HandsOn: enabling embodied, creative STEM e-learning with programming-free force feedback

Gordon Minaker[1], Oliver Schneider[1], Richard Davis[2], and Karon E MacLean[1]

[1] University of British Columbia, Vancouver BC, Canada
[2] Stanford University, Stanford CA, USA

Abstract. Embodied, physical interaction can improve learning by making abstractions concrete, while online courses and interactive lesson plans have increased education access and versatility. Haptic technology could integrate these benefits, but requires both low-cost hardware (recently enabled by low-cost DIY devices) and accessible software that enables students to creatively explore haptic environments without writing code. To investigate haptic e-learning without user programming, we developed *HandsOn*, a conceptual model for exploratory, embodied STEM education software; and implemented it with the *SpringSim* interface and a task battery for high school students. In two studies, we confirm that low-cost devices can render haptics adequately for this purpose, find qualitative impact of *SpringSim* on student strategies and curiosity, and identify directions for tool improvement and extension.

1 Introduction

Recognition of the value of a hands-on, embodied approach to learning dates to 1907, when Maria Montessori opened a school where she used *manipulatives* to teach a wide array of concepts ranging from mathematics to reading, e.g., by introducing the alphabet through children tracing their finger along large, cut-out letters [13]. Constructivist learning theories posit that well-designed manipulatives can assist understanding by grounding abstract concepts in concrete

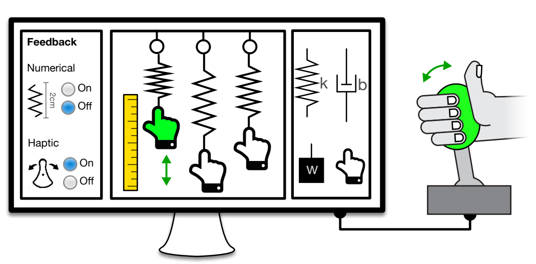


Fig. 1. Students, teachers, and researchers can explore science, technology, engineering, and math (STEM) abstractions through low-fidelity haptics, incorporating elements into system designs.

representations [15, 16], and are an accepted core principle in early math and science education, confirmed empirically [3].

More recently, digital technologies are radically altering learning environments. Massive Open Online Courses (MOOCs) expand access, games motivate, and with graphical simulations (e.g., PhET [18]), students can interact with abstractions to develop their understanding. However, these experiences are disembodied. Indirect contact via keyboard, mouse and screen introduces a barrier of abstraction that undermines the connection and path to understanding.

Haptic (touch-based) technology should bring benefits of physicality and embodied learning [5] to interactive virtual environments. It adds a sensory channel as another route to understanding [2]; when deployed appropriately, active exploration can improve understanding [12] and memory [7] of new concepts. Haptic tools have already shown promising results in many specializations, demographics and age groups, both to enhance lesson fidelity and to increase engagement and motivation through tangibility and interactivity; e.g., with devices like Geomagic Touch[3] [19] and SPIDAR-G [17].

Unfortunately, existing approaches have both hardware and software limitations. Actuated learning tools introduce physical issues of cost, storage, and breakage; devices are too bulky, complex, or expensive for schools or self-learners. For software, it is hard for users to construct and explore their own haptic environments. Typically, users load a virtual system to interact with it haptically. This sidelines the rich learning potential of involving users with model construction [15]. We address hardware with the HapKit [14], a $50, simple, low-fidelity device constructed from 3d printed materials.

Our focus here is on software, with a new learning environment that lets users both construct and explore haptic systems. Until now, the only way for a user to construct a haptic system was by programming it herself. Our approach, inspired by Logo [15] and Scratch [11], is to ultimately provide much of the power of a programming language while hiding distracting complexity.

Approach and Present Objectives: To study *how* to unlock the potential of hapticized virtual environments in STEM education, we need a viable front-end. To this end, we first established a *conceptual model* (*HandsOn*): central interface concepts, supported operations and language [9] that can be employed in a broad range of lessons involving physical exploration and design.

Next, we implemented the *HandsOn* conceptual model (CM) in *SpringSim*, a first-generation learning interface prototype narrowly focused in a module on mechanical springs and targeted at high school physics students. To render forces we used the HapKit, a simple device with a 3D-printable handle providing affordable, self-assembled 1 DOF force-feedback for about $50 USD. As an evaluation instrument, this single-lesson implementation allows us to (a) measure a given hardware platform's fidelity for a representative perceptual task; (b) attain insight into the kinds of lessons such a system can leverage; and (c) assess its learning-outcome efficacy relative to conventional methods. With these answers, we will be able to design a more powerful tool.

[3] Prev. Sensable Phantom www.geomagic.com/en/products/phantom-omni/overview

We report results from two user studies: (1) the HapKit's ability to display differentiable springs with and without graphical reinforcement, and (2) a qualitative evaluation of *SpringSim* for a carefully designed set of educational tasks. We confirm that the *SpringSim* interface and its conceptual model *HandsOn* are understandable and usable, describe the role of haptics compared to mouse input, and provide recommendations for future evaluation, lesson and tool design.

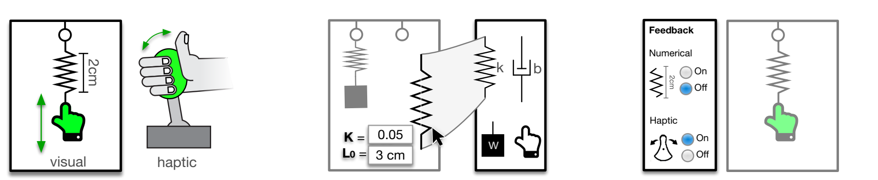
2 Tool Development: Conceptual Model and Interface

Our goal was to find a software model to use and evaluate low-cost force feedback in an educational setting. We began by choosing a device, establishing requirements, and exploring capabilities through use cases and prototypes. From this, we defined *HandsOn*. We then implemented essential features in a medium-fidelity prototype, *SpringSim*, for our user studies.

Initial design (requirements): We established six guiding requirements. First, we developed initial prototypes with HapKit 2.0 through two pilot studies with middle school students (described in [14]). These highlighted two aspects of a practical, accessible approach for junior students: 1) no programming; instead 2) a graphical implementation of an exploratory interface within a lesson plan. We also needed to build on known benefits of traditional classroom practices, and enable learning-outcome comparison. We must 3) support the same *types* of traditional education tasks, e.g., let students compare and assemble spring networks as easily as in a hands-on physics lab; but also 4) *extend* them, to leverage the flexibility offered by a manipulative that is also virtual. Similarly, to support future formal comparisons, our model needs to 5) support both haptic and non-haptic (mouse) inputs. Finally, to ensure generality we also needed to 6) support diverse STEM topics, like physics, biology, and mathematics. Further design yielded a model that addressed these requirements: *HandsOn*.

Conceptual Model: *HandsOn* is a programming-free (R1) graphical interface supporting learner exploration (R2), with a number of key *concepts*: *Interactive Playground*, *Hands*, *Design Palette*, *Objects*, *Properties*, *Haptic* and *Visual Controls*. Exploration is supported at various levels (Figure 2).

The *Interactive Playground* provides a virtual sandbox where users can interact with virtual environments (VE). *Hands* allow users to select, move, and



A) Interact with the system in the *Interactive Playground* using a selected *Hand*, manipulating and monitoring state via multimodal feedback.

B) Create and Modify the system with a *Design Palette*, adding or removing *Objects* and changing object properties.

C) Customize interaction itself for learners, teachers, and researchers, adjusting input/output modalities with *Visual Controls* and *Haptic Controls*.

Fig. 2. The *HandsOn* CM enables three kinds of exploration based on requirements.

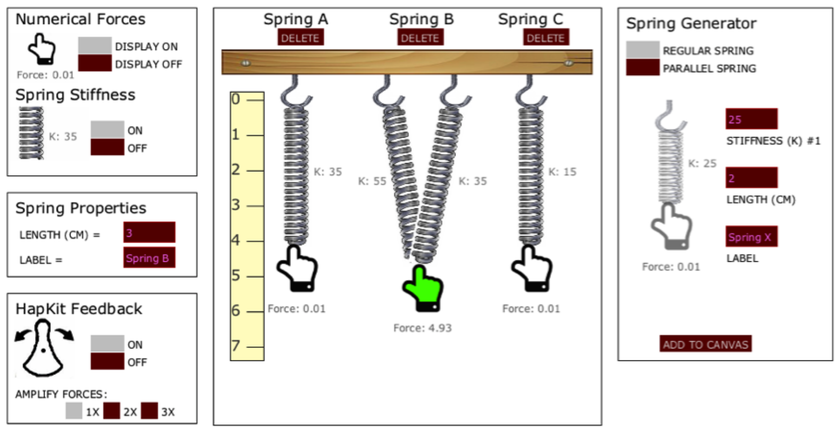


Fig. 3. *SpringSim* interface, a *HandsOn* sandbox for a single lesson module on springs.

manipulate components in the Interactive Playground. Control occurs with either the mouse or a haptic device to receive force-feedback (Figure 2A) (R5). In the design and modification phase, users can add or remove *objects* like springs, masses, gears, or electrons by dragging them to and from a *Design Palette* (R3). Once added to the scene, users can modify their physical properties (e.g., a spring constant k) and make changes to the VE (Figure 2B). After construction, the user can customize their interaction with their VE by adjusting *Visual Controls* and *Haptic Controls* options that extend interactions in new ways afforded by haptics (R4) (Figure 2C). Because of the flexibility afforded by having multiple *objects* in the playground with multiple *Hands* for interaction points, and customization of interaction and feedback, *HandsOn* can support different STEM topics (R6), from biology to mathematics. To confirm the viability of this approach, we built an initial prototype with essential features: *SpringSim*.

Implemented Prototype: Our first *HandsOn* interface is *SpringSim* (Figure 3), which supports a spring lesson – spring systems are natural as a virtual environment of easily-controlled complexity. In *SpringSim*, *objects* include single springs and parallel spring systems, with properties spring rest length (cm), stiffness (N/m) and label. The *Design Palette* includes the *Spring Properties* and *Spring Generator* UI components. Implemented *Visual Controls* are toggling numerical displays of spring stiffness and force; *Haptic Controls* toggle HapKit feedback and output amplification. The open-source repository for SpringSim is available at https://github.com/gminaker/SpringSim.

3 Study 1: Perceptual Transparency

Before evaluating *SpringSim*, we needed to confirm that the HapKit could render spring values sufficiently for our qualitative analysis.

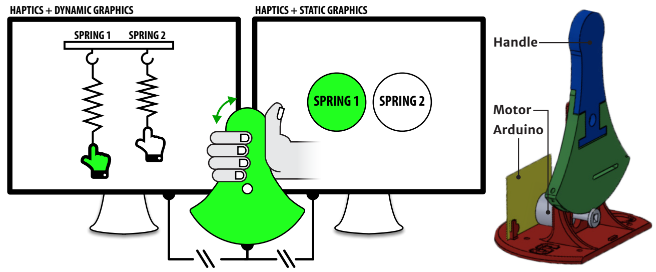


Fig. 4. In the *Hapkit+Dynamic Graphics* condition, graphical springs responded to input (left); static images were rendered in the *Hapkit+Static Graphics* condition (right); in both, HapKit 3.0 [14] was used as an input/force-feedback device (far right).

Methods: 14 non-STEM undergraduate students (8 females) participated in a two-alternative, forced choice test with two counterbalanced within-subject conditions: *HapKit + Dynamic Graphics*, and *HapKit + Static Graphics* (Figure 4). Three spring pairs (15/35, 35/55 and 55/75 N/m) were each presented five times per condition, in random order. For each pair, participants indicated which spring felt more stiff, and rated task difficulty on a 20-point scale. Following each condition, participants rated overall condition difficulty, mental demand, effort, and frustration on 20-point scales derived from the NASA TLX [8]. Following the completion of both conditions, a semi-structured interview was conducted to address any critical incidents. Each session lasted 20-30 minutes.

Results: All tests used a 5% level of significance and passed test assumptions.

Accuracy: A logistical regression model was trained on task accuracy with spring-pair and condition as factors. No interaction was detected; spring-pair was the only significant factor. Post-hoc analysis revealed that spring-pair #1 (15/35 N/m) was significantly less accurate than spring-pair #2 (35/55; p=0.0467). Performance averaged 88.57% (15/35), 96.49% (35/55), and 94.45% (55/75).

Time: Task time ranged from 3-160s (median 117s, mean 96.41s, sd 47.57s). In a 3-way ANOVA (participant, spring-pair, and visualization condition) only participant was significant ($F(13, 336) = 4.17$ $p = 1.947e - 06$).

Difficulty rating: A 3-way ANOVA (factors: participant, spring-pair, and visualization condition) detected one two-way interaction between participant and spring pair ($F(26, 336) = 2.10$, $p = 0.00165$).

Discussion: Study 1 revealed that (a) for stiffness intervals 15/35/55/75 N/m, the HapKit provides distinguishability equivalent to dynamic graphics. Individual differences influenced difficulty and speed, suggesting that learning interfaces may need to accommodate this variability. (b) Accuracy was not dependent on individual differences, suggesting that learning interfaces can consider task time and perceived difficulty separately from accuracy when using the HapKit (at least, for these force ranges). (c) Performance was mostly above 90%, and confidence intervals for our small sample size estimate no lower than 82% accuracy at

Task	Bloom	Description
1	Understand (2)	Rank three springs in order from least to most stiff
2	Understand (2)	Plot the relationship between displacement and force for two springs.
3	Apply (3)	Estimate the stiffness of an unknown spring, given two reference springs with known stiffness value
4	Analyze (4)	Predict the behaviour of springs in parallel.
5	Create (6)	Design a parallel spring system that uses two springs to behave like an individual spring of stiffness 55 N/m.
6	Apply (3)	Predict the behaviour of springs in series.
7	Evaluate (5)	Describe any relationships you have noticed between spring force, displacement, and stiffness.

Table 1. Learning tasks used with *SpringSim* in Study 2. *Bloom* level is a measure of learning goal sophistication [1]

the lowest (15/35). We speculate that the HapKit's natural dynamics are more pronounced at lower rendered forces, and may interfere with perceptibility.

4 Study 2: Tool Usability and Educational Insights

Methods: 10 non-STEM participants (1st and 2nd year university undergrads with up to first year physics training, 6 female, 17-20 years) volunteered for 45-60 minute sessions. After an introductory survey, participants were randomly assigned to one of two conditions, *Mouse* (4 participants, M1-4) or *Hapkit* (H1-6). HapKit 3.0 was calibrated for force consistency between participants. After allowing participants to freely explore *SpringSim*, a survey assessed understanding and usability of various *SpringSim* interface components; misunderstood components were clarified. Three exit surveys elicited value of *SpringSim* components on 7-point Likert scales, cognitive load [10], understanding, and curiosity on 20-point scales, and preferred learning modality [6], respectively.

Learning Tasks: We iteratively designed and piloted a task battery of escalating learning-goal sophistication [1] to expose strategies for force feedback use and general problem-solving (Table 1). Tasks did not require physics knowledge, and were suitable for both mouse and HapKit input.

Analysis: We conducted t-tests on self-reported understanding, cognitive load, engagement, understanding, curiosity; and on objective metrics of time-on-task and number of spring interactions. Qualitative analysis of video and interview data used grounded theory methods of memoing and open & closed coding [4]. Together, these yielded insight into the usability of *SpringSim* and the *HandsOn* CM, and several themes describing the role of haptics in our tasks. Two participants were excluded from analysis of Task 1 due to technical failure.

Results - Usability: After free exploration of *SpringSim*, participants rated their understanding of CM objects (yes/no) and their ease-of-use [1-7]: *Ruler* (10/10, 7.0), *Numerical Force Display* (10/10, 6.5), *Playground* (10/10, 6.0), *Hand* (9/10, 6.0), *Spring Properties* (9/10, 6.0), *Spring Generator* (**7/10, 5.0**), *HapKit* (6/6, **4.5**), and *Haptic Feedback Controls* (5/6, **4.5**). While generally usability was good, interface clarity needed improvement in highlighted cases. Participants specifically noted confusion on radio button affordances, and *Spring Generator* input fields (due to redundant availability in *Spring properties*).

Results - Task Suitability for Haptic Research: Regardless of prior physics knowledge, all participants were able to complete education tasks 1-6 (Table 1) in the allotted 60 minutes. We found no evidence that any task favoured one condition over another. When participants in the mouse condition were asked how their workflow would change with physical springs, participants weren't sure: "I don't know if that would've given me more information" (M4).

Results - Haptics & Learning Strategies: We observed several themes relating to the influence of force feedback on a student's learning strategy.

Haptics creates new, dominating strategies. Learning strategies used by participants in the HapKit condition (H1-6) were more diverse than those in the mouse condition (M1-4). In Task 1, M1-4 all followed the same strategy, displacing all 3 springs the same distance and comparing the numerical force required to displace them. They then correctly inferred that higher forces are associated with stiffer springs (the *displace-and-compare* strategy).

By contrast, all 5 H participants included in analyses (H2 excluded due to technical failure) used force-feedback as part of their approach to Task 1. H1 describes applying the same force to the HapKit across all 3 springs, recording displacement to solve the task, while H5 described looking at the speed at which the HapKit was able to move back-and-forth in making his determination of stiffness, rather than through direct force-feedback of the device. Only H6 indicated that he "looked at the numbers for a sec", but no participant fully used the *displace-and-compare* strategy we observed for M participants.

While the single-strategy approach worked for easy tasks, it was linked to errors and dead-ends in at least one instance in the mouse condition. In Task 5, M2-4 used *displace-and-compare* to validate their newly designed spring; M1 did not seek verification of his design. In contrast, H1,2,5,6 used haptic feedback to verify their designs. They did this by comparing how stiff their parallel spring system felt to a target reference spring. H4 guessed at an answer without verification. H3 used the *displace-and-compare* strategy, checking that equal forces were required for equal displacement.

Haptic impressions of springs are enduring and transferrable. HapKit participants were able to use their previous explorations to solve problems. In Task 3, M1-4 interacted with all three springs to find a ratio between force and stiffness. However, H participants interacted with springs fewer times (mean 1.5, sd 3.21) than M (6, sd 1) ($p=0.018$). H2-4,6 did not interact with any springs, and H1 interacted with only one. This was because they had already interacted with the springs in previous questions: "I remember spring C was less stiff" (H3). Further suggesting the strength of haptic impressions, when H1 designed an inaccurate spring system for Task 5 ($k=80N/m$ vs. expected $k=55N/m$), she described the haptics as overriding the visual feedback: “they just felt similar. Even though the numbers weren’t really relating to what I thought.” Similarly, H2 arrived at an approximate result (k=40N/m), after using force-feedback and acknowledges “... [it’s] slightly less than the reference spring, but it’s closer.”

Haptics associated with increases in self-reported curiosity and understanding. Participants’ self-reported curiosity significantly increased over the course of HapKit sessions from a mean of 6.3 (sd 3.83) to 10.8 (sd 3.92) in the Hapkit condition (p=0.041). No significant changes in curiosity were detected in the mouse condition. Participants’ self-reported understanding significantly increased over the course of HapKit sessions from a mean of 3.67 (sd 4.03) to 11.83 (sd 3.19) (p=0.014). No significant changes in understanding were detected in the mouse condition (before: 9.25, sd 5.32; after: 9.25, sd 5.32; p=0.77).

In interviews, participants commonly made references to how the HapKit influenced their understanding: “I can use this thing for help if I really need some physical, real-world stimuli” (H5); “almost all of my thinking was based on how the spring [HapKit] ended up reacting to it” (H6). M2, who had a stronger physics background than others (IB Physics), was the only user to report a drop in curiosity and understanding over the course of the physics tasks, despite initial excitement: “the fun part is messing around with [SpringSim],” he exclaimed near the beginning of the exploratory phase.

5 Study 2 Discussion

Tool and Tasks: Suitability for Learning and as Study Platform

Adequacy and comprehensibility of underlying model: Overall, *HandsOn* concepts proved an effective and comprehensible skeleton for *SpringSim*. Specific implementations rather than concepts themselves appeared to be the source of the reported confusions, and we observed that *HandsOn* should be extended with additional measurement tools (e.g., protractors, scales, calculators, etc).

SpringSim performance: This *SpringSim* implementation adequately supported most students in finishing learning tasks; extending available objects, properties and tasks will support advanced students as well. Future iterations should more clearly map *Design Palette* elements to the objects they support, increasing rendering fidelity and reconsider colors to avoid straightforward affordance issues. While participants did not heavily use haptic and visual controls, we anticipate these will be important for instructor and researcher use.

Learning task suitability: The learning tasks used here were fairly robust to time constraints of user-study conditions, did not require previous physics knowledge, avoided bias from standardized physics lessons, and exposed haptics utilization strategies without penalizing non-haptic controls. Currently, the task set ends by asking students to predict a serial system’s behavior; some students found predicting new configurations a large jump. Future task-set iterations could support integrative, prediction-type questions with interface elements that are successively exposed to allow prediction testing.

Evidence of the Role of Force Feedback in Learning

Curiosity and understanding leading to exploration: Self-reported curiosity and understanding increased when forces were present. While these trends must be verified, curiosity is of interest since it can lead to more meaningful and self-driven interactions. Iterations on both tasks and tool should support this urge with an interface and framing that supports curiosity-driven exploration.

Alternative strategies enabled by force feedback: The HapKit's additional feedback modality enabled alternative task workflows, e.g., estimations of force appeared to supplant mathematical strategies for stiffness estimation. While possibly risky as a crutch, force assessments might be a useful step for students not ready for technical approaches (e.g., M3/Task 3 when stalled in attempting cross-multiplication). Future task-set iterations could encourage more *balanced* strategy use, e.g. mathematical *and* perceptual rather than primarily perceptual.

HapKit salience, resolution & implications: Overall, HapKit 3.0's fidelity was enough to assist participants verify a correct hypothesis. However, those who started with an *incorrect* hypothesis and used only HapKit to test it generally arrived at solutions that improved but were still inaccurate. Given the confidence that forces instilled, this is an important consideration. A formal device characterization will allow us to keep tasks within viable limits; we can also consider using low-fidelity forces more for reinforcement and exploratory scenarios.

Limitations and Next Steps: Our studies were small and used non-STEM university students as a proxy for high-school learners. Despite both limitations, they were useful for our current needs (rich, initial feedback establishing suitability and usability for *HandsOn* through *SpringSim*); but may overestimate general academic ability and maturity. As we we move into evaluation of learning outcome impact, larger and more targeted studies are imperative.

Future interfaces can both increase physical model complexity and breadth (e.g., complex mass-spring-damper systems), and extend *HandsOn* for more abstract education topics, such as trigonometry. We also plan to extend the *Playground* to support more engaging, open-ended student design challenges, such as obstacle courses using trigonometry concepts; this in turn requires new measurement tools and tasks that are more exploratory and open-ended.

6 Conclusions

Haptic feedback's potential in STEM education use can only be accessed with a comprehensible, extendable, and transparent front-end. We present *HandsOn*, a conceptual skeleton for interfaces incorporating virtual forces into learning tasks, and assess its first implementation, *SpringSim* and task set. Our findings (on interface usability, task effectiveness, and impact of haptic feedback on learning strategies, understanding and curiosity) underscore this approach's promise, as we proceed to study haptic influence on learning outcomes themselves.

7 Acknowledgements

We thank Melisa Orta Martinez and Allison Okamura for the HapKit, Paul Bucci for illustration assistance, and Brenna Li for study facilitation. This work was supported by NSF, and conducted under UBC BREB #H14-01763.

References

1. Bloom, B.S., Engelhart, M.D.: Taxonomy of Educational Objectives: The Classification of Educational Goals: By a Committee of College and University Examiners: Handbook 1. David McKay (1969)
2. Calvert, G.A., Brammer, M.J., Iversen, S.D.: Crossmodal identification. Trends in Cognitive Sciences 2(7), 247–253 (Jul 1998)
3. Carbonneau, K.J., Marley, S.C., Selig, J.P.: A meta-analysis of the efficacy of teaching mathematics with concrete manipulatives. Journal of Educational Psychology 105(2), 380–400 (2013)
4. Corbin, J.M., Strauss, A.L.: Basics of qualitative research: techniques and procedures for developing grounded theory. Sage Publications (2008)
5. Dourish, P.: Where the action is: the foundations of embodied interaction. MIT Press (2004)
6. Fleming, N., Mills, C.: Not another inventory, rather a catalyst for reflection. To Improve the Academy 11 (1992)
7. Glenberg, A.M., Gutierrez, T., Levin, J.R., Japuntich, S., Kaschak, M.P.: Activity and Imagined Activity Can Enhance Young Children's Reading Comprehension. Journal of Educational Psychology 96, 424 (2004)
8. Hart, S.G., Staveland, L.E.: Development of NASA-TLX (Task Load Index): Results of Empirical and Theoretical Research. Advances in Psychology 52(C), 139–183 (1988)
9. Johnson, J., Henderson, A.: Conceptual models: begin by designing what to design. Interactions 9(1), 25–32 (2002)
10. Jones, L.A., Hunter, I.W.: A perceptual analysis of stiffness. Experimental brain research 79(1), 150 (1990)
11. Maloney, J., Resnick, M., Rusk, N., Silverman, B., Eastmond, E.: The scratch programming language and environment. Trans. Comput. Educ. 10(4), 1–15 (2010)
12. Martin, T., Schwartz, D.L.: Physically distributed learning: Adapting and reinterpreting physical environments in the development of fraction concepts. Cognitive science 29(4), 587–625 (2005)
13. Montessori, M.: The Advanced Montessori Method, vol. 1. Frederick A. Stokes Company (1917)
14. Orta Martinez, M., Moriomoto, T., Taylor, A., Barron, A., Pultorak, J., Wang, J., Calasanz-Kaiser, A., Davis, R., Blikstein, P., Okamura, A.: 3-D Printed Haptic Devices for Educational Applications. Haptics Symposium (2016)
15. Papert, S.: Mindstorms: Children, Computers, and Powerful Ideas. Basic Books, 2nd edn. (1980)
16. Piaget, J.: Science of Education and the Psychology of the Child. Orion (1970)
17. Sato, M., Liu, X., Murayama, J., Akahane, K., Isshiki, M.: A haptic virtual environment for molecular chemistry education. In: Transactions on edutainment I, pp. 28–39. Springer (2008)
18. Wieman, C.E., Adams, W.K., Perkins, K.K.: PhET: Simulations that enhance learning. Science 322, 682–683 (2008)
19. Williams, R.L., Srivastava, M., Conaster, R., Howell, J.N.: Implementation and evaluation of a haptic playback system. Haptics-e 3, 160–176 (2004)