computational thinking. As with Phase One, youth were middle and high school-aged youth (grades 6–12) and their visiting groups. Prior to visiting, youth completed a survey about their previous experience and self-efficacy related to computer programming, a written exercise about computer programming, and demographic information. On arrival, participants completed a problem decomposition task to list believable characteristics from a short *Pixar* video clip (ex. floating balloons from *Up*), and to pick one feature and describe what steps they imagined might be used to help create or improve that feature. Each group of participants was then assigned to interact with a two to four nearly final versions of Phase One exhibits in specific combinations of design strategies.[23]

Participants ($N = 224$ individuals) were observed and timed by researchers and completed a reflection survey after each exhibit interaction. After using all assigned exhibits, youth shared what about what they had learned was most surprising or interesting and completed a post-survey about their interest, self-efficacy, and demographic characteristics, followed by the problem decomposition task they had taken prior. Lastly, a follow-up survey was sent one to four months after their visit that asked what exhibits they remembered using and found most interesting, the types of programming variables they remembered, if they discussed and/or thought about what they learned afterwards, and their current perceptions of computer programming and self-efficacy. During analysis, researchers identified similarities and differences between groups based on demographics and prior experience with computer programming.[24]

Phase One: designing exhibits that encourage computational thinking strategies

In Phase One, researchers analyzed novice and expert observations and interview responses to understand strategies used by each sample group. Strategies fell into themes of identity, programming and computational thinking, interactions with activity controls and variables, and exhibit critique. Specific design changes in exhibits conveying computational thinking strategies were developed based on a combination of these themes. At times, novices and experts showed similar behaviors, but the nuances and sophistication of responses between novices and experts helped characterize specific scaffolds for exhibit design.

Identifying with the work

Novices and experts alike recognized that Pixar technical designers create movies. Some study participants imagined what it would be like to be a Pixar designer. For example, "I thought of seeing myself trying to do all the stuff in the actual movie, how cool would that look" [Novice]. Others recognized that it requires a team of designers to create a movie "because everything does need to have some kind of attention" [Novice]. When discussing how Pixar designers used the real world to create believable imagery, novices and experts discussed how this task involved manipulating variables on a computer. Compared to novices, experts were able to provide greater detail about what real-life decisions entail:

> You have to know what you are trying to portray and know its patterns in order to portray it in a believable sense. So you have to know that these kinds of fish swim together and these kinds of fish are solitary, and these kinds of fish go together to protect themselves … . [Expert]

Although both groups remarked on the attention to detail required of Pixar designers, experts identified how this work can be a creative and "playful iterative thing" [Expert]. Experts also identified with Pixar designers by reflecting on their own prior experiences, which included discussing knowledge beyond resources provided by the exhibits or reflecting on their own skills and strategies. For example, "The light rays are usually going through [the objects] perfectly, they're not real light, and they're just rays that are being shot out" [Expert]. This depth of knowledge led some experts to share that the Pixar exhibits were only showing "the tip of the iceberg" when it comes to doing the work of animation and/or computer programming.

Describing interactions with activity controls and variables

Participants' interactions with and understanding of variables at each of the exhibits contextualized their learning and sense-making at each exhibit. Variables were parameters (for example, straight or bendy grass) that participants could interact with by moving an on-screen slider or turning a button switch on or off. Participants used a range of resources and strategies to make sense of these variables, such as naming variables, comparing (qualitatively and quantitatively) different states of a single variable, comparing the effects of two or more variables, and appraising variables based on prior knowledge. Experts and novices used all strategies, but experts used more quantitative description and variable appraisal based on prior knowledge. For example, one novice group thought the most interesting part of an exhibit was seeing "how [buttons] change the color and these make the texture look awesome and these just make it bumpy" [Novice]. When this question was asked to an expert, they utilized prior knowledge to consider how the exhibit's algorithm functioned, which they felt added to their level of interest, "I was trying to make [the grass] shorter. But I think maybe the algorithm does it proportional[ly], because when I made it shorter, it didn't make it shorter" [Expert].

Recognition of programming and algorithms

Novice responses about computer programming aligned with the basic ideas and concepts at the exhibits, such as recognizing that "computer programming comes in to make it look more realistic" [Novice]. Experts were able to elaborate about the specific mechanisms of what the computer is doing to create realistic scenes, such as a field of grass, relaying that

> Each [grass blade] is pretty simple by itself, but when you take sort of this whole thing and you multiply it out, each one has slightly different settings you start to get this beautiful effect that actually looks like real grass. [Expert]

When recognizing relationships between math and computer programming, novices focused on math more directly related to the exhibits. These included concepts such as modeling, shading, and coding. For example, one novice liked that "you got to control the fish just by simple codes like that" [Novice]. Experts recognized and/or described a greater variety of math, such as probability, functions, modeling, physics, multiplication, and coding. They also discussed how computers can use math to create variation and randomness in a scene, such as using

> Procedural graphics … you make a lot of stuff that looks the same so that you don't have to make individual grass blades, you can just make an algorithm and then it will fill in the whole field. So that's how people do grass and forests. Even cities. [Expert]

Study participants recognized the intricacies and the role of automation in the animation process, although the level of granularity differed between groups. For example, while novices understood that "even small things like grass and trees and rocks and stuff have to be done with high tech kind of things" [Novice], experts provided greater detail on this idea and mentioned that one can control individual aspects, as well as groups of objects ("It's millions of objects, so you can't just do them one at a time, you need some way to control it") [Expert]. Experts also identified that the outcome is based on the inputs and/or parameters set by the designer.

Suggested exhibit enhancements

While novice critiques focused more within the parameters of the exhibit, such as adding more variables or backgrounds, experts tended to focus more on the exhibits' programming piece. For example, one novice said they wanted to change "how [the grass] is kind of going in different directions" [Novice] in order to make the scene they wanted. In contrast, one expert wanted to add an additional slider to compare two variables in one exhibit, saying "It's clear that air is different from water and glass is different from diamond. But if I saw a slider, I could understand a little more that it was all sort of on the same range" [Expert]. Experts were also concerned about the clarity and portrayal of programming and algorithm concepts. For one expert, "an algorithm is a bunch of code, and this is a bunch of fun little slide bits, which is different to me" [Expert]. Another pointed out that the exhibit interactions were more complex than the controls given to visitors and was trying to investigate "what other parameters were preset behind the scenes" and pointed out that there was "obviously more code here than is being brought to the surface" [Expert]. This sentiment is not surprising, however, as experts overall desired a better understanding of how the exhibit programs and variables functioned, as some felt that these interactions did not match their expectations.

Designing embedded supports to help novices think more like experts

An analysis of visitor behaviors focused on understanding the relationship between visitor actions and thought, and on characterizing different strategies that experts and novices use to make sense of and learn about computational thinking. Information learned from Phase One helped to generate specific ways in which novices may need support to develop more expert-like thinking strategies. We incorporated these insights into the final designs of the six exhibits we studied and in strategies used throughout the exhibition. Overall, the exhibits tried to reveal authentic connections between the programming process and the computational thinking involved so that novices could understand the work that goes into creating scenes and images from the Pixar movies, as well as build an understanding of how it is achieved. It was also important to convey that this work can be fun and highly creative.

Authentic context

One strategy embedded within these exhibits was contextual and process-oriented authenticity to support novice recognition of programming, as well as their agency and identity around doing this type of work. In Phase One, experts were more likely to discuss the playful nature and benefits of using computers to do this work. Emphasizing the creative nature of computer programming was an important goal of the exhibition in part to help break down stereotypes of the work and to broaden its appeal. Based on how experts identified with these tasks, supports were designed to help novices make the connection to programming and understand that this work can be playful. Embedded supports helped convey that Pixar's designers use computers to approach real world problems.[25] The exhibits relayed authentic challenges that Pixar's designers have to grapple with when creating a movie scene, as well as authentic processes for working toward a solution. For example, rather than designing the sliders in “Programming Natural Variety” to change the on-screen grass in real-time, a “generate” button was added that helped participants understand that any changes s/he created, as the designer, had to be computed using a computer (Figure 3). These supports helped both groups speculate about the amount of work that goes into creating movies, as well details about what that work entails.

Authentic language

In our study, novice participants did not always relate programming to the exhibit activity, even if they thought it connected to animation. Experts had more prior knowledge to build on and connect to their understanding of the exhibit task while using more quantitative and qualitative description when discussing and manipulating variables at each interactive. Based on how experts identified connections to programming and described activity variables, exhibit developers added symbols and programming language to foster novice recognition of computational thinking and knowledge integration. Language associated with programming was added throughout the exhibit labels, which encouraged novices to recognize that they were taking part in a process and/or task related to programming or math. For example, moving sliders in “Crowd Simulation Workstation” correlated to numbers on screen (Figure 4). This helped novices understand that manipulations entailed calculations within the computer. This embedded strategy used concrete signals to help novices connect their actions to programming.

Opening the black box

Novices were primarily outcome oriented when thinking about exhibit modifications, which included adding or suggesting the ability to manipulate different objects. As such, experts spoke more about “how” they would program different objects. To bring novices more in line with the way experts wanted to understand how a program functioned and how variables related to one another in the activity, strategies were needed to support understanding the beginnings of how programmers would make changes. This entailed revealing underlying mechanisms or the algorithms behind the exhibit program without overwhelming novices to show novices more of what is behind the activity and help make that extra connection to programming. A “more info” button was added to exhibits in order to “open the black box,” and describe the underlying mechanism or algorithm associated with each activity.

Phase Two: affordances of exhibits and embedded supports on novice youth

The aim of Phase Two was to investigate how novice youth interact with and learn computational thinking skills from different exhibit experiences. It was anticipated that aspects of learner identity, including gender, prior experience, and attitudes and interest in computing, would influence learning behaviors and outcomes. This Phase identified the affordances of using any combination of computational thinking focused exhibits, as well as impacts that individual exhibits had on female participants. Below are the range of outcomes that were achieved when youth interacting with nearly final exhibits about problem decomposition.

- ***Youth interest in learning about and doing programming increased after exhibit use.*** Youth were asked how if at all their interests changed immediately after using the combination of exhibits. About half (55%; $n = 116$ of 211) of youth were more interested in learning about computer programming or making their own computer programs (50%; $n = 105$ of 209). A quarter of youth (27%; $n = 58$ of 211) said that they were more interested in programming careers. The solution exploration exhibits in particular were associated with greater levels of change in interest in computer programming.
- ***Youth perceptions of problem decomposition in computer programming became more sophisticated after exhibit use.*** Based on youths' pre and post beliefs and perceptions of computer programming, 62% ($n = 132$ of 213) of youth indicated a more sophisticated understanding of the process of problem solving and problem decomposition after using the exhibits. Additionally, the extended-post survey found that students' beliefs about how experienced programmers solve problems changed after exhibit use, with students more students agreeing that there are a variety of ways to break down problems and that experienced programmers break problems down into steps.[26]
- ***Youth perceptions of creativity in computer programming increased after exhibit experiences.*** Based on youths' pre and post beliefs and perceptions of computer programming, research found that youth considered computer programming to be more creative than they had previously thought. Girls experienced greater changes in their beliefs about the creativity of the work of computer programming when controlling for pre-visit beliefs.[27]
- ***Youth self-efficacy for computer programming increased after exhibit use.*** Self-efficacy is an important measure of a youth's confidence in their abilities to accomplish a specific task, and also predict their ability to self-regulate. Youth had significant gains in their self-efficacy for programming after using the exhibits.[28]
- ***Youth understood the power and value of math and programming after exhibit use.*** Youth were asked to discuss what was most interesting or surprising about what you learned immediately after using the combination of exhibits. Half of the participants (55%; $n = 122$ of 227) shared that decomposing or using variables in problem solving was the most interesting. However, some of their responses reflected surprise or revelation about the power of programming and math, as well as how math and programming can be powerful tools for creating realistic imagery.

Our analysis also found that girls started with lower interest in computer programming prior to using the exhibits, however, two exhibit approaches worked particularly well for this audience. It appears that solution explorations, which guided an individual through an expert's solution to complex problem, worked well in raising interest in computational thinking, particularly to engage and heighten interest in girls. Girls spent significantly more time at these activities than boys and rated them "very interesting" significantly more frequently than boys.[29] Additionally, multimedia narratives, which shared how a complex problem had been broken down into functional parts, could be effective for increasing youth understanding of problem decomposition, especially for girls.[30] This was present even when controlling for pre-visit problem decomposition capacity.

The affordances of engaging novice youth in computational thinking exhibits demonstrate that these exhibits were successful in helping novice youth recognize that they were doing work related to programming. While this research was conducted in a museum context, the design of the three different types of exhibit experiences aligns with design strategies common across learning environments. We would welcome further research that considers how these scaffolding strategies could be translated into formal education experiences. It would also be interesting to understand the potential effectiveness of these strategies in conveying topics beyond problem decomposition in computer science.

Furthermore, embedding research into the exhibition design process helped the *Pixar* development team better articulate their philosophy of learning for exhibits focused on problem decomposition, as well as differentiate the value of the different modalities for conveying this content. This research helped the team clarify their thinking around problem decomposition and detail the hoped-for outcomes of visitors' interactions. In the end, this feedback strengthened the impact that the *Pixar* exhibition could have on youths' capacity for problem decomposition. The development team hopes to incorporate research into future exhibition projects so that this depth of learning is prevalent from the start of an exhibition's creation process.

Notes

1. Wing, "Research Notebook: Computational Thinking."
2. Ibid.
3. NSF project number: CNS 1339244.
4. *The Science Behind Pixar* first opened at MOS in June 2015 and is scheduled to tour nationally and internationally through 2027.
5. Brennan and Resnick, "Using Artifact-based Interviews."
6. Reiser, "Scaffolding Complex Learning," 274.
7. Quintana, Zhang, and Krajcik, "A Framework for Supporting Metacognitive Aspects," 237.
8. Schunn and Nelson, "Expert-novice Studies," 3.
9. Hmelo-Silver, Marathe, and Liu, "Fish Swim, Rocks Sit, and Lungs Breathe," 319; Schunn and Nelson, "Expert-novice Studies," 3.
10. Ibid.
11. Quintana, Zhang, and Krajcik, "A Framework for Supporting," 236.
12. Reiser, "Scaffolding Complex Learning," 277.
13. Schunn and Nelson, "Expert-novice Studies," 1.
14. See note 11 above.
15. Reiser, "Scaffolding Complex Learning," 278.
16. Quintana, Zhang, and Krajcik, "A Framework for Supporting," 240; Reiser, "Scaffolding Complex Learning," 284.
17. Reiser, "Scaffolding Complex Learning," 280.
18. Reiser, "Scaffolding Complex Learning," 284.
19. Grover and Pea, "Computational Thinking in K-12," 40.
20. Novice groups were primarily recruited through posts to the museum's social media platforms and experts were recruited through local college and tech professional listervs.
21. Demographics included gender, age, and race/ethnicity.
22. Differences in their interpretive approaches were identified through inductive, open-coding coding techniques using NVivo coding software; Patton, *Qualitative Research and Evaluation Methods*; Corbin and Strauss, *Basics of Qualitative Research*.

23. Exhibit combinations included: (1) two multimedia narratives & two solution explorations, (2) two multimedia narratives & two creative design activities, (3) two solution explorations & two creative design activities, or (4) all six exhibits.
24. Data were analyzed through inductive, open-coding in Excel, while quantitative data were analyzed in SPSS; Patton, *Qualitative Research and Evaluation Methods*; Corbin and Strauss, *Basics of Qualitative Research*.
25. Grover and Pea, "Computational Thinking in K-12," 39; Barr and Stephenson, "Bringing Computational Thinking to K-12," 50.
26. Immediate post-qualitative problem decomposition task; Extended post-survey-paired t-test: $t = -1.189$; $p = 0.032$; effect size (cohen's d) = 0.23; $n = 81$).
27. Extended post-survey; girls: ($B = .201$; $p < 0.024$); ($B = .594$; $p < 0.001$; overall model adjusted $R^2 = .421$; $p < .001$).
28. Extended post-survey; Of the 13-item measure that captured youths' beliefs and preferences for computer programming, three of these items pertained to creativity in computer programming; ($B = .201$; $p < 0.024$); controlling for pre-visit beliefs ($B = .594$; $p < 0.001$; overall model adjusted $R^2 = .421$; $p < .001$).
29. Timing: Girls spent an average of 10 min, 39 s (SD = 4:18) at these exhibits, while boys spent 8 min, 39 s (SD = 5:03). Interest: (81.5% of girls and 59.3% of boys).
30. ($B = 0.13$; $p < 0.05$); Adjusted $R^2 = .57$).

Acknowledgements

"Building Computational Thinkers through Informal Exhibit Experiences" would not be possible without the leadership of PI Christine Reich and Co-PI Ben Wilson. A special thanks also goes to "The Science Behind Pixar" exhibition development team who designed and created the exhibits for this project, as well as the R&E research assistants who spent many hours helping with data collection and entry for this project.

Disclosure statement

No potential conflict of interest was reported by the authors.

Funding

This work was supported by National Science Foundation [Grant Number 1339244].

About the authors

Leigh Ann Mesiti (M.Ed.) is the Assistant Manager of Research and Evaluation at the Museum of Science, Boston (MOS), where she has worked for the past eight years. During that time, she conducted research and evaluation for a range of MOS exhibitions and programming, as well as multi-institutional studies. At MOS, Leigh Ann collaborateed with in-house exhibit teams to design and prototype new interactive exhibits. Her most recent work has focused on *The Science Behind Pixar* traveling exhibition and the associated research project, *Building Computational Thinkers through Informal Exhibit Experiences*.

Alana Parkes (M.Ed.) is the Supervisor of Exhibit Content Development at the Museum of Science, Boston. She is responsible ensuring the work of three content developers aligns with the goals of the Museum and each specific exhibit and for implementing the pedagogical approach for developing exhibits. She has worked as an Exhibit Content Developer for 18 years. Her skills include defining the overall scope of an exhibit, creating exhibit messages and goals, developing interactives, and writing label copy. Her most recent project is the 13,000-square foot traveling exhibit, *The Science Behind Pixar*.

Sunewan C. Paneto is a Research and Evaluation Assistant at the Museum of Science, Boston, where she has worked for the past seven years. She has assisted on various research and evaluation projects for both exhibitions and museum programs. She worked on the summative evaluation for the *Science Behind Pixar* exhibition and was also part of the *Building Computational Thinkers through Informal Exhibit Experiences* project.

Clara Cahill was the originating Principal Investigator of the *Building Computational Thinkers through Informal Exhibit Experiences* research study. At the Museum of Science, Boston, she worked on the summative evaluation for the *Science Behind Pixar* exhibition and was also part of the *Building Computational Thinkers through Informal Exhibit Experiences* project.

ORCID

Alana Parkes http://orcid.org/0000-0003-0121-4572

Bibliography

Barr, Valerie, and Chris Stephenson. "Bringing Computational Thinking to K-12: What is Involved and What is the Role of the Computer Science Education Community." *ACM Inroads* 2, no. 1 (2011): 48–54.

Brennan, K., and M. Resnick. "Using Artifact-based Interviews to Study the Development of Computational Thinking in Interactive Media Design." Paper presented at annual AERA, Vancouver, BC, Canada, 2012.

Corbin, Juliet, and Anselm Strauss. *Basics of Qualitative Research: Grounded Theory Procedures and Techniques*. Thousand Oaks, CA: Sage, 2015.

Grover, Shuchi, and Roy Pea. "Computational Thinking in K-12: A Review of the State of the Field" *Educational Researcher* 42, no. 1 (2013): 38–43.

Hmelo-Silver, Cindy, Surahbi Marathe, and Lei Liu. "Fish Swim, Rocks Sit, and Lungs Breathe: Expert-Novice Understanding of Complex Systems." *Journal of the Learning Sciences* 16, no. 3 (2007): 307–331.

Patton, Michael Quinn. *Qualitative Research and Evaluation Methods*. Newbury Park, CA: Sage, 2002.

Quintana, Chris, Meilan Zhang, and Joseph Krajcik. "A Framework for Supporting Metacognitive Aspects of Online Inquiry Through Software-Based Scaffolding." *Educational Psychologist* 40, no. 4 (2005): 235–244.

Reiser, Brian J. "Scaffolding Complex Learning: The Mechanisms of Structuring and Problematizing Student Work." *The Journal of Learning Sciences* 13, no. 3 (2004): 273–304.

Schunn, C., and M. Nelson. "Expert-novice Studies: An Educational Perspective." In *Psychology of Classroom Learning: An Encyclopedia.* Macmillan Reference USA/Gale Cengage Learning, 2009.

Wing, J. M. "Research Notebook: Computational Thinking—What and Why?" *The Link: News from the School of Computer Science*, Spring. https://www.cs.cmu.edu/link/research-notebook-computational-thinking-what-and-why, 2011.