From the museum to the browser: Translating a music-driven exhibit from physical space to a web app

S. M. Astrid Bin[1,2], Christina Bui[2], Benjamin Genchel[2], Kaushal Sali[2], Brian Magerko[2], Jason Freeman[1]
Center for Music Technology[1] & Expressive Machinery Lab[2]
Georgia Institute of Technology
Atlanta, GA
a.bin, cbui6, bgenchel3, ksali3, magerko, jason.freeman @gatech.edu

ABSTRACT

This paper describes the process of developing a browser-based version of GrooveMachine, a tangible museum exhibit that aims to foster interest in computer science (CS) through the music-driven exploration of a computational system. GrooveMachine is aimed at kids aged 10-14, and specifically targets learners from from groups currently under-represented in computing by demonstrating CS applications that challenge stereotypes. While an observational study suggests that GrooveMachine triggers situational interest, long-term engagement with CS requires this interest to be deepened and developed. To provide an opportunity for interest development, we have implemented a browser-based GrooveMachine. This not only offers the opportunity for learners to continue their exploration of CS through creative interaction, but provides a pathway to other music and CS learning platforms where they can deepen this interest. In this paper we describe the theoretical underpinnings of interest, how it relates to CS, and how it intersects with identity. We also describe the differences between the museum and browser contexts. We detail the design and implementation of GrooveMachine in the museum and explain how we translated it to the browser, including the rationale behind our central design decisions and a discussion of our technical implementation. In this way we provide valuable insight for researchers who want to reach larger audiences by developing browser-based versions of physical installations.

CCS Concepts

•Human-centered computing → User interface design; •Applied computing → Sound and music computing; •Information systems → *Browsers;*

Keywords

collaborative computing, user interfaces, interfaces for music, children, CS learning

Web Audio Conference WAC-2019, December 4–6, 2019, Trondheim, Norway.

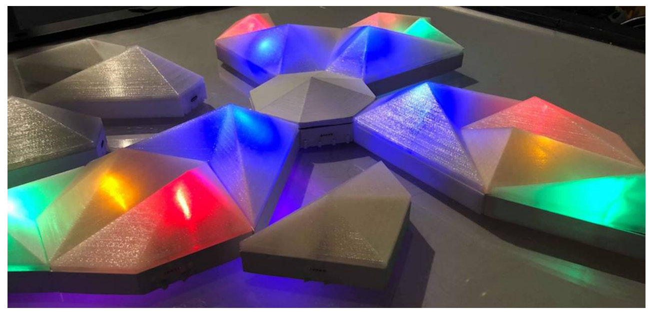

Figure 1: GrooveMachine's tangible blocks

1. INTRODUCTION

GrooveMachine is a tangible, interactive museum exhibit that aims to foster interest in computing through music-driven exploration of a computational system. It is aimed at children aged 10-14, and places specific focus on children from groups currently under-represented in computing. This paper describes how we translated GrooveMachine from a physical museum exhibit to a browser-based web app. First, we discuss why generating interest in computer science (CS) is important, the role music can play in this process through identity, and how this relates to broadening participation in computing. We also discuss the differences between the museum and browser contexts and their implications for design. Second, we describe the museum implementation of GrooveMachine, and briefly describe an observational study that suggests triggered situational interest. Third, we detail the design decisions made in the browser implementation of GrooveMachine, including our approach to technical implementation, in order to provide insight for researchers interested in broadening participation in computing through browser-based interactive environments.

2. BACKGROUND

2.1 Why interest matters

Computer programmers invent, design and develop the technology that affects all of our lives. Despite its broad impact, women and people of colour (PoC) are vastly under-represented in the technology-creating population [9, 22]. As a result, women and PoC are not reaping the economic benefits of CS careers; the technology that affects us all is designed by and for a small segment of the population, and untold human potential is going unrealised. As Bennedsen states, "introducing students to computing is still one of computing education's grand challenges" [5]. Strategies for greater inclusion are needed.

Joining the technology-creating population requires a high degree of intrinsic interest in CS, meaning a desire to engage in computing not because one has to, but for the sake of it [2]. CS courses are challenging with high failure rates [17], but intrinsic interest is a major factor persevering through obstacles, including poor teaching and a lack of support [22].

Though intrinsic interest is sometimes viewed as an inborn trait, Hidi and Renninger propose that this interest can be developed. They suggest that there are two broad types of interest, *situational* and *individual*, and that interest develops over four phases. Situational interest, which has Triggered and Maintained phases, begins with environmental stimuli and is highly motivating, but does not last. However, this Situational Interest can be developed into Individual Interest (through the Emerging and Well-Developed phases), producing an enduring motivation to re-engage with the subject matter in question [14].

2.2 Using music to affect computing identity

Though women and PoC are under-represented in the technology-creating population, there is not a lack of engagement with technology in these groups. Many young Black men have a passion for video games yet don't engage with learning CS [8], and girls in high school may be enthusiastic about but computing but few pursue it at the college level [12] (and those that do are more likely to drop those classes [7], and those remaining are more likely to leave academic careers at early stages [16]).

Instead, stereotypes about technology creators cause people in these under-represented groups to *disidentify* with CS. Computer scientists are stereotyped as male and socially awkward [24], which conflicts with young Black men's concept of masculinity [8], and stereotypes of CS as a solitary pursuit clashes with young women's tendency to want to work with others [22]. The result is that many young people assume that CS is "not for me" [23].

It is here that STEAM approaches (using the Arts to drive engagement with Science, Technology, Engineering and Maths) can bridge the gap between interest and identity. GrooveMachine builds on existing research that demonstrates the effectiveness of music as a driver for CS engagement, particularly among under-represented groups [11, 20], because music's wide cultural relevance gives learners an opportunity to engage with CS via a creative activity that is meaningful to them. Further, music-driven engagement demonstrates the creative potential of computing, which challenges the stereotypes that typically serve to disidentify.

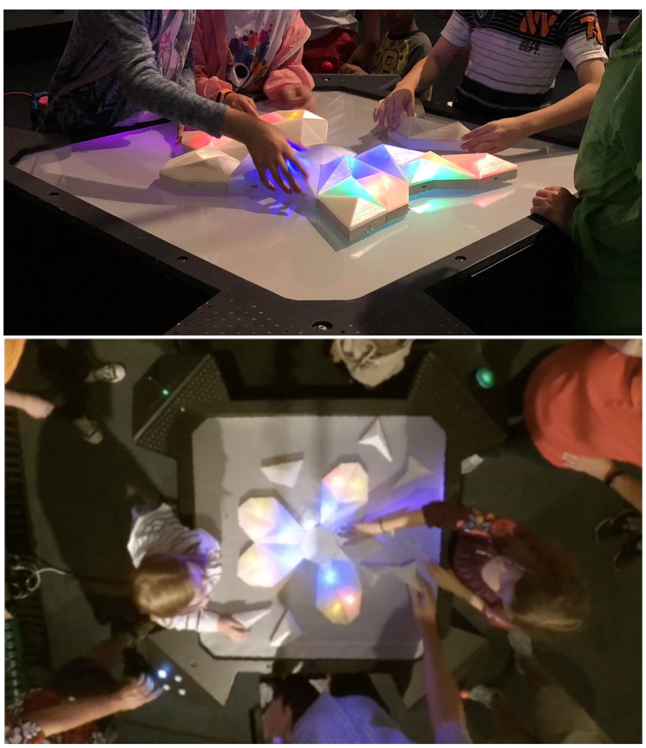

Figure 2: Top: Players exploring GrooveMachine in the museum. Bottom: Overhead view of GrooveMachine in use.

2.3 Fostering interest in the museum

GrooveMachine is a museum exhibit. The museum context is ideal for exploratory learning because museums are focused on fostering curiosity [19, p 33]. They are full of novel, hands-on, multisensory exhibits, where visitors can follow their own interests without obligation or constraint [1]. However, these same characteristics that make museums rich learning environments also present significant challenges for designing learning experiences: lots of exhibits are vying for attention and an exhibit must continuously engage learners, because they are free to walk away from any exhibit they find uninteresting.

The nature of CS presents another dimension of challenge for the museum context. CS is a process of identifying a problem, devising a detailed plan to solve it, and translating those instructions into a language a computer can understand [25]. A pre-existing, robust understanding of the nature, makeup, capabilities and function of computational systems and ways of understanding and approaching computational problems must be in place before writing any code, let alone seeing the outcomes and potential rewards of learning CS. As museum learners follow their own interests, voluntarily participate, and can leave at any time, focusing on the scaffolding knowledge necessary to meaningfully use a computer – knowledge which has little relation to the act of programming [17], let alone the compelling possible outcomes – is unlikely to retain learners.

Because of these challenges in maintaining attention in the museum, considerable interest has developed in active prolonged engagement (APE) [1, 15], which refers to visitor engagement with interactive exhibits. APE has a number of characteristics, such as positive collaboration, meaningful discussion, and prolonged engagement with the exhibit. Our observational study of GrooveMachine (discussed in Section 3) measures one of these aspects, prolonged engagement.

2.4 The museum and browser contexts

The situational interest triggered in the museum is motivating, but if it is not further developed it soon dies off. Because GrooveMachine is part of an online, music-driven CS learning ecosystem [1], there are two opportunities presented by a browser-based version: to provide the opportunity to develop interest through further exploration of GrooveMachine, and to provide a pathway to discovering these other platforms for deeper exploration.

The most influential factor in translating GrooveMachine from the museum to the browser is the differences in *context*, as both places are not only very different in terms of their physical, social, interactive and cognitive affordances, but also in what people expect from interaction.

Context is a primary consideration in interface design. Bannon [3] established the concept of a *human actor* in HCI theory, proposing that the context in which a system is used is a primary consideration in how it should be designed. There are stark contrasts between the museum and browser contexts, across the axes of who the person is with, why they do things in that context (their motivations), and the features of the experience:

Who are they with? Science museum exhibits are designed to support collaborative, group interaction. By contrast, a computer browser has one set of controls and is designed for a single user.

Why are they here? Museums are places of learning, and visitors voluntarily take part based on their personal interests. Browsers, however, have a multiplicity of uses: they are used to play games, engage in communities, search for information, communicate with others, complete tasks.

How do they experience it? There are three main differences in experience. First, museums are full of novel, unique experiences and visits are occasional, whereas browsers are familiar interfaces that most of us use them every day. Second, museums are multimodal, using "full sensory and expressive capabilities including visual, sonic, haptic, and kinesthetic/proprioceptive" [6], whereas browsers are primarily experienced visually and aurally. Third, museum interaction is embodied, as visitors "physically explore concepts and systems by moving within and acting upon an environment" [6], while a browser is largely disembodied with users interacting through the physical manipulation of a mouse to control a pointer on the screen.

3. MUSEUM IMPLEMENTATION

3.1 Tangible and musical interaction

GrooveMachine is a tangible tabletop exhibit. The interface is based on a step sequencer, with the table divided into eight radial "steps". In the middle of the table is a hub. The stepper moves around the table, and lights in the hub indicate which section is active.

[1]https://tunepad.live, http://earsketch.gatech.edu

Figure 3: Rendering of the GrooveMachine exhibit.

To construct musical patterns, players attach tangible blocks to the central hub. There are two types of blocks, samples and modifiers. Samples can attach to the hub, and modifiers can be added to the samples. When a block is attached successfully it illuminates from the inside, indicating that it is recognised (see Figure 1).

The shape of these tangibles are derived from the Islamic system of geometry (specifically a four-fold star pattern). This approach was chosen for a number of reasons: these patterns are beautiful and deeply mathematical [18], tessellation has been shown to be an effective method of engaging children in mathematical exploration [10], and the symmetry of this layout means that players can look to the actions of others to gain intuition of what to do.

Most importantly, tessellation is a method of assembling instructions that is dependent on exploration, and not on a pre-existing knowledge of computational syntax. In the museum environment, exhibits must continually engage learners in order to retain their attention and curiosity, and here we give curious players the means to discover the nature of this system not through learning its prerequisites, but instead through the opportunity to act [26].

3.2 Physical form

GrooveMachine is a square table. The square shape fosters collaboration (by encouraging players to distribute themselves around it as there are four obvious places to stand), and also places constraints on the player (as not everything is within easy reach). Since players cannot easily reach all the steps of the sequencer, they must negotiate with others.

Each of the table's four corners features arcade controls, chosen because they are familiar to kids and invite interaction. Each of these sets of controls affects a different global variable of the GrooveMachine system: volume, the genre of the music, tempo, and the direction in which the step sequencer is travelling.

3.3 Connection to computing

The CS content in GrooveMachine is delivered via embodied metaphor. As learners interact, they engage with metaphors for computational systems: the loop (the step sequencer), computational objects (the samples), paramaterisation (the modifiers), variables and variable scope (the arcade controls). In this way we establish and reinforce mental models of computing through a fun experience driven by music.

3.4 Technical implementation

GrooveMachine contains an embedded hardware network of four Arduino Megas, and a central Mac Mini. The Arduino Megas track the samples and their modifiers as they arrive and leave. The Mac Mini controls audio playback, moving the stepper around the table and playing audio samples based on the tangibles that are present in a given step. The audio playback, which uses pre-recorded samples and some real-time processing, is implemented using Pyo (a Python-based DSP library).

3.5 Hold time study

In July 2018 we conducted a preliminary hold time study at Chicago's Museum of Science and Industry, where we installed GrooveMachine on the open museum floor. We observed groups of visitors use the installation, noting the number and approximate ages of group members, and timing their interaction. We wanted to determine if the lengths of time that our target audience spent with GrooveMachine could be considered "prolonged": since one of the characteristics of situational interest is focused attention triggered by the environment [13], long hold times could be a first indication. In the seminal APE studies the average time spent at APE exhibits was 3:18 [15, p 13] (similar to findings in [4]), so we used this as our threshold for "prolonged" engagement.

We observed 72 groups, with an average hold time of 4:26. We performed a one-tail t test to determine if this hold time was higher than the threshold for "prolonged" interaction, and found that it was statistically significantly higher (M=4:26, SD=2:59, $t(71)=3.2596$, $p=<0.01$).

33 of these groups included at least one child in our target age range of 10-14, and this group had an average hold time of 5:34. We performed an independent-samples one-tail t test to compare the average hold time of this subset with the 3:18 threshold, and found that it was statistically significantly higher (M=5:34, SD=2:57, $t(32)=4.38101$, $p=<0.01$).

More in-depth study is needed to determine the quality of and reasons for this effect, but this does suggest that GrooveMachine is triggering situational interest. In the museum context learners engage voluntarily, can leave at any time if they are uninterested, and there are plenty of exhibits competing for their attention, but GrooveMachine captures the interest of learners in our target age range for significantly longer than the average "prolonged engagement" time threshold.

This suggestion of situational interest is promising, but gave us pause. Hidi and Renninger indicate that though motivating, situational interest will die off if not deepened through further exploration [14]. Because our broadest aim is to foster interest in computing among people underrepresented in computing, just triggering situational interest through music-driven exploration is not enough. A browser-based version of GrooveMachine was the next logical step to create an opportunity to further develop interest in CS through music, and to provide a pathway to other learning platforms that can take learners further.

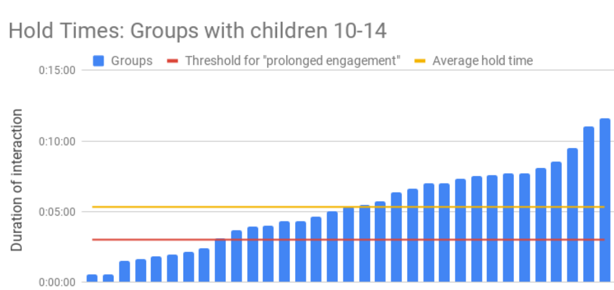

Figure 4: Hold time durations of the 33 observed groups containing at least one child in the 10-14 age range. Red line: Prolonged engagement threshold. Yellow line: Average group hold time.

4. MOVING TO THE BROWSER

As discussed in Section 2.4, there are profound differences between the museum GrooveMachine and the browser GrooveMachine, and these are a function of differences in context. In order to translate between the two contexts, we first had to determine the GrooveMachine's core aspects that are independent from context, both *formal* (design features that are stable), and core *functional* (effects on the person as a result of using it). Through team discussion we determined that the core formal aspects are the step sequencer interface that is activated through adding tessellating blocks, and that the core functional aspects are exploration and discovery.

4.1 Formal aspects: Interaction and UI

The interface of the web-based GrooveMachine is an octagonal space with a central hub. On the top on the left are outlines of the GrooveMachine blocks. Along the bottom are controls for the global variables, as well as a button to save a groove and buttons to toggle the code visualisation (see Figure 5).

The browser version differs from the physical installation in a number of specific ways:

Octagon shape: In the museum, GrooveMachine's physical form is square, to encourage collaboration. In the browser these collaborative and physical aspects are not relevant, so we adopted an octagonal interface to reinforce the step sequencer's loop metaphor.

Tangible drawers: GrooveMachine's tangible blocks are placed on and around the table allowing learners to explore their shape and try them out. In the browser we wanted to avoid visual clutter but still maintain the element of discovery, so we placed these into expandable "drawers" that expand when the shape is clicked.

Connection to computing: Using GrooveMachine in the museum requires learners to engage in physical metaphors for computing, and we have the benefit of being able to add printed material around the exhibit to drive this connection. In the browser we have the opportunity to link this music-computer relationship more directly by including code visualisation. When the code toggle at the bottom is clicked, code is visualised on top of the interface. This is updated in real time, and includes both system state and executing functions (see Figure 5, bottom).

Variables: In the museum, GrooveMachine's arcade controls allow learners to manipulate global variables, and place them outside the interaction to embody a metaphor for their global scope. In the browser embodied metaphor is not possible, but we did make these controls readily accessible at the bottom of the interface (Figure 5).

The core mechanic of GrooveMachine is placing blocks onto the steps of the sequencer, and a block causing the system to produce a specific sound when that step is active. We maintained this interaction, so learners drag blocks from the collapsible drawers and onto the interface. The difference in the interactions is spatial: In the museum the spatial manipulation of tangibles is an important element of the interaction, but this isn't possible in a browser. In the browser version, learners drag blocks from the drawer to the interface, and as a block passes over the interface it automatically orients itself to fit in a given step.

4.2 Functional aspects: Exploration and discovery

GrooveMachine is designed with exploration and discovery in mind, and though future work is needed to determine the precise connection between this and extended hold times we wanted to preserve these features. Exploration is a key factor in experiencing this computational system, and for this reason both the museum and the browser have no required onboarding, and instead are tolerant to learners trying things out and visually responding when they get it right (in the museum version the tangibles light up, in the browser the blocks are outlines when being dragged, and filled in when attached). (Though there is no onboarding for the browser, we have included a simple Help box that provides minimal instruction, as there are no other learners to watch or guides to ask.)

In both versions, discovery is key. In the museum learners can move between exploring the tangibles and working the arcade controls, can learn from watching others, and can move to other vantage points if they wish. These physical aspects aren't relevant in the browser, but we included discovery by locating the tangibles in collapsible drawers. All other controls are readily available, but to find tangibles learners need to find them.

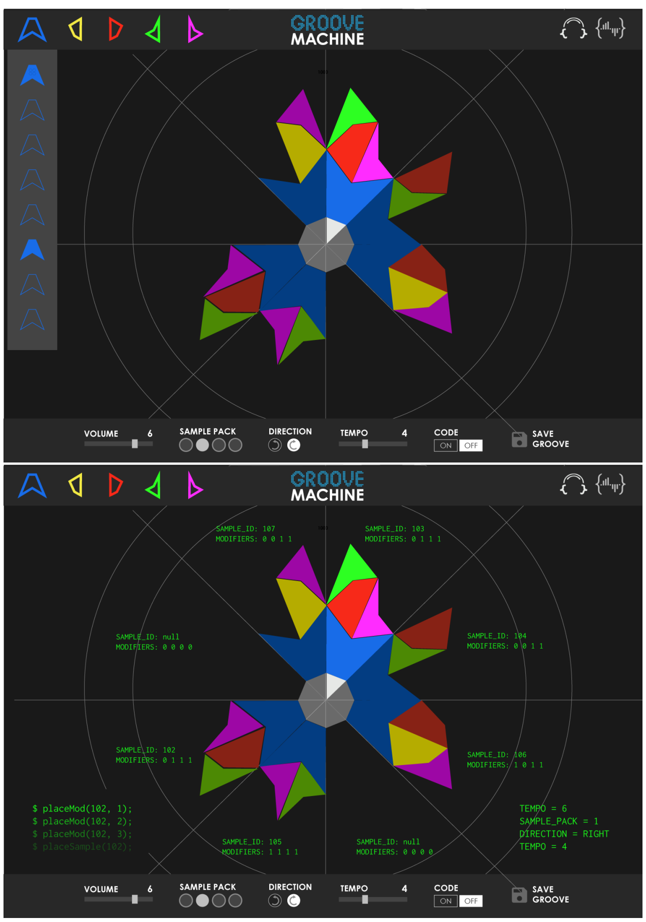

Figure 5: GrooveMachine browser interface. Top: Interface with drawer open. Bottom: Interface with code layer visible.

4.3 Technical implementation

To build the browser-based interface we used PixiJS[2], a 2D WebGL rendering library that uses HTML5 and JavaScript. This allowed us to build an interface where pieces could be dragged and dropped, emulating the physical table interaction.

Though Pixi is powerful and easy to use, a caveat is that it is not ideal for designing UI elements. For example, the collapsible drawers holding the blocks are implemented using a combination of Pixi and HTML. Slide-out drawers allow for discovery and are straightforward to implement in HTML, but since the tangibles are Pixi elements they must be contained in the canvas. To mitigate this, the expansion is triggered by an HTML element (the shapes at the top of the interface), but the expanding drawer is a Pixi element.

All audio in the browser is handled by Tone.js [21], described as "a framework for creating interactive music in the browser". Tone.js is built on top of the Web Audio API[3], and provides scheduling, synths, effects, and buffers for reading and playing back audio files.

At present, all sounds are read from a set of audio files that are downloaded from a database while the application loads. Certain effects that can be applied to the base samples via effects tangibles, namely high pass and low pass filtering, are not realized in real time but stored in the database as static files as well. A file exists for the high-passed and low-passed versions of each base sample. Other effects, such as sample reversal, are provided by Tone directly and applied in real time.

The timing for the entire application is driven by Tone's internal clock. When the application loads, a Tone.js Sequence object is created that invokes a call back function 8 times with the iteration step as argument. The callback function has two advantages: it advances PIXI's game clock, eliminating the need for PIXI's own internal clock which is timed by frame count, and it plays the sample and associated effects for whichever step of the virtual table is currently active. Additionally, it checks for and applies global changes such as tempo and volume change. The Sequence object runs in an indefinite loop.

4.4 New affordances from the browser

Though some physical aspects of GrooveMachine in the museum are lost when we move to a web app, the browser does offer some unique advantages.

Easy iteration and expansion. A drawback of physical installations is that production is labour-intensive and expensive, and changing aspects compounds this. The browser-based GrooveMachine, however, does not have these constraints, and we can easily and quickly iterate on its design and function in response to testing.

Located in a computer. One of the challenges of the GrooveMachine exhibit has been connecting this experience with computing. This has been addressed somewhat with printed didactic material around the exhibit, but making that connection without causing disidentification because of negative stereotypes is challenging. In the browser, this connection is obvious. Browsers are ubiquitous and familiar, and seeing code on a screen is not out of place. In this way we can direct interest further towards computing instead of

[2]https://www.pixijs.com/
[3]https://www.w3.org/TR/webaudio/

simply having a fun experience.

Easy connection to other platforms. GrooveMachine is part of an ecosystem of online CS learning environments that are music-based (EarSketch and TunePad). These are considerably more in-depth but offer much deeper exploration of CS through music. A browser-based GrooveMachine offers an easy connection to these platforms, enabling learners to develop their interest in CS to a more profound level than is possible in a museum interaction.

5. CONCLUSIONS AND FUTURE WORK

GrooveMachine is designed to foster interest in CS, particularly among children from underrepresented groups. There are preliminary indications that it triggers situational interest in the museum, but for this interest to last it must be developed. In order capture and deepen this triggered interest we developed a browser-based version of GrooveMachine. Through careful consideration of GrooveMachine's core features we translated this exhibit from museum to browser, and we leveraged web-based tools to implement it.

Our next steps are to refine this interface through testing, to develop a pathway from museum to the browser, and to build infrastructure between GrooveMachine, EarSketch, and TunePad (such as single sign-on, exporting and importing projects, etc). To broaden participation in computing experiences that challenge computing stereotypes are needed, as well as ways for these learners to develop lasting personal interest in CS.

6. ACKNOWLEDGMENTS

We gratefully acknowledge the National Science Foundation for their ongoing support of this project (grant 1612644).

7. REFERENCES

[1] S. Allen. Designs for learning: Studying science museum exhibits that do more than entertain. *Science education*, 88(S1), 2004.

[2] M. Apiola, M. Lattu, and T. A. Pasanen. Creativity and intrinsic motivation in computer science education: experimenting with robots. In *SIGCSE*. ACM, 2010.

[3] L. J. Bannon. From human factors to human actors: The role of psychology and human-computer interaction studies in system design. In *Readings in HCI*. 1995.

[4] C. Barriault and D. Pearson. Assessing exhibits for learning in science centers: A practical tool. *Visitor Studies*, 2010.

[5] J. Bennedsen and M. E. Caspersen. Failure rates in introductory programming: 12 years later. *ACM Inroads*, 2019.

[6] D. Birchfield, H. Thornburg, M. C. Megowan-Romanowicz, S. Hatton, B. Mechtley, I. Dolgov, and W. Burleson. Embodiment, multimodality, and composition: convergent themes across hci and education for mixed-reality learning environments. *Advances in Human-Computer Interaction*, 2008.

[7] L. Carter. Why students with an apparent aptitude for computer science don't choose to major in computer science. *SIGCSE Bulletin*, 2006.

[8] B. DiSalvo, S. Yardi, M. Guzdial, T. McKlin, C. Meadows, K. Perry, and A. Bruckman. African american men constructing computing identity. In *CHI*. ACM, 2011.

[9] B. J. DiSalvo, K. Crowley, and R. Norwood. Learning in context: Digital games and young black men. *Games and Culture*, 2008.

[10] R. S. Eberle. The role of children's mathematical aesthetics: The case of tessellations. *The Journal of Mathematical Behavior*, 35, 2014.

[11] J. Freeman, B. Magerko, T. McKlin, M. Reilly, J. Permar, C. Summers, and E. Fruchter. Engaging underrepresented groups in high school introductory computing through computational remixing with earsketch. In *SIGCSE*. ACM, 2014.

[12] S. Graham and C. Latulipe. Cs girls rock: sparking interest in computer science and debunking the stereotypes. In *SIGCSE Bulletin*. ACM, 2003.

[13] S. Hidi. Interest: A unique motivational variable. *Educational research review*, 2006.

[14] S. Hidi and K. A. Renninger. The four-phase model of interest development. *Educational psychologist*, 41(2), 2006.

[15] T. Humphrey and J. P. Gutwill. *Fostering active prolonged engagement: The art of creating APE exhibits*. Routledge, 2017.

[16] M. Jadidi, F. Karimi, H. Lietz, and C. Wagner. Gender disparities in science? dropout, productivity, collaborations and success of male and female computer scientists. *Advances in Complex Systems*, 2018.

[17] T. Jenkins. On the difficulty of learning to program. In *LTSN for Information and Computer Sciences*, 2002.

[18] D. H. Kaplan, Craig S & Salesin. Islamic star patterns in absolute geometry. *ACM Transactions on Graphics*, 23(2), 2004.

[19] G. D. Lord and B. Lord. *The manual of museum management*. Rowman Altamira, 2009.

[20] B. Magerko, J. Freeman, T. Mcklin, M. Reilly, E. Livingston, S. Mccoid, and A. Crews-Brown. Earsketch: A steam-based approach for underrepresented populations in high school computer science education. *Transactions on Computing Education*, 2016.

[21] Y. Mann. Interactive music with tone. js. In *Proceedings of the 1st annual Web Audio Conference*. Citeseer, 2015.

[22] J. Margolis, A. Fisher, and F. Miller. The anatomy of interest: Women in undergraduate computer science. *Women's Studies Quarterly*, 2000.

[23] J. Margolis, J. Goode, and D. Bernier. The need for computer science. *Educational Leadership*, 68(5), 2011.

[24] A. Master, S. Cheryan, and A. N. Meltzoff. Computing whether she belongs: Stereotypes undermine girlsâĂŹ interest and sense of belonging in computer science. *Journal of Educational Psychology*, 2016.

[25] D. Sleeman. The challenges of teaching computer programming. *Communications of the ACM*, 1986.

[26] H. Van der Meij. Principles and heuristics for designing minimalist instruction. *Technical communication*, 1995.