Embodied Fabrication:
Body-Centric Devices for Novice Designers

by

Benjamin A. Leduc-Mills

B.A., University of Santa Cruz, 2003

M.P.S., New York University, 2008

M.S., University of Colorado, 2013

A thesis submitted to the

Faculty of the Graduate School of the

University of Colorado in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

Department of Computer Science

2014

This thesis entitled:
Embodied Fabrication:
Body-Centric Devices for Novice Designers
written by Benjamin A. Leduc-Mills
has been approved for the Department of Computer Science

Michael Eisenberg

Clayton Lewis

Tom Yeh

Date

The final copy of this thesis has been examined by the signatories, and we find that both the content and the form meet acceptable presentation standards of scholarly work in the above mentioned discipline.

Leduc-Mills, Benjamin A. (Ph.D., Computer Science)

Embodied Fabrication:

Body-Centric Devices for Novice Designers

Thesis directed by Prof. Michael Eisenberg

We present a class of devices under the umbrella moniker of “embodied fabrication”. These devices and the development of the term “embodied fabrication” is rooted not only in computer science, but also in cognitive science, children’s educational theory, digital fabrication technology, and the convergence of these strands present in the do-it-yourself community known as the“maker movement”. As such, we operate under a certain set of premises that guide and direct this work. First, that embodied cognition - which places the body at the center of our cognitive operations - provides a framework in which to ground our decisions to design physical peripheral devices as opposed to purely screen-based software. Second, that a strong line of research supports providing children with tangible, “manipulative” objects to learn with. Third, that digital fabrication technologies - 3D printing in particular - provide a wonderful new opportunity for children and novice designers in general to make, play, and explore creatively - and that the current design options for 3D printers are not suited to meaningful design and creation of objects by non-expert 3D modelers. Finally, that by continuing in the best traditions of body-centric interaction design for children, a suite of devices can be created to provide an educationally and technically rich environment to connect kids to the creative potential of 3D printing. We unpack these ideas more in the introduction, followed by a discussion of all three of the prototype devices and the companion software, a chapter on related work, a chapter discussing the three user studies we performed evaluating these devices, and finally we present a vision of the future of this work and of embodied fabrication devices as a whole.

Dedication

To all the killas and the hundred dolla billas.

Acknowledgements

I was incredibly fortunate to have found the supporting cast that I had during my time at Colorado. My advisor, Michael Eisenberg, is one of the true gems of humanity. An unbelievable scholar, gifted writer, and a wonderfully supportive and warm person. Over our many hours of discussion he taught me how to love and appreciate the world where children's crafts, cognitive science, and computation come together. My father, for urging me to visit Mike as I was applying to grad school. You always let me know how proud you were. You and I are more similar than I ever would have imagined as a child. I see that as a wonderful thing. My mother, for her unwavering faith that I would be excellent. You were much better at convincing me than I was at convincing myself. Your visits were always a chance to recharge my batteries, feel loved, and be fed extremely well. Monika, for taking the risk of leaving your world behind to join me here in Colorado, and for supporting me so well during a very stressful time. You're the best cheerleader a guy could ask for. Many others contributed to this work, academically and socially. I owe an enormous debt of gratitude to Nathan Siedle, whose generosity allowed my work to reach its completion. My committee members were all amazing. Ann Eisenberg kept the lab from self-destruction more times than I can count. My CTG and CU compatriots for welcoming me to Colorado and convincing me to stay. Lindsay Levkoff Diamond, for being an incredibly understanding and flexible boss during my studies. The DOE and all my friends at SparkFun Electronics - Toni, Amanda, Dave, Brennen, Jordan, (etc.) without you, I don't know if I would have made it. And finally, Fran, Anna, Adrian, Victor, Galen, and the amazing kids at Gold Crown for their support. Thank you all.

Contents

Tables

Table

Figures

Figure

Chapter 1

Introduction

Digital fabrication technologies are increasingly finding their way into educational spaces of all shapes and sizes. These new technologies (3D printers, laser cutters, etc.) afford opportunities for exploring these new ways of 'making' and how they may change the way we learn, explore, and play. Although there is much excitement surrounding the 'maker movement' - and 3D printing in particular - there has been little examination of how to introduce a younger audience to 3D printing in an empowering way. This proposal argues that tangible interfaces - as opposed to 2D screen-based media - can be designed not only to support spatial reasoning and mathematical intuitions in children by engaging them in exploratory modeling and play, but that these interfaces can act as a democratizing force by enabling children to create physical objects with digital fabrication devices. The proposed work presents a series of novel tangible input devices for enhancing mathematical and spatial reasoning in kids with a focus on generating output for 3D printing. We discuss related work, the status of the proposed work, additional improvements to be made, a timeline for completion, and a discussion of risks, limitations, and outcomes inherent in the proposal.

A number of computer scientists, technologists, and educators have declared that the era of personal fabrication is upon us[10][50]. New devices aimed at increasing the ability of the individual to physically manufacture their own ideas are being released at breakneck speed. The cultural and technological shifts caused by this change are taking many forms, yet few technologies associated with the 'maker' movement have received as much attention as 3D printing - the ability (by various means) to digitally design and then print out physical 3-dimensional objects. Media outlets from

Forbes[57] to The Economist[1] have extolled the disruptive and democratizing possibilities that 3D printing offers - at least as it affects the traditional manufacturing supply chain. Less examined has been how to introduce novices, specifically pre-teens and early adolescents, to 3D printing - and perhaps more importantly - discussing what (and how) they might learn by being exposed to it. While the variety of desktop 3D printers continues to increase and the cost of adding a 'fab lab' of digitally-based manufacturing tools in the home or classroom steadily declines, the types of interfaces by which children can easily and intuitively design and explore the capabilities of 3D printers still remains a barren landscape consisting primarily of software-only solutions. It is this landscape that we are interested in seeding, following the best practices in computational and cognitive science with particular attention to children-centered design.

To this end, we present a class of tangible user interfaces (TUIs) designed to scaffold a child's ability to design, explore, and play in three dimensions, with a particular focus on enabling output for 3D printing. Significant work has been done on three devices that allow users to specify points on a physical, interactive, volumetric interface that simultaneously displays active points in real-time on a computer. The software on the computer allows for certain modeling operations on the set of input points (e.g., taking the convex hull), as well as exporting shapes to stereolithography (.stl) format, the preferred format for 3D printing. We propose that these designs form a new class of tangible input devices, and present early work on a portable pop-up book building upon the ideas expressed in the two earlier designs. The rest of the paper proceeds as follows: discussion of related work, description of completed work, the proposed work, a timeline for completion, and concluding thoughts.

The work presented here draws on the stages of childhood developmental theories and conception of space developed by Piaget and refined by Papert, notions of cognitive development and embodied mathematics discussed by Lakoff and Nuñez, the democratization of digital fabrication technologies discussed by Gershenfeld and Lipson, and the previous adaptation of these achievements into computer science.

Chapter 2

Prototype Systems

Over the past several years we have been working on the creation of a family of child-friendly tangible user interfaces that would serve as input devices for exploring 3D modeling and digital fabrication in an "embodied" fashion. As discussed in the first chapter, the motivations behind this work are thematically diverse, but can be distilled into an attempt to create a more intuitive, body-centric way for novices to design for 3D printing while also strengthening the user's sense of spatial translation from 3D to 2D (screen based) representations. To this end, we have created three prototypes: the UCube, an initial proof-of-concept device, SnapCAD, a more expressive and capable iteration of the UCube, and PopCAD - a paper-based interface addressing several of the cost and portability concerns raised by SnapCAD. These systems all communicate with versions of a companion software program running on desktop computer. This chapter describes (in chronological order) the development of these three systems, the software that interfaces with them, the motivations behind their design, and the technical work involved in their creation.

2.1 UCube

The UCube represents our first attempt to create a cooperative system of hardware and software that encapsulated and combined our beliefs about embodied cognition and the importance of accessible digital fabrication. The idea for the UCube originally came from the attempt to create a "3D Geoboard". 2.1 shows a rudimentary 2D geoboard consisting of a 3x3 grid of nails stuck into a wooden block. Simple geometries, such as the triangle shown in the referenced image, can be made by stretching rubber bands around some number of “pegs”. The geoboard invites a kind of tangible, exploratory, and embodied play that (as we discuss in Chapter 3) promotes children’s learning in powerful ways. The initial design goal was to capture the “gestalt” of the traditional 2-dimensional geoboard and extend it - into 3-dimensions, and with a computationally-enhanced interface that could translate physical manipulations on a device into a software program that could display the actions performed on the geoboard in a “meaningful” way - that is, in a way that could potentially extend spatial reasoning abilities between the 3D representations created on the device and the 2D, screen-based images displayed on the computer screen.

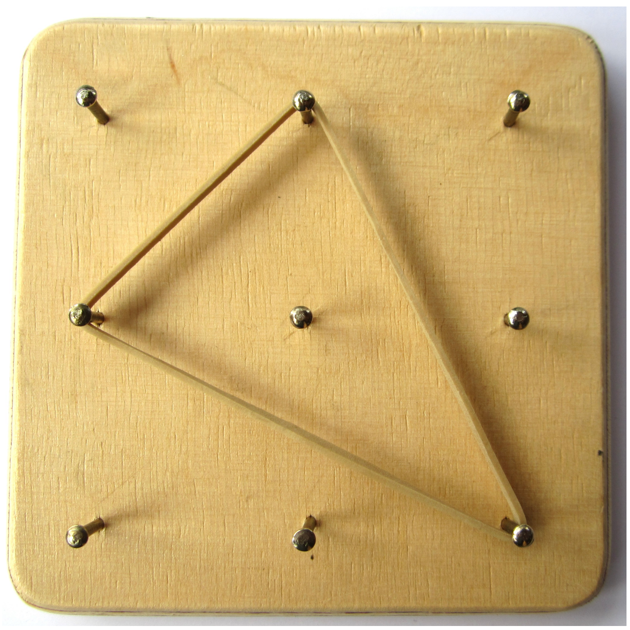


Figure 2.1: A simple 3x3 geoboard, with a rubber band stretched around several pegs, forming a triangle.

The UCube (as seen on the left in 2.2) is the initial result of this goal. The physical interface consists of a set of vertical “towers” that are placed (and optionally re-placed) onto a grid of 4x4 evenly spaced nodes or sockets into which the towers are placed, acting somewhat like the nails in the 2D geoboard. The towers themselves contain four switches placed vertically along the tower, creating a potential for 64 (4x4x4) distinct points to be activated. The towers are “plugged in” when placed into one of the 16 socket nodes, connecting them to the underlying circuitry responsible for providing power to the towers and relaying the state of each of the switches to the computer, via an Arduino Mega[87] microcontroller. Thus, when a tower is placed in a specific node on the board and a switch is flipped on, a particular (x,y,z) coordinate in three-dimensional space is activated and sent to a piece of software on the computer. An abstracted illustration of the hardware system is seen on the right in 2.3.

In turn, the UCube software (discussed more thoroughly later in the chapter) takes the incoming coordinate data from the microcontroller and translates it into a real-time visualization on screen. The graphical user interface centers around a "ghosted" grid of all the potential points, with the active points being highlighted. In the first version of the software, the interface also provides a set of operations that can be performed on the set of active points in addition to normal scene manipulations like zoom and rotate. These functions include: taking the convex hull of the point set (as imagined in 2.2), creating a sequential path or knot through the active points, exporting the convex hull or knot to .STL format for 3D printing, drawing a (non-printable) spline through the active points, saving and loading a shape, and editing the vertices of a convex hull via a click-and-drag interface.

As a first step in discussing the UCube's role in spatial designand in discussing the broader issue of children's three-dimensional designthis section is devoted to a more thorough description of the UCube and its operation. To begin with an overview, then: the UCube system is the combination of two elements: the physical input device of "towers" placed on a board, and the companion display software. These two systems work together to take the embodied actions of the user and display corresponding points and shapes on the computer. A sense of the scale of the device can be inferred from **??**, which shows a photograph of a middle-school student holding a newly-placed tower in the UCube platform while pointing simultaneously at the desktop computer screen beside it. This photographwhich we will also return to in the discussion of pilot testing in a later sectionreflects the essential nature of interaction with the device: points are designated in a spatial region provided by the platform, and then represented in real time on the computer screen.

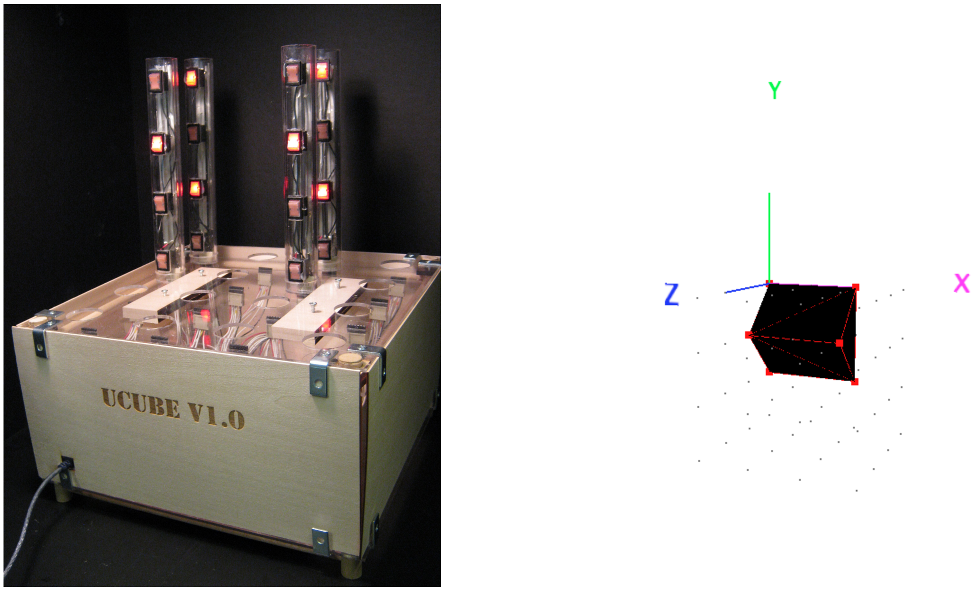


Figure 2.2: Left: The UCube device, with four towers and six lit switches, representing the six vertices of a triangular prism. Right: An early version of the UCube software, representing the convex hull formed with the six active points from the picture to the left.

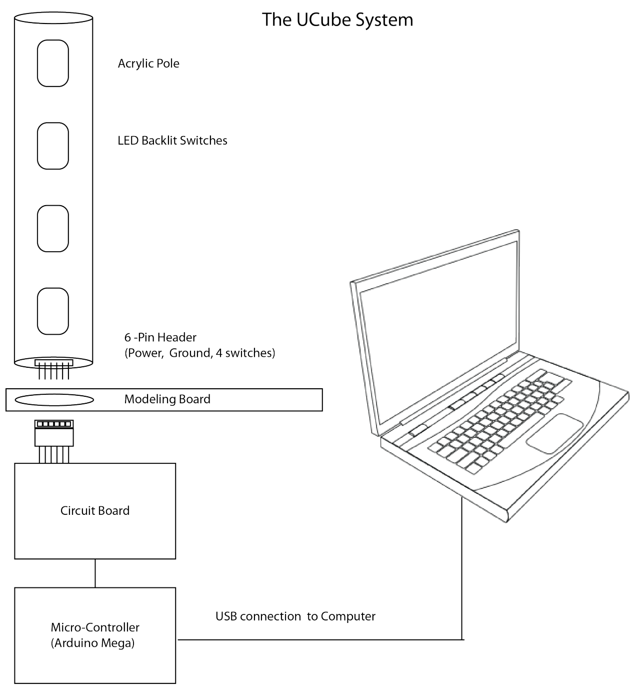


Figure 2.3: A schematic illustration of the UCube hardware.

Thus, the UCube promotes an attention to the correspondence between the selected spatial points above the platform and the (more abstract) representation on the computer screen.

2.1.1 Technical Implementation

The physical system for our first UCube prototype, as outlined earlier, consists of a platform with a four-by-four grid of potential sites, each of which can hold one tower with four switches, thus describing a 4x4x4 array of 64 potential points. The platform structure consists of three different horizontal "layers". The top (or upper surface) layer has a four-by-four grid of circular holes, into which the towers fit snugly. This layer of 1/4" thick laser-cut clear acrylic acts as a brace to hold the towers upright, and ensures that they are resistant to being knocked over. The next layer down holds the headers, which allow the towers to "plug in" and connect to the rest of the circuit. Wires from the headers go down to the bottom layer, which holds the breadboarded circuit and Arduino Mega microcontroller. The towers are made of transparent acrylic, the side paneling of basswood. The towers were laser-cut in order to house the four switches and corresponding circuitry elements. The switches are LED-backlit when active, making it more apparent which points are active as well as giving a more accessible "gestalt" of the shape being modeled. It also allows for some potentially interesting applications in dimly-lit circumstances, such as modeling constellations in a classroom or planetarium: in these situations, the lights of the selected spatial points stand out especially vividly.

Each tower connects to the platform through a six-pin header (one pin each for power, ground, and four switches). The switch connections are then routed through a breadboard containing current limiting resistors for the LED switches to pins on a microcontroller (an Arduino Mega[87]). The Arduino is then able to communicate (via asynchronous serial communication) the active switches (and corresponding coordinates) to the computer through a USB cable. 2.3 depicts a schematic diagram of the UCube hardware.

2.1.2 A Sample UCube Scenario

As a sample scenario, imagine that we wish to create a triangular prism solid employing the UCube. We can begin this process by selecting three points to form a triangle, as shown in Figure 6; then, by placing two more towers and creating the same triangular shape ”shifted over” by two units (Figure 7) we create the entire prism. Naturally, there might be many alternative pathways to forming the same eventual shape: for example, we might begin by placing four (or more) towers in the platform, and then experiment or fiddle with the chosen lights to approach the eventual goal of creating our prism. Alternatively, we might begin without any towers in the device at all: by placing our hands or fingers above the device, roughly indicating where the prism should be, we might then use our imagined locations as ”guides”, helping us to place the necessary towers in the platform and select the correct lights for the vertices of the prism. In any event, having designed the prism using the UCube platform, and having checked that it looks like the correct shape on the computer screen, the final step is to export the shape into a format suitable for 3D printer output. The UCube software, as noted earlier, includes a feature for doing just this; and finally, we print out the prism, as shown in Figure 8. Figure 6. The first step in constructing our triangular prism: here, we create a planar triangular shape toward the left side of the platform, and can see the resulting shape on the computer screen shown at right. Figure 7. Completing the triangular prism. Here, we have added a second (”shifted”) version of our original triangle to produce the six vertices needed to form the prism.

2.1.3 Limitations

It will probably not have escaped the reader’s notice that the UCube, as a three-dimensional modeling device, has significant limitations. To take the most glaring of these: the user can only model those shapes whose vertices are among the sixty-four locations accessible from the device. Moreover, those available locations are evenly spaced in the form of a three-dimensional grid, or lattice; thus, there are numerous simple-but-interesting shapes (such as the regular dodecahedron, composed of regular pentagonal faces) that cannot be designed in the current version of the UCube. Likewise, shapes with curved surfaces (such as a cylinder), demanding at the very least a high resolution of accessible points, could not be modeled in the current UCube. We will return to these issues in the final section of the paper, in the discussion of ongoing and future work.

2.2 SnapCAD

Based on the feedback from these two user studies, a second, more powerful instantiation of the ideas from the UCube has been created. SnapCAD (formerly known as UCube v2) consists of a total input space of 7x7x7 points, forming 343 distinct coordinates. In our user studies with UCube v1, we noticed that users often encountered initial difficulties when required to "find a middle" in the shape they were attempting to model, given an even number of total grid spaces. For example, to model a pyramid on on a 4x4x4 grid, one needs to construct a 3x3 subset of the 4x4 grid, using the middle point within the 3x3 set as the top of the pyramid. This influenced our decision to create an odd-numbered layout, creating a more "natural" middle point in the hardware. The greater number of inputs vastly increases the expressive potential of SnapCAD (compared to the UCube) while still maintaining a manageable interface. SnapCAD, as seen in 2.4 has some very obvious differences from its predecessor.

Working on the scale of multiple hundreds of inputs necessitated the design of custom circuit boards to relay information effectively to the microcontroller. This change in scale also meant rewriting most of the modeling software to effectively handle the greater expressiveness of the physical system.

The use of conductive, magnetic snaps along towers constructed of custom-printed circuit board allow for more than one color of illumination, as different colored LED boards can be snapped onto any socket on the tower. This not only results in the ability to represent multiple shapes at once, but for the SnapCAD to become a platform for all manner of multi-player interactions (e.g. games, puzzles, shape matching contests), with each "player" assigned a unique color. To this end, we have created a simple "3D Tic-Tac-Toe" implementation on the SnapCAD. Additional changes

Figure 2.4: Left: the SnapCAD interface, showing the hardware configuration corresponding to the picture below in 2.5. Right: a detail of the SnapCAD hardware - the PCB tower is housed in a 3D-printed shell, which plugs into a shift-register board. The LED boards snap on to the towers via magnetic snaps.

to the software include supporting multiple but separate convex hulls of different colors, the ability to create and export shapes created from the minimal spanning tree of a set of input points, and the ability to adjust the width of the segments in the knot/path and minimal spanning tree modes. The click-and-drag editing mode now includes the knot/sequential path and minimal spanning tree modes as well as the convex hull mode. We also adjusted the knot-forming algorithm to handle paths that cross or self-intersect, as well as providing a "close knot" button to complete a circuit in a shape, allowing for even more kinds of 3D-printable objects. While significant work has been done to bring the UCube and SnapCAD to their current states, we believe not only that there is room for additional improvements to be made, but that, as opposed to focusing on a incremental but essentially similar interface as the subject of a thesis, it is far more intellectually interesting to focus on a class of objects that demonstrate multiple incarnations of a set of ideas.

2.2.1 A Sample (Red/Green Player) Strategy Game for SnapCAD

Each of the previous sample projects could have been undertaken with the UCube 1.0 (although in every case, the larger scale of the UCube 2.0 allows for a far greater quantitative range of constructionfor example, one can create polyhedra or paths in a 7-by-7-by-7 grid that would be

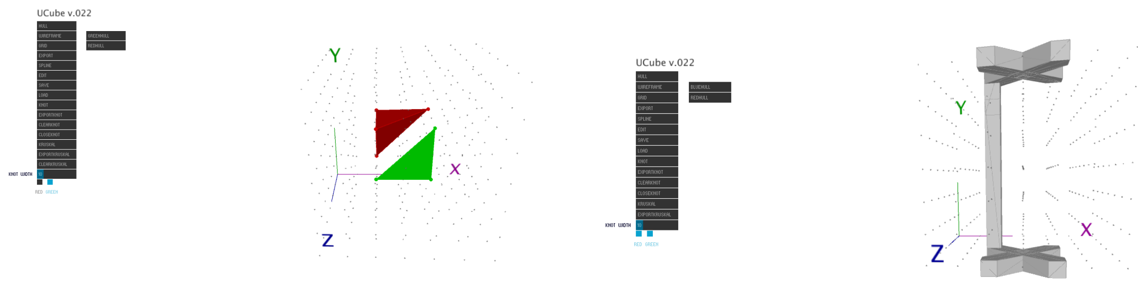


Figure 2.5: Left: The SnapCAD software showing two convex hulls of different colors. Right: the SnapCAD software showing a minimal spanning tree model.

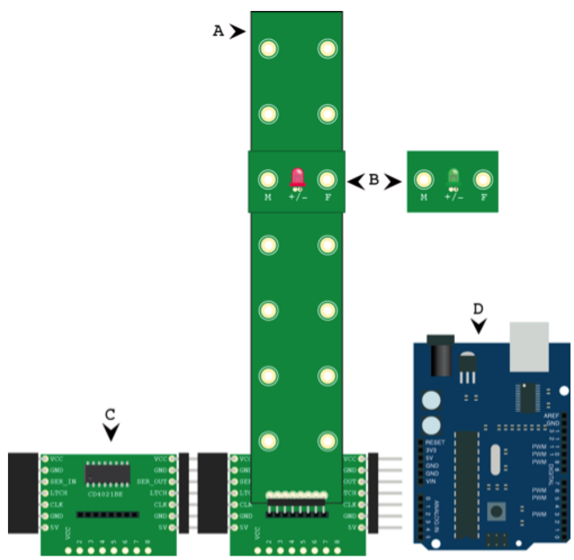


Figure 2.6: A schematic of the SnapCAD technical design, showing a sample tower (A), LED light element (B), shift register board (C) and Arduino (D). The Arduino microcontroller's role is to send coordinates (and colors) of the LED lights, once placed, to a desktop computer. A fuller description of this schematic is provided in the accompanying text.

impossible in a 4-by-4-by-4 grid). In this last example, we will make use of the two-color capability of the UCube to suggest a hypothetical game, or genre of game, that could be created with the system. The imagined game in question is a geometric strategy game between two players, "Red" and "Green". At the outset of the game, each player is given four lights of her own color; the two of them are told to place their lights at the eight corners of a cube in the positions shown in the photograph shown in Figure 9. Now, the computer could display the convex hull of the present set of lights (a cube), as shown in Figure 10; and then (in our scenario) the computer tells the Green player to move one of her lights to create the new convex hull shown at the right of Figure 10. Thus, the Green player's job is to change the ”cube” hull to the new hull with one move of one green light. A correct answer to this challenge is shown in the photograph of Figure 11; and if the Green player makes this correct move, the Red player is now given the (current) convex hull and yet another hull that could be created with one move of a red light. In this fashion, the two players take turns moving lights of their own color to produce a new overall configuration of lights at every step, until one player fails to solve the current challenge, at which point the game is over. There are, of course, many variants or extensions of this game that could be imagined (for instance, a player might be asked to shift two lights, or to add a new additional light in her color, to create a new convex hull). The purpose of this example is simply to show that, with the inclusion of two available colors for spatial points in UCube 2.0, a sizeable potential landscape of geometric activities and puzzles becomes feasible.

2.3 PopCAD

Our motivations for creating alternative interfaces to the UCube and SnapCAD stem from the desire to explore this intellectual space more generally; it is far more interesting to discuss a **class** of tangible interfaces for scaffolding digital fabrication than it is to discuss a singular device. To this end, we looked at some of the weaknesses of SnapCAD and towards technologies we had yet to explore. While SnapCAD can admirably perform a number of modeling tasks, it was always envisioned as one device amongst an ‘ecosystem’ of next generation fabrication tools. It has strengths, but obvious weaknesses as well; in particular, the SnapCAD hardware was expensive to produce, and so would be a difficult proposition for some schools or fab labs; it is also rather unwieldy and unportable - it moderately heavy, fairly large, and has many separate pieces that could break or go missing. Thus, an interface with cheaper and more portable materials was desirable.

To address these issues we chose to build a pop-up book combining traditional paper-crafts and paper-friendly electronics such as copper tape. In recent years, revolutionary work has been done in combining electronics and paper crafting[103][88], leading to new techniques and new uses for traditional materials. Paper is inexpensive (especially when compared to circuit boards), light, and easily portable, making it an ideal material choice for a device that would not suffer the same limitations present in the SnapCAD. Although we often think of 'paper' as a rather static material, there are in fact many variations in the size, weight, color, transparency, and composition of contemporary paper products.

We will cover the two paper-based prototypes we created in this vein, dubbed "PopCAD v1" and "PopCAD v2".

2.3.1 PopCAD v1

For the initial prototype, we used a simple construction paper as it provided a balance between strength and flexibility as well as having a consistency well-suited to laser etching and cutting. The pop-up book (named PopCAD) has a 3x3x3 array of 27 points which are evenly spaced 3 inches apart on a 12" x 18" paper surface. The book folds on a single center crease making the closed footprint of the book roughly 12" x 9".

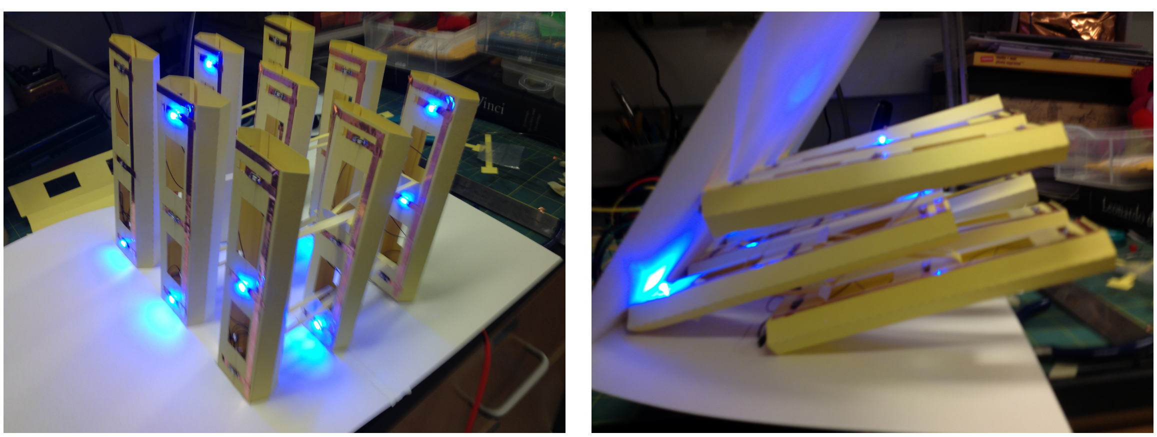


Figure 2.7: Two views of the pop-up book prototype, showing the paper towers and LEDs in both open and closed states.

Each tower has a copper tape circuit consisting of three LEDs on the front face and three corresponding capacitive touch sensors on the left face. The copper tape acts as a paper-friendly conductive material to connect the electronic components together much like traditional wire. The LEDs are soldered onto the copper tape for greater stability. The capacitive sensors are simply a piece of copper tape which is connected to a pin on a microcontroller (in the first version, this is an Arduino Mega Pro). By bringing the internal pull-up resistor connected to the pin "LOW" (to ground) and then timing how long it takes to get back to a "HIGH" state we can tell if the connection is being influenced by a capacitive force. For example, if there is no interference on the circuit, the timer will normally only get to "1" before the resistor is back to a HIGH state; if a finger is placed on the copper tape, the reading will be much higher (typically around "17"). Based on this change, we can detect which switch was touched and toggle the associated LED on or off. The hollow interior of each paper tower is used to solder thin 30-gauge wire to the three LEDs, the three switches, and ground. These seven wires are soldered to a row of headers that stick through the bottom of the first layer of the pop-up book. Wires are then run along the backside of the top layer of paper from these headers to the microcontroller. The entire circuit in then encased in a cloth-covered cardboard binder that acts as a book cover as well as a means to protect and hide the electronics.

2.3.2 PopCAD v2

This is where we talk about the differences: conductive tape, laser cutting, paper choice circuit wiring is more efficient, no horizontal struts...

Although the first PopCAD iteration was a fully-functional prototype, as we approached evaluating the PopCAD in user testing it became apparent that there were several compelling reasons to iterate on the original design. Through a few informal user evaluations as well as our own reflections on the device, we identified several key issues that could be improved upon: (a) the paper engineering design, (b) the structural integrity of the book as a whole, and (c) the lack of "paper-ness" with respect to the circuitry and electrical components of the design.

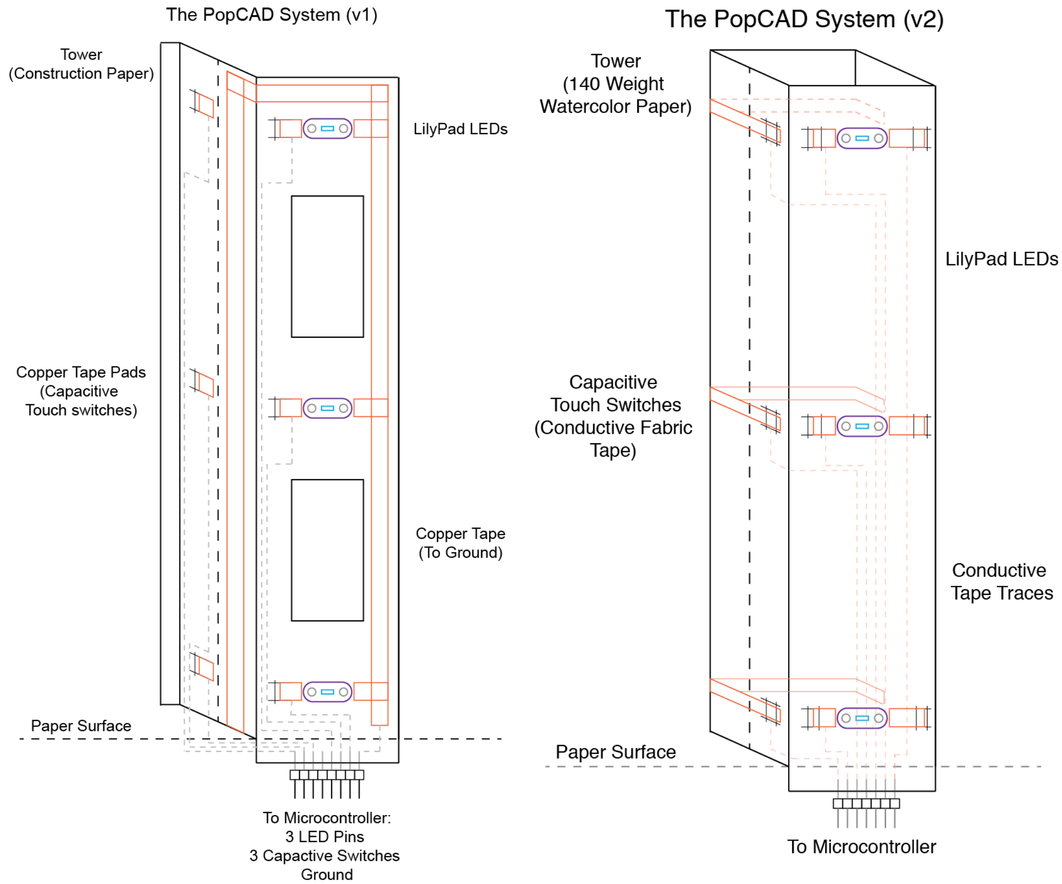


Figure 2.8: The two PopCAD designs side-by-side: PopCAD v1 (left) uses copper tape and 30 gauge wire for the paper circuit, while PopCAD v2 (right) uses fabric-based conductive tape without needing any wires.

2.3.3 Discussion

The software originally written for the UCube and SnapCAD was adapted to work with the pop-up book, making it capable of similar types of algorithmic modeling and stereolithography output for 3D printing. As the grid is 3x3x3, it also makes sense to adapt some of the game-playing aspects of the larger devices (e.g., it would still be possible to play 3D tic-tac-toe). In addition to adding this functionality, there are several improvements and finishing touches to be made on the book itself. Additionally, the current hardware setup for the pop-up book does not allow for the LED's to be snapped on or off, making certain multi-player or multi-shape operations impossible. Weather or not this functionality is crucial to the pop-up book will determine if changes need to

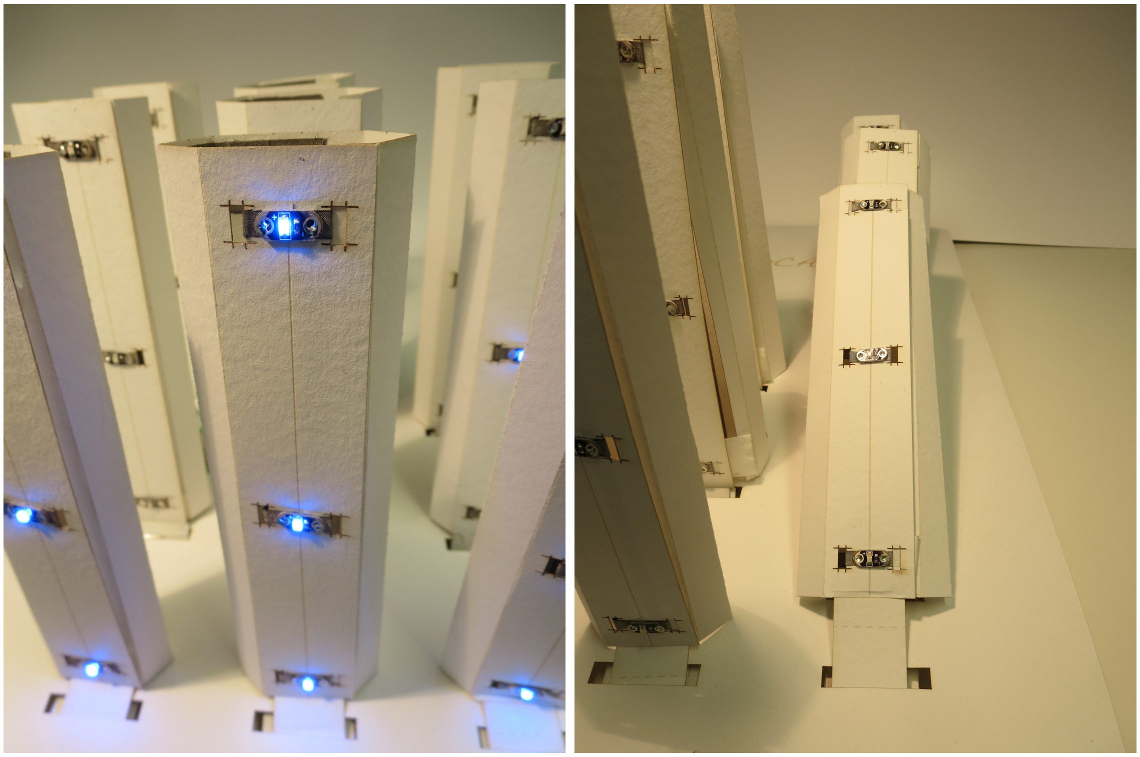


Figure 2.9: Two views of PopCADv2 design: with towers raised and LEDs lit (left), and with the rightmost column of towers laid flat (right).

be made.

Given the different medium of the pop-up book (paper as opposed to circuit boards), it is worth exploring the possibilities afforded by a cheaper, more flexible material. For instance, the flexibility of paper might provide the means for new types of modeling actions. It is plausible to imagine paper tabs or other mechanisms that perturb the LEDs off the integer lattice, or alter the overall topology in such a way that new shapes are possible (e.g. by deforming an equidistant grid into a spherical shape). There may be additional sensors or hardware that could be embedded into the book to provide new functionality (rotation, proximity, pressure). Additionally, due the inexpensive and portable nature of the pop-up book, it is worth exploring the sorts of interactions that could occur between several pop-up books (e.g., extending the input field to include two or more grids, networked interactions like cooperative modeling tasks, or competitive games like 3D-battleship). By using paper as a material to think with, we may find further possibilities as

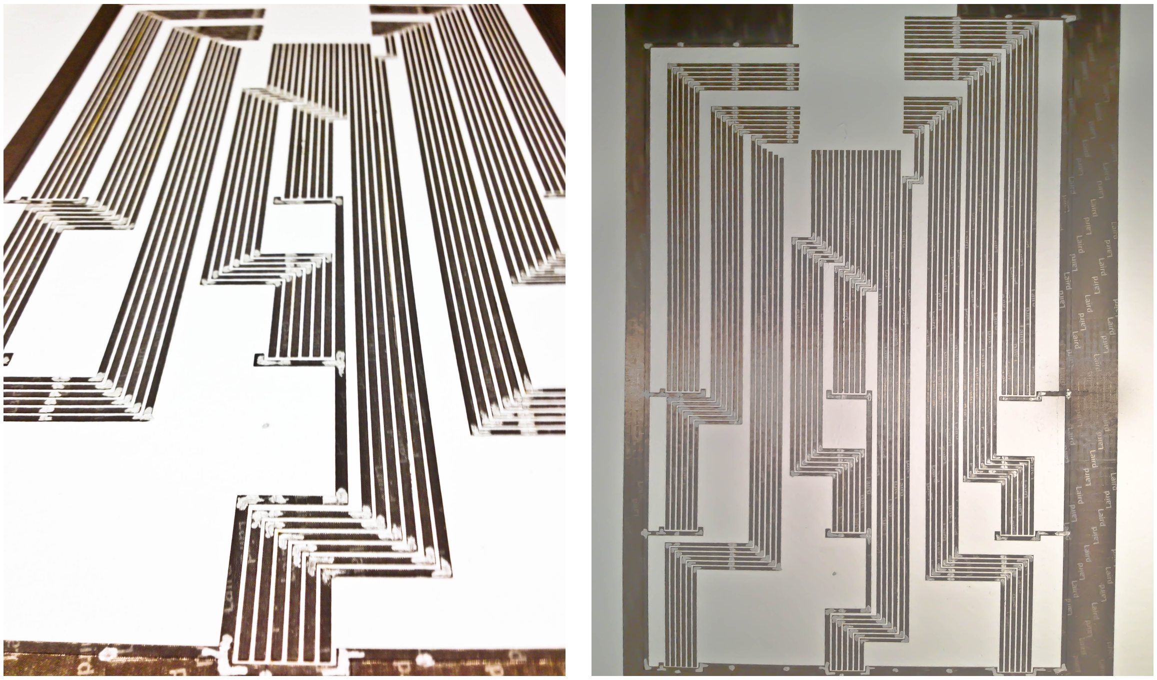


Figure 2.10: Two views of the conductive tape circuit connecting the paper towers to the Arduino Mega microntroller. The circuit was constructed by laser cutting a design through conductive tape (but not through the paper beneath it) and removing the excess material.

development continues.

2.4 Software

Put stuff about software development here. Details. Screenshots.

The UCube Desktop Software The software for the UCube utilizes the Processing framework to read in the active coordinates from the Arduino microcontroller; it then displays those coordinates onscreen as larger red points against a grid of grey dots. This on-screen model can be manipulated in a number of ways. Clicking and dragging along any axis rotates the model, as does the use of the arrow keys on the keyboard. Holding the shift key while performing either action moves the entire model around the screen (essentially re-centering it). The "control" key plus an up or down arrow key zooms in or out along the z-axis. In addition to camera movements, there are a limited number of functions represented by a simple graphical user interface (17 buttons, a slide bar, and two checkboxes) which aid and expand the modeling capabilities of the UCube: there are toggles for turning on and off the convex hull of the active points (either all the points or just the hull of a particular color), viewing the hull as a wireframe or solid object, and a toggle that shows or hides the background grid. In addition, there are several import and export buttons: an export to STL (stereo lithography) format, the standard format for 3D printer files, as well as save and load options which allow users to save and re-load their shapes for use within the UCube system. Figure 2.11 shows the UCube software upon reading in the eight points that were selected in Figure 1: in the bottom panel, the "convex hull" option has been chosen so that a solid cube is displayed (with the input points visible as the highlighted vertices of the cube). Now, by exporting this form (as mentioned above) to STL format, and sending the result to a 3D printer, we can produce a physical model of our specified shape, as shown in Figure 4.

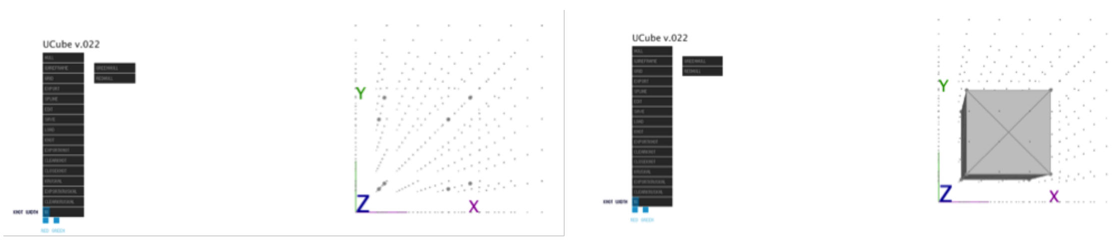


Figure 2.11: Two screen views (left and right) of the UCube software, illustrating the way in which the software reads in the points selected in Figure XXXXXX. At left: the software is displaying the eight selected spatial points as gray dots. At right: the software computes the convex hull of the points and produces the cube designated by the eight selected points in Figure XXXXXX.

Before returning to the software itself, it is worth pointing out that, in effect, the events depicted in Figures 1, 3, and 4 constitute a typical scenario for employing the UCube. The user begins by selecting the vertices of a shape that she wishes to create (Figure 1); checks that shape against its appearance on the computer screen (Figure 3); and, if satisfied, sends that shape on to be output by a 3D printing device (Figure 4). There are, of course, limitations to this scenario - and we will touch upon these in the ensuing discussion. Nonetheless, it is the overall simplicity of the scenario that originally inspired the design of the UCube: the user need not construct a shape on a two-dimensional screen, nor be deeply familiar with the terminology and operations of modeling software. Instead, the creation of a desired shape takes place by moving one's hands in space.

To return to the UCube software: while spare in its features, it does allow the user to perform some operations on the current set of 3D pointsoperations that mitigate some of the inherent limitations suggested by the initial scenario just described. By putting the software in "edit" mode, users can click-and-drag points off the integer lattice, creating shapes that would be impossible with direct manipulation of the UCube towers. There is also a primitive version of a "spline" function, which connects all the active points with a curved spline. (These are new features, added since the earlier version of the system described in [79].) As a guiding heuristic for our design, it should be noted that the UCube software is intentionally minimal. Although there are still some additions to the software that we expect to see implemented in further iterations (as noted in the final section), our aim is not to produce another sophisticated software modeling program. Instead, the software is meant to aid the user in clarifying their physical actions with the UCube towers and switches.

2.4.1 Modeling Modes

The previous section described the architecture and implementation of the UCube 2.0. In this section, we outline a variety of 3D design projects that can be undertaken with the system.

2.4.1.1 Convex Hull: Creating Polyhedral Forms

The most typical type of 3D modeling done with the UCube is to create polyhedral forms, such as the cube of Figures 1, 3, and 4. The basic scenario here is that the user selects a set of locations in space by placing lights at those locations; much as in Figure 1, the lights can be interpreted as the outer vertices of a convex shape. The UCube software can then display and print out the convex hull of the selected

2.4.1.2 Points as “Blocks”: Creating Non-Convex Polyhedral Forms

While a “standard” UCube project interprets the locations of lights as vertices of a polyhedron, the device allows for myriad different semantics for spatial locations. For example, we might wish to interpret the location of a light as signifying the presence, not of a point, but of a cube in space, centered at the given location and with an edge-length of one “hole-interval unit”. By selecting (say) four successive light locations along the length of one tower, then, one could specify a rectangular prism (such as the one shown lying horizontally toward the bottom right of Figure 6). Likewise, by selecting three point locations in an “L” form, one could specify the non-convex polyhedral form seen at the far left of Figure 6. The complex form shown toward the back of Figure 6 represents what happens when we take this idea beyond shapes specified by just a few locations. For those readers interested in recreational mathematics, seven of the shapes shown in Figure 6 will be recognizable as the component pieces of the ”Soma” puzzle; these pieces can be arranged together to form a cube. The UCube could be employed in similar fashion to produce many such dissection-type puzzles.

Figure 7. Two “printed path” forms created by specifying a temporal sequence of points with the UCube. The form on the right is a trefoil knot.

2.4.1.3 Paths: Creating Linear Forms and Knots

In the examples of the previous paragraphs, we have not made use of the fact that the UCube samples selected points in real time: thus, when a user adds or subtracts a point in space, that change is registered immediately in the desktop software. What this means is that the user can exploit not only the overall set of selected points, but can also make use of the order in which those points are selected. A sequence of selected points need not represent only vertices of a solid; it can also represent a path over time in 3D space. Figure 7 shows a sample project based on this idea. Here, the UCube software has been employed to read points as successive positions of a path in 3-space. The resulting paths have been printed out on a 3D printer. In both cases, the path is closed, finishing at the same location where it started; the path printed out at right in Figure 7 is in fact a well-known mathematical form, a trefoil knot. (It may be worth mentioning here that such a knotted form would be rather tricky to create in standard 3D modeling software, but the form can be created “by hand” with the UCube, selecting light positions in space along the path of the knot.)

2.4.1.4 Point Clouds: Creating Minimal Spanning Trees

Instead of interpreting points as vertices of a solid (as in the convex hull examples) or as the successive stations of a temporal path (as in the “knot” example above), we could in fact simply treat our set of points as just what they arenamely, a set of points. Starting with this interpretation, we might produce a form such as a minimal spanning tree of the set of points (a set of edges of minimal total length connecting all the points). Figure XXXXX shows an example of a form created this way: here, the UCube software has computed a minimal spanning tree from a set of six selected points, and the tree is then printed out in solid form.

2.4.2 Other Software Functionality

Edit mode, etc.

Chapter 3

Related Work

The belief that tangible objects[1] play an important role in children's education is relatively recent. Friedrich Froebel's use of 20 wooden forms he dubbed 'gifts' in the first Kindergarten was in 1837[49]. It took until 1907 before an extension of Froebel's ideas and a focus on physical, manipulative objects and tasks was implemented by Maria Montessori in the first Casa Dei Bambini[91]. The interest in children's learning incorporating the use of manipulatives progressed steadily, most notably by Jean Piaget and his work on 'genetic epistemology'. Piaget wrote extensively on the stages of development during which certain kinds of knowledge emerged[65], including logical-mathematical knowledge related to the kind we wish to foster. Although Piaget's specific theories have been strongly challenged[123][107], his influence was extremely important. Seymour Papert, one of Piaget's intellectual descendants, published Mindstorms[98] in 1980 and with it introduced his own ideas about constructivism. Combined with the advent of the physical Logo turtle, Papert brought many constructivist ideas into the modern age and opened the door for a technical and cognitive exploration of how computation and interactive objects could be combined to examine the link between tangibles and children's learning.

While a rich and diverse lineage of tangible and embedded user interfaces has progressed since (and partially because of) Papert, the genealogy of the proposed work derives from an interest not only in constructivist-like activities, but in theories about how interaction with physical objects

[1] It is worth noting the difference in this work between 'tangible objects' of the sort that a child might play with (e.g. Lego) and 'tangible user interfaces' (TUIs) that a child might interact with - typically a peripheral device (apart from the keyboard and mouse) that communicates physical interactions to a computer.

may be beneficial to learning. In cognitive science, the area of embodied cognition examines the ways in which our interactions with the physical world shape our cognitive experiences from a body-centric point of view. More specifically, embodied cognition holds that our cognitive processes are 'deeply rooted in the body's interactions with the world'[132]. This is in stark contrast to decades of research in cognitive science wherein the mind was viewed as a sort of central but detached information processing unit where motor-sensory functions were more-or-less secondary inputs and outputs to a main system[32]. Although there are several different tenets of this body-centric view, the primary conclusion relevant to our proposal is that interactions with physical objects can shape, clarify, and reinforce our cognitive processes in scores of disparate areas. For example, Goldin-Meadow shows that through an analysis of hand gestures, one is able to predict a subject's 'readiness' to learn[52]; that is, the gestures they make while explaining a concept are literal clues as to the state of their cognitive processes. Of keen interest for this work in particular is a domain referred to as embodied mathematics. Lakoff and Nuñez[78] give a fascinating account of the origins of mathematics from an embodied point of view. They propose that humans, by virtue of their interactions with the physical world, inevitably form certain intuitions of a mathematical nature. Recognizing small numbers of objects (e.g. the pre-verbal ability to do arithmetic with less than five objects), estimation, and simple comparisons are a few of the examples given in[78]. From these basics, they argue that four kinds of physical operations (object collection, object construction, using a measuring stick, and movement along a path) form the basis of simple arithmetic. Although the book postulates about concepts as ungrounded and seemingly abstract as infinity, for our work it is enough to suggest that the interactions present in our designs follow from these four operations and may in fact contribute to the solidification of more complex mathematical ideas in 3D modeling and digital fabrication (e.g. forming correct mental models of 3-dimensional objects).

In their section on 'Thinking Through Doing', Klemmer et al.[75] give a particularly poignant summary of why we ought to consider the body as instrumental in any human-computer interaction design, stepping through many of the concepts outlined above. In fact, the marriage of ideas derived from Papert's work with the conclusions of embodied cognition are not new, and appear to substantiate our motivations to produce tangible, manipulative interfaces as opposed to purely 2-dimensional screen-based work. In the mid-to-late 1990's, research examining the ways in which physical objects might be infused with computational ability started to coalesce around several themes[40]. Resnick's work with 'digital manipulatives'[109][136] specifically references the contributions of Froebel and Montessori in the design of a series of 'programmable bricks' with computational ability whose aim is to make certain specific concepts (e.g. systems-level thinking) more salient for the user. Ishii's work on breaking down the divide between physical and virtual worlds into 'tangible bits'[68][67] has subsequently set the stage for a new family of tangible interface designs that support the kind of embodied interactions that our work seeks to produce. By constructing environments and artifacts that focus on the possible physical representations of computational components, these works (among others) created the philosophical space to delve into how tangible objects might affect users at a cognitive level. Our proposal is a confluence of both tangible and cognitive design; as Resnick states, 'We are interested in Things That Think only if they also serve as Things To Think With'[109].

Of particular interest for the current work are explorations focusing on 3D modeling and perception with tangible interfaces. Prime examples include software that allows for 3D shapes to be flattened into paper-printable, origami-esque polyhedra[41], a construction kit with kinetic memory so as to record and playback certain user-generated manipulations[106], as well as several variations of 'smart-cube' interfaces [127][115] that encourage spatial and logical reasoning in order to make use of the computational aspects of the cubes. While diverse in their implementation, these kits point to ways in which interface design can tease out the kind of 3-dimensional problem-solving and exploration present in the proposed work.

Related contributions focus more on the cognitive processes involved when exploring embodied interfaces with children. Research on supporting creative problem solving with children[21], arguing for a kindergarten-influenced approach to creative thinking[108], embodied approaches to analyzing children's interactions with smart objects[12], as well as the embodied design of interfaces for introducing mathematical concepts to kids[8] have shown a great degree of correlation between

Figure 3.1: Left: The ActiveCube system. Right: The Roblocks system.

physical interaction and learning in children.

Yet so far, there have been few attempts to design embodied interfaces for children that specifically address the growing presence and availability of digital fabrication tools. KidCAD[47], a deformable pad that captures the 2.5D geometry of depressions made on the underside of the surface, was a very promising idea in that it allowed very young children to take small objects from their surroundings (or their hands) and 'stamp' them into the pad - an intuitive and satisfying experience. Unfortunately, the authors intentions to be able to output the geometry to 3D printers has not yet manifested. Easigami[62] is a set of interchangeable and interlocking polyhedral faces with smart 'hinges' that can reproduce the morphology of a set of connected faces while connected to a computer. In contrast, Easigami **is** able to export this morphology to a stereolithography file ready for 3D printing. There are several other interfaces that deal with 'interactive fabrication'[131]; devices that manipulate materials interactively based on various input from a user, such as controlling a laser cutter with a laser pointer (instead of through a CAD program)[94], or a wearable device that takes in a CAD file and provides haptic feedback to make the physical creation of the device by hand easier, even for a non-fabricator[135]. These projects, as well as several others that deal specifically with digital fabrication for laser cutting[70][130], are examples of the subset of tangible interfaces to which this work belongs - namely, those concerned with providing a means to engage with digital fabrication technologies in a more intuitive, embodied fashion. However, with the exception of KidCAD and Easigami these designs are not made with children in mind, nor

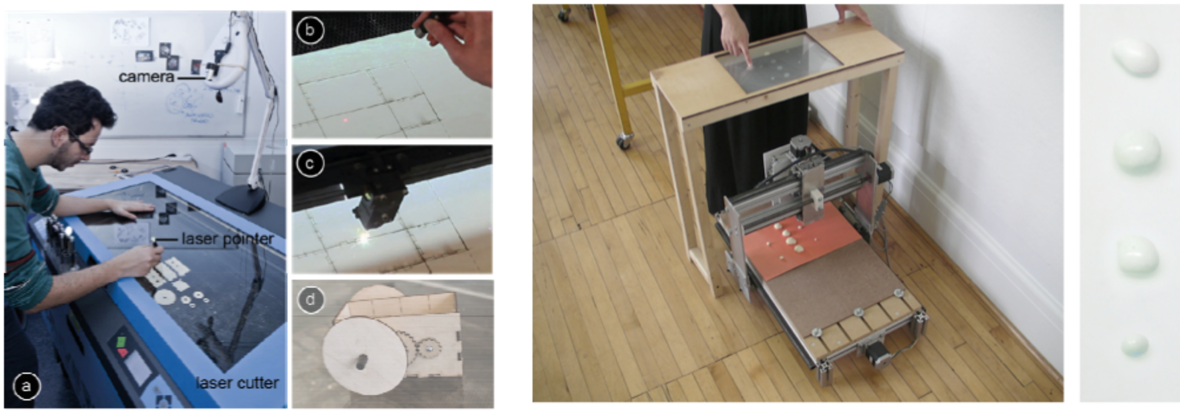


Figure 3.2: Examples of interactive fabrication interfaces: Constructable (left) allows users to control a laser cutter with a set of physical tools as opposed to a pre-defined design file. Shaper (right), and interactive fabrication tool using expanding polyurethane foam.

do they cover the range of possibilities for child-friendly input devices that focus on 3D-printing. Thus we argue that there is room for exploration in this area, as well as a lineage that suggests meaningful results may follow from the incorporation of tangible interfaces with embodied design.

There are several strands of research that have strongly influenced the design (and motivation) for the UCube. Perhaps the most fundamental of these is in the area of "embodied mathematics" that is, the notion that mathematical thinking and learning are affected by, and perhaps grounded in, metaphors derived from bodily experience. The most thorough and discursive (though largely theoretical) discussion of these ideas is in the foundational text by Lakoff and Nuez [78]: the authors discuss physically-derived metaphors that underlie such essential mathematical ideas as numbers, operations, and sets. Such notions of embodied mathematics haveeven before the Lakoff/Nuez textplayed a role in discussions of the development or instruction of mathematical ideas. The link between physical experience and mathematical growth was a strong element, for instance, in Montessori's work (see, e.g., [56]); much of the motivation behind traditional mathematical "manipulatives" such as number rods and balance beams can also be traced to this intellectual tradition. More recently, theoretical discussions of embodied cognition have given rise to fine-grained observations of the connections between bodily activity and mathematical learning: Goldin-Meadow[53],

Figure 3.3: Left: The KidCAD interface showing a model Zebra and its 2.5D impression on screen. Right: The Easigami system, showing a series of connected polygonal faces with smart-hinges and embedded electronics.

for instance, describes a fascinating line of research in which children's nonverbal gestures appear to both reflect and, in some cases, anticipate their verbal understanding of concepts such as conservation and "inverse operations". Pedagogical research in embodied mathematics has, moreover, proceeded hand-in-hand with the development of desktop, embedded, or portable technological artifacts to support the link between bodily actions and mathematical conceptualization. Papert's discussions of the Logo computer language [98] reveal this connection early in the history of children's computing: Papert discussed, for example, the way in which the program for a Logo circle resonated with children's bodily understanding of moving in a circular path. More recently, Nemirovsky et al. [95] describe the use of a computer-based motion detector system to assist children in the development of intuitions behind graphing; Howison et al. [60] used a device based on a Wii remote to assess children's understanding of ratio (the children attempt to move their arms in a manner illustrating a target ratio); Bakker et al. [18] created a collection of handheld objects ("MoSo Tangibles") with embedded sensors to help children learn about musical ideas via hand motions such as waving, squeezing (pressing hands together), and shaking up and down, among others; Mickelson and Ju [90] use sophisticated video and projection equipment as the basis of activities through which children can learn about mathematical ideas (e.g., symmetry, rotation angles) via large- scale physical movements. The development of the UCube follows within this tradition, in that the device was created to enable children to specify and identify three-dimensional shapes by hand motions (instead of, by contrast, using symbolic commands directed at a two-dimensional screen display). At the same time, the UCube is not simply a device for mathematical instruction, but is more generally a tool for mathematical design. As noted at the outset of this paper, the intent of the UCube is to enable youngsters not only to learn about but also to build mathematical shapes. Specifically, we see the device as part of a larger, burgeoning "technological ecosystem" around the activity of three- dimensional printing. The first section of this paper noted several prominent researchers who argue for the democratization of this technology, and for its applications to education. Indeed, exciting early work has been done in applying 3D printing to education in fields such as architecture [23], solid geometry [58], and mechanical design [83]. The UCube is designed so that it can be employed by younger studentsyounger, for instance, than the typical (undergraduate-age) architecture student. At the same time, we see no reason at all why the device could not be used by adult or professional-level studentsparticularly if (as we anticipate) the device and software are made more expressive or powerful in future iterations. It should also be noted, along these lines, that our early pilot test experience suggests a potentially fruitful use for the UCube as an assessment device for children's spatial cognition. (The young subject who suggested that it could be made into a "puzzle game" is anticipating our thoughts here!) A researcher could, for instance, give children a pattern of lights and ask them to match that pattern to one of a set of physical or pictorial solid representations; or one might ask children to recreate a variety of physical solids (such as a plastic prism or tetrahedron) by selecting the appropriate set of lights, and note their development and difficulties in doing so. By using the UCube as an experimental device in this fashion, one can position this work as part of a tradition (dating back at least to Piaget [100]) in understanding spatial thinking and its development (cf. also [96] for a more recent treatment of the subject).

Having shown several representative prototypes of our own work in paper mechatronics, it is now worth situating that work in relation to that of other researchers in this (still embryonic) field.

The blending of traditional papercrafts with emerging technology is in fact still a relatively novel technique, but there is a remarkable community of researchers beginning to explore this area. For us, a special debt is owed to Leah Buechley's High-Low Technology group at the MIT Media Lab; that group first (to our knowledge) introduced conductive ink and copper tape into paper-based projects. Early (c. 2008) use of conductive ink with microcontrollers on a paper substrate can be found in [29] and [39] with the development of paper-based Arduino processors and simple electronic components (e.g. LEDs, toy motors, switches) that could be placed onto conductive paint to form an electronic connection. This work culminated with a paper application usually reserved for home remodeling: a "living wallpaper"[30] where passers-by could trigger light, movement, and sound by interacting with different parts of the surface.

These early efforts in turn spawned developments that further refined the expressive potential of paper-based electronics, infusing traditional papercrafts with new elements and abilities. An electronic pop-up book by Qi and Buechley[103] reimagined the traditional pop-up by infusing each page with paper-friendly, interactive circuitry (e.g. by using a copper tape circuit to power LEDs in a pop-up cityscape). Other projects in this vein include techniques to animate origami structures through shape-memory alloy (SMA)[104], using SMAs in the design and fabrication of printable paper-based devices (e.g. speakers and lamps)[113], storytelling and craft-making through electronically-enhanced storybooks and workshops [69][31][124] and the use of small microcontrollers incorporated into programmable paper-based scupltures[88].

These efforts have focused on the creation of compellingc(either electronically or digitally enhanced) papercrafts. As noted in the introduction, there are numerous technological developments that, in combination, serve to accelerate the development of paper mechatronics. For instance, Mueller et al.[94] describe the use of a laser cutter to produce origami figures; Kawahara et al.[74] describe how inkjet-ready conductive ink can allow circuits to be printed easily and directly onto paper; and Koizumi et al.[76] present a toolkit for wireless control of moveable paper toys, Zhu et al.[134] describe a method for wireless power transfer for paper computing, and Coelho et al.[33] have acheived the direct embedding of conductive components during the papermaking process.

These last efforts are powerful examples of expanding techniques–they signal the emergence of a new territory within which to explore paper-based electronics. Our own prototypes are intended to continue this communal development of techniques and examples, but there are several factors that distinguish our work from that of other efforts. First, several of our own prototypes (e.g., PopCAD, the bicycle rider, and the cherry blossom painting) may be seen as incorporating paper elements as portions of larger, composite systems. PopCAD is a paper-based input device; one might think of it as one early foothold in an unexplored landscape of paper input devices for children's activities. The bicycle rider is an artifact that combines a desktop computer screen with a paper model; again, one could think of it as an exemplar of blending papercrafts with (e.g.) high-resolution graphics, or one could imagine websites designed to work as "background graphics" for electronically-controlled paper constructions. The cherry blossom painting is (in a sense) the "flip side" of PopCAD; whereas PopCAD is a paper-based input device, the cherry blossom painting is a paper-based display for output. And once more, one could take the example still further: paper mechanisms or models could be moved or controlled as components of extended output displays that combine physical and screen-based elements.

More generally, we see our prototypes as (still-early) pointers toward a new genre of activities for children. In the final section, we turn our attention to the mechatronic future of children's papercrafts.

Chapter 4

Evaluation

This section is devoted to the description and discussion of three separate user studies with the devices discussed in Chapter 2. Two studies were performed with the original UCube device (one more informal than the other), while a longer, more detailed study involved both the SnapCAD and PopCAD systems.

4.1 UCube Pilot

4.1.1 Procedure

Early in 2011, we conducted an initial (and informal) pilot test of the UCube with a group of 12-14 year olds. Fourteen participants, consisting of five girls and nine boys, were divided into six groups (five groups of two, one group of four). Participants were asked to model a sequence of five shapes of increasing complexity using the UCube along with the companion software. The target shapes were displayed on one half of a computer screen, while the UCube software showing the live model was displayed on the other half as in 4.1. The first shape that participants were asked to model was a straight vertical line; after this, the requested shapes were a diagonal line, a cube, a triangular prism, and finally an irregular polyhedral object. No shape required more than four towers to complete, and shapes were always presented in the same order.

Participants were instructed to place the poles on the board (but not shown how), and were told that the software model could be rotated and filled in using the keyboard and mouse, should that help them complete the task. The participants were not given any hints as to how to complete

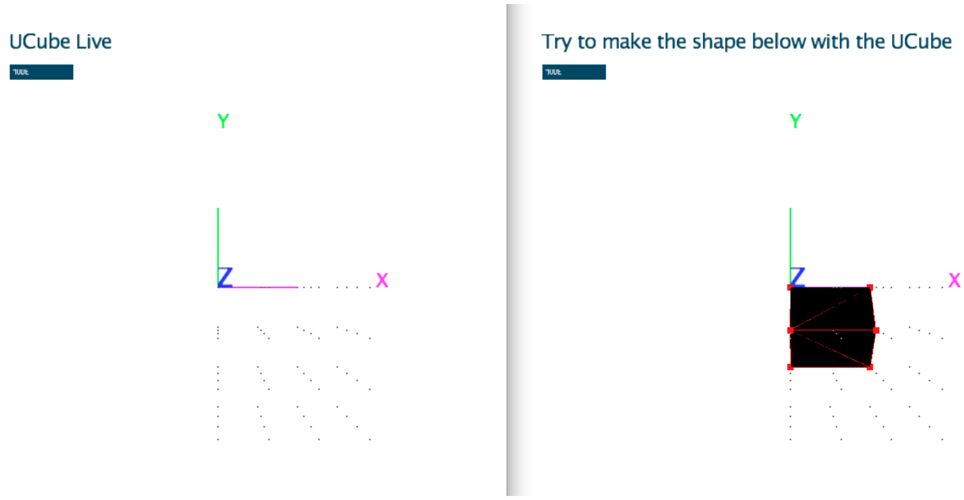


Figure 4.1: A screenshot of the testing setup, with the live output from the UCube on the right and the target shape on the left.

the shapes and were not told when they had the correct configuration (they had to indicate their belief that the model was done). Participants were also instructed to 'think aloud' about their actions. The main purpose of the pilot study was to get an initial impression of how the UCube would act as an accessible 3D modeling toolhow well it could help "3D novices" overcome the "2D bottleneck".

4.1.2 Results and Discussion

Of the six groups who participated, four groups successfully modeled all five shapes, one group ran out of time after three shapes, and one group finished one shape. Sessions lasted between 17 and 30 minutes. A variety of problem-solving strategies were observed during testing, as the participants tended to treat the exercise as a sort of puzzle to be solved. Simple methods equivalent to "try and see" were common, and seemed to serve as a base point from which to draw conclusions about the relationship between the 3D model and 2D on-screen representation (e.g. "No, not there, up one"). More sophisticated strategies were also observed "deconstructing" more complex shapes into smaller, easier-to- model shapes (e.g. thinking of one side of a cube as a square) was observed from several groups. Another popular technique was to systematically match the on-screen perspective from the live model with the shape they were attempting to model (e.g. "Okay, first let's do the top view, and then go from the side"). By orienting the two models similarly, participants were able to make more accurate modeling decisions as well as check their model against the on-screen shape. Counting distance in terms of spaces on the board, between switches, or between dots on the screen was also a very common technique of reasoning about and describing position. For example, by counting that two vertices of a shape were separated by "two dots over and one down" on the screen, subjects were able to count the distance out on the physical UCube board. A few of the more mathematically-advanced participants used terms such as "axis" and "origin" to orient themselves and describe various positions on the board to their partners. Another revealing observation in the pilot study was that, in the few instances of mechanical failure (certain switches not lighting up, towers not plugging in properly, or points not showing up on screen) the participants were still able (with a high degree of certainty) to complete the assigned tasks. This appears to indicate that, as opposed to arbitrarily moving the towers around until the two sides of the computer screen looked the same, participants had formed a more substantial mental model of the relationship between the UCube interface and the 2D representations on the screen. That opens the possibility that by performing the embodied interactions necessary to operate the UCube, participants had actually strengthened their understanding of how 3-dimensional space is typically represented on a 2D screen. Although further testing and observation is needed, this finding would strengthen the argument for using the UCube in an educational setting to improve understanding of 3D space, as well as providing a gateway for youngsters to move on to more complex modeling software. While the variety of problem-solving techniques we witnessed is a testament to the participants' ingenuity, it is also indicative of the fact that parts of the UCube are not immediately intuitive. While none of the participants had trouble understanding how to place the towers on the platform, the positions of the towers and switches had to be reasoned out explicitly. It was common for groups to clear the board of any poles when starting a new shape, even in cases where an overlap of points or tower positions existed. (Figure XXXX, for exampleshown earlier in the context of explaining the UCube's operationdepicts one of the students placing a tower and checking the screen to see whether the tower placement is appropriate.) Although most groups completed all the shapes (or ran out of time), there were some expressions along the way of the difficulty of the task (e.g. "This is hard", or "This is like a puzzle"). This indicates that design changes can be made in future iterations to help clarify the correspondence between positions on the UCube platform and the on-screen representation; for example, labeling the both the physical and software grid with a simple alphanumeric system. Despite these drawbacks as well as the inherent limitations of the UCube design, these early results indicate a promising ability of youngsters to effectively engage with the UCube interface. In fact, despite various levels of success in completing the assigned tasks, the vast majority of participants exhibited a high level of engagement with the UCube. For example, although the group that completed only one shape seemed unmotivated to attempt to model the other shapes, they continued to play with the interface and observe the results, even stating "this is fun" and "I like the switches". Participants also saw potential uses for the UCube outside of the specific exercise we assigned. Comments (unsolicited) included, "you should use this to teach geometry" and "you could make this a puzzle game". At the very least, these early results indicate that the majority of participants were able to take a 2-dimensional representation on the screen and model its 3-dimensional equivalent using the UCube, a very encouraging result in our eyes.

4.2 Further UCube Study

Early in 2012, we conducted a further user study of the UCube with a group of 11-13 year olds. The group consisted of ten participants, eight boys and two girls, from a local middle school multimedia class. Every participant was individually led through two separate exercises (outlined below) using the UCube.

4.2.1 Procedure: Modeling

Participants were handed a 3D-printed shape (modeled and printed from the UCube) and were instructed to attempt to model the shape using the UCube. The participant was initially allowed to hold the shape for approximately 10 seconds, after which they would hand the shape back to the facilitator and attempt to model the shape from memory. Participants were instructed that they may ask to hold the shape again, at which point they were allowed to hold it throughout the duration of the modeling task. Additionally, users were instructed that they had the option to skip a shape and return to it at a later point in the exercise. The five physical shapes presented were: a cube, a tetrahedron, a diamond, a “house” (a cube with a pyramid on top), and a complex irregular polyhedron. The models were presented to the user starting with the cube (as this was deemed to be the most basic shape with regard to modeling complexity). To avoid an ordering bias, we randomized the presentation sequence of the next four shapes using an online random order generator. If, after skipping a shape and returning to it, the participant was still having difficulty, we offered them the opportunity to attempt modeling the shape with the help of the UCube software, the effects of which are discussed in the results section. Participants were given a total of 25 minutes for the modeling exercise. We recorded, but did not limit the modeling time per shape, only the total time for all five shapes.

4.2.2 Procedure: Matching

Participants were instructed to face away from the UCube while the facilitator modeled a set of lights on the UCube corresponding to one shape among a set of physical models laid out on the table next to the UCube. Once the lights on the UCube were set up, the participant was instructed to turn around, and indicate which physical object they thought the set of lights on the UCube corresponded to. There were nine physical models presented on the table, and consisted of a cube, a tetrahedron, the house shape, a diamond, a triangular prism, an elongated hexagon, a parallelogram, a trapezoid, and an irregular polyhedron (see 4.2 for a picture of all the models). The shapes were always presented on the table in the same order and orientation to avoid discrepancies in perception or association. Of the nine shapes, the participants were asked to match five of them (the cube, the triangular prism, the parallelogram, the elongated hexagon, and the trapezoid). Thus, only the cube was presented in both the matching and modeling exercises. As with the modeling exercise, the cube was presented first, with the remaining four shapes presented in a computer-generated randomized order.Participants were given a total of ten minutes for the matching exercise, corresponding to two minutes per shape, and were instructed to think aloud during the process.

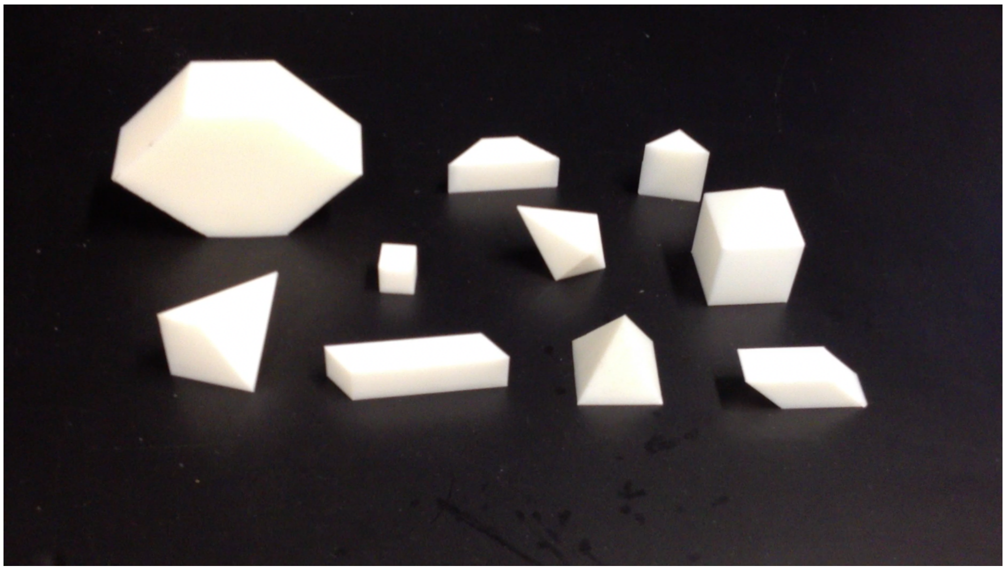


Figure 4.2: The nine models used during the user study: diamond, trapezoid, parallelogram, cube, elongated hexagon, irregular polyhedron, triangular prism, tetrahedron, house.

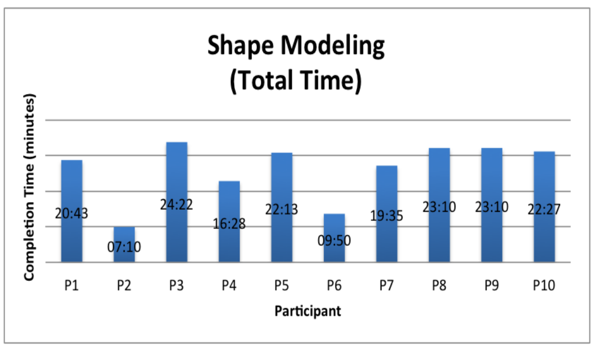
4.2.3 Results

While many established forms of 3D modeling systems can be confounding and operationally too complex for a child to navigate, the UCube was positively received and system instruction was accomplished with just a minor introduction and demonstration (system instruction and demonstration lasted approximately 2-3 minutes). We found this first instance of system comprehension to offer some validation that the UCube worked well as a user-friendly 3D modeling device. This section will detail the outcome of both the modeling and matching tasks performed.

4.2.3.1 Exercise 1: Modeling

Modeling occurred under three conditions:recreate the object from memory, construction of the object while it was in the participants possession, and modeling the shape with the help of the

UCube software. Overall, 21 of 50 shapes were completed from memory, 12 of 50 were completed while holding the shape, and a further 8 of 50 were completed with the aid of the UCube software, for a total of 41 out of 50 shapes modeled successfully (82%). Of the nine missed shapes, seven were of the same shape, the complex polyhedron. The remaining two misses were from the same participant, who ran out of time before completion. Of the 10 participants, 8 were able to recreate the cube from memory, whereas only 4 were able to recreate the diamond and the tetrahedron from memory. Half of the participants constructed the house from memory, and no participants were able to complete the irregular polyhedron from memory. However, once shown the software the majority of the participants found the modeling task significantly easier to perform. The irregular polyhedron was by far the hardest shape and was only able to be completed by 3 of the 10 participants either after continued possession of the shape or using the software.



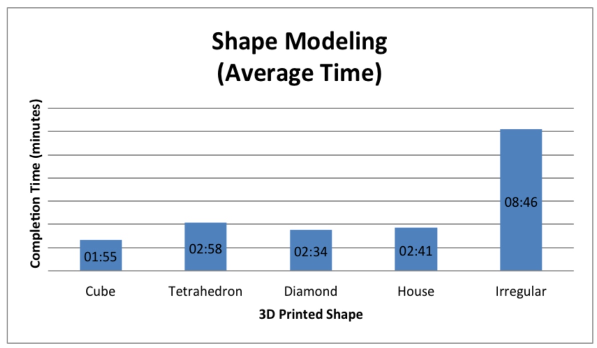


Figure 4.3: Results of the modeling task, showing total modeling time spent per participant (left) and average modeling time spent per shape across participants (right).

4.3 represents the total completion times per participant (on the left) and average time per shape (right). Two exceptional completion times were observed, where participants finished modeling all the shapes in under 10 minutes. However, the majority of participants finished the task in the 19-25 minute range. Only one of the participants ran out of time. Once participants had been introduced to the software, 9 of 10 of participants were able to complete all but the irregular polyhedron. It is interesting to note that of the 10 participants, the child that had the most difficult time modeling, the lowest shape completion rate, and the longest completion time during the matching exercise was the youngest participant.

4.2.3.2 Exercise 2: Matching

Out of 50 matching tasks (five per participant), all but three tasks were completed in 20 seconds or less. 4.4 displays the total time spent on the matching task per participant (left) and the average completion times for each shape (right). No participant selected the wrong shape (a few preliminary "mis-selections" were made that the participants quickly corrected), and all participants completed the task in well under the allotted 10 minutes. The lack of errors in the matching task is highly encouraging as a basis from which to reason about youngsters' abilities to perceive and reason about convex hulls as a set of lit vertices in space, meaning that this kind of 3D modeling interface might be applied to other domains (e.g., as a cognitive assessment tool, a puzzle game, etc.) with some optimism.

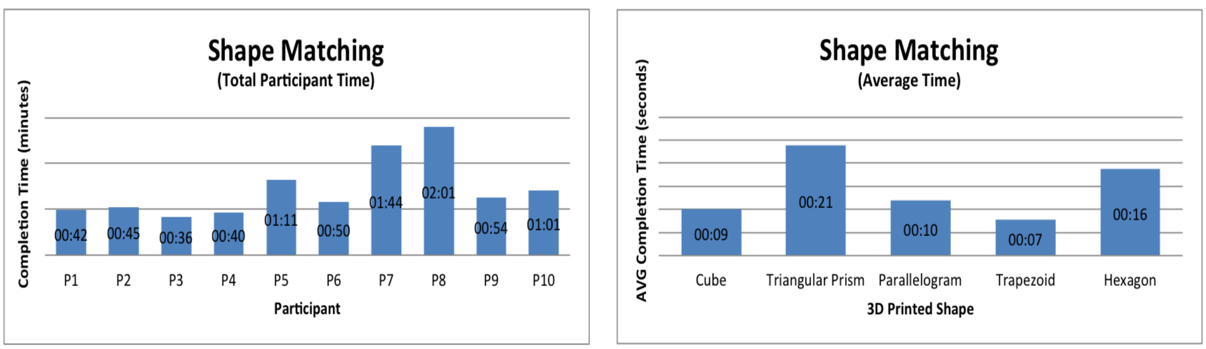


Figure 4.4: Results of the matching task, showing total time spent per participant (left) and average time spent per shape across participants (right).

4.2.3.3 Observations

Modeling trends as well as distinct modeling behaviors were documented in the process. Common observations included building from the ground up (lowest vertices first), building in the orientation that the object had been presented in, not clearing the poles/lights from the UCube before starting to model a new shape, and modeling a shape by breaking it up into discrete parts (e.g. a participant building a house would commonly build a cube first and then add on a vertex to the top; a participant constructing the diamond might combine two opposite facing triangles.).

Unique behaviors were exhibited in the modeling process as well, reflecting a type of user-specific construction-based problem- solving. One participant used their arm to connect the red lights of the UCube for shape definition. A few participants oriented the object differently than how it had been presentedtypically this occurred for the modeling of those objects with a pyramidal apex (tetrahedron, house, diamond). Apex formation was perhaps one of the most difficult concepts for most participants to grasp, as it required them to strategically align the base on a 3x3 grid so there was a middle plug for them to create the apex. If participants were fixated on designing from a 4x4 grid then there was no center plug for them to create a midpoint. Some participants ended up building an oblong polyhedron as opposed to a cube, or an oblique polyhedron as opposed to an equilateral tetrahedron. Other observed behaviors included a participant who modeled shapes by turning on lights for an entire shape edge, as opposed to just the corners and a participant who built shapes that were floating, as opposed to resting on the base of the UCube. There were also some notable behaviors regarding physical and gestural actions of the participants. Many participants modeled with both hands simultaneously, placing towers and flipping switches without a clear preference for a dominant hand. Participants would often gesture with their arms following an arc in parallel with a face of the object they were currently modeling. This "tracing" behavior was also noticed when participants were holding a physical model and tracing a side of the object with their fingertip, often while rotating the object with the other hand. Finally, during object possession phase three participants actually placed the 3D object on top of the UCube in the modeling space while they reasoned out the construction (see 4.5 for an example).

4.3 SnapCAD and PopCAD

Starting in early 2014 we conducted a study using both the SnapCAD and PopCAD devices with a group of 11-18 year olds at a local drop-in enrichment program focuses on children from under-served and low socioeconomic communities. Twenty participants enrolled in the study, con-

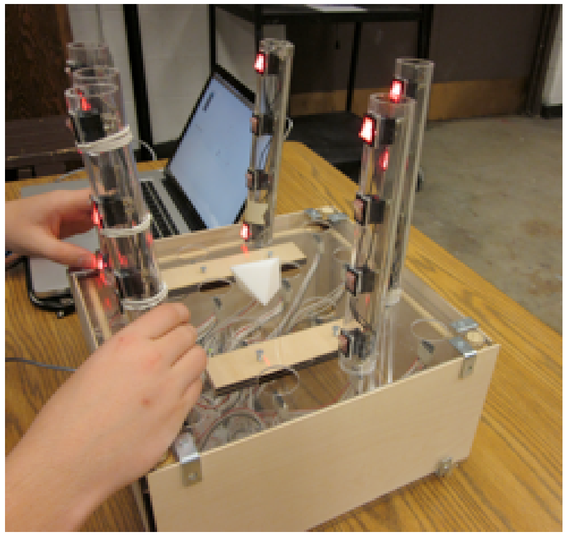


Figure 4.5: A participant modeling with the UCube, using a strategy of placing the physical model on top of the UCube while modeling, as well as using both hands simultaneously to manipulate the towers.

sisting of 12 boys and 8 girls (no one responded with other, although it was an option). We collected some basic demographic information, including age, race, grade level, 3D modeling experience, 3D printing experience, computer ownership and use, interest in engineering, and how difficult they thought classes in school were.

To present these findings, then: the participants were primarily of Latino/Hispanic descent, but also included those of African-American, American-Indian, Asian, and Caucasian descent. Grade levels ranged from 6th-12th, with an overall average of 7.9 (8.33 for boys, 7.75 for the girls). Average age was 14 years, 1 month, 20 days (14 years, 6 months for boys, 13 years, 7 months for girls). 16 of 20 participants had a computer at home. Describing their comfort level using a computer on a scale from 1 to 10 (10 being most comfortable), the participants averaged 7.9 (8 for boys, 7.75 for girls), with no scores below a 5. Of the participants who had a computer at home, two reported using it only a few times a year, five used it a few times per month, four used it a few times per week, and five reported using the computer everyday. Only three of the participants had any experience with 3D modeling software. Interestingly, only two of the participants had never heard of 3D printing before enrolling in the study, but none of them had ever designed or printed anything using a 3D printer. When asked about their interest in engineering, only seven children (all boys) stated they were definitely interested. However, only two kids (both girls) stated that they were definitely not. The rest (11 kids) stated that they were either 'maybe' interested, or 'not sure'. When asked how difficult they felt school classes were, six responded 'easy for me', 10 said 'somewhat easy for me', and four responded 'somewhat hard for me' (no one responded 'hard for me').

The study ran for seven weeks and comprised several stages, the first being a pre-assessment of spatial reasoning skills. The spatial reasoning assessment was done using the "Children's Mental Transformation Task" developed by Susan Levine ([35] p.1260-1261). In the task, participants are shown two pieces of paper, side-by-side. One piece shows a geometric shape, split apart and rotated in one of several different ways. All shapes were symmetrical either horizontally or vertically, and thus split along either a vertical or horizontal line of symmetry. Shapes were translated in one of four different ways: (a) translated perpendicular to the line of symmetry (direct translation), (b) translated and then moved diagonally apart (diagonal translation), (c) rotated 45 degrees outward from the line of symmetry (direct rotation), or (d) rotated and then moved diagonally apart (diagonal rotation). The other piece of paper contained the geometric shape, recombined correctly, along with three incorrect choices. In the study we conducted, participants were given two sets of 10 shapes, one set as a pre-assessment before doing any modeling, and another (completely different) set after completing the entire study, as a post-assessment. Figure 4.6 shows an example instrument, with the four possible translations.

After the pre-assessment, participants were split into two groups of 10 students each - the selection alternated evenly based solely on order of participation - with group A modeling first on the PopCAD and group B modeling first on the SnapCAD (as described in Chapter 2). Each session proceeds roughly as follows: a brief introduction to the device, during which the participant is only told how to operate the device, but not what any of the buttons do, and given free time to become comfortable with the interface. Participants were encouraged to explore both the interface, and the buttons in the software that control the three primary modeling modes (convex hull, path,

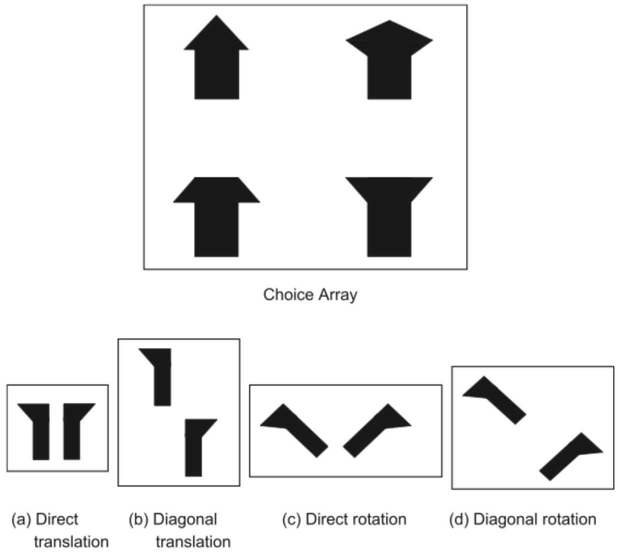


Figure 4.6: An example problem from the spatial reasoning exercise. The figure at the top shows the choice array of four shapes, where the lower left figure is the correct option. Examples (a) through (d) show the four different types of translations found in the exercises - direct translation, diagonal translation, direct rotation, and diagonal rotation.

minimal spanning tree).

Once the subject indicates that they are ready to move on, we move into a series of three modeling exercises that explore each of the aforementioned modes. The basic operation and a brief explanation of each mode were given to the participants as an introduction to each mode. Four 3D-printed models representative of each mode were presented to the user in an order judged to be from least complex to most complex (and thus was the same for each user), for a total of 12 modeling tasks across the three modes. 24 models were used - one set of 12 was used across every user's first session (independent of device), with a remaining 12 models used in every user's second session. Figure XXXXX shows the two sets of models side-by-side.

The tasks that follow are the same for each device:

Task 1: Convex Hull Modeling

The participant will be given a brief demo of how the 'convex hull' modeling mode interprets the points from the device. The user will then be presented with a series of four (4) plastic, 3D-

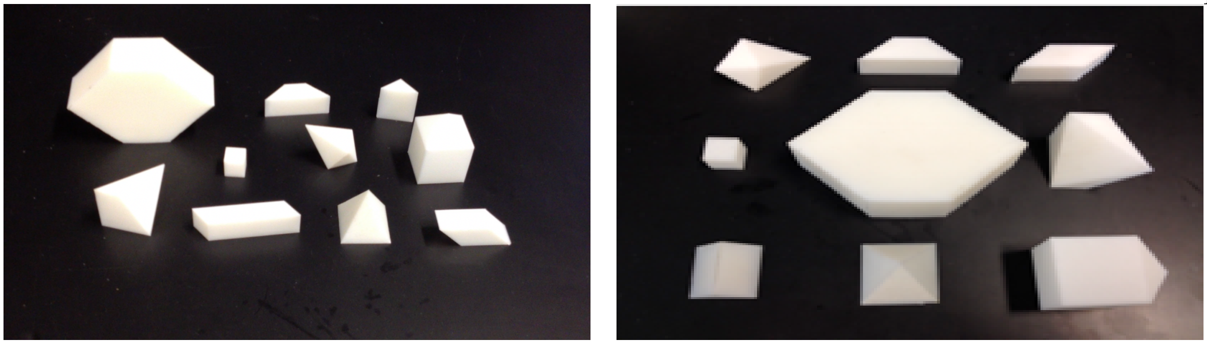


Figure 4.7: The two groups of 12 3D printed models used in the first session (left) and second session (right).

printed models that were modeled on the device using convex hull mode. For each of these shapes, the participant will attempt to recreate the shape using the modeling abilities of the device. The user will be instructed to indicate when they believe they are done, as well as to think aloud about their modeling process. Each modeling task will be capped at ten minutes. The time to completion (of lack thereof), completion code, observational notes, and video shall be recorded.

Task 2: Path Modeling

The participant will be given a brief demo of how the path modeling mode interprets the points from the device. The user will then be presented with a series of four (4) plastic, 3D-printed models that were modeled on the device using the path mode. For each of these shapes, the participant will attempt to recreate the shape using the modeling abilities of the device. The user will be instructed to indicate when they believe they are done, as well as to think aloud about their modeling process. Each modeling task will be capped at ten minutes. The time to completion (of lack thereof), completion code, observational notes, and video shall be recorded.

Task 3: Minimal Spanning Tree Modeling

The participant will be given a brief demo of how the minimal spanning tree (aka tree) modeling mode interprets the points from the device. The user will then be presented with a series of four (4) plastic, 3D-printed models that were modeled on the device using the tree mode. For each of these shapes, the participant will attempt to recreate the shape using the modeling abilities of the device. The user will be instructed to indicate when they believe they are done, as well as to think aloud about their modeling process. Each modeling task will be capped at ten minutes. The time to completion (of lack thereof), completion code, observational notes, and video shall be recorded.

Task 4: Freehand Modeling

After the modeling tasks are complete, participants are invited to “freestyle” model an object of their choosing, using any of the three modeling modes. By asking participants to think aloud about their intentions and thinking processes during this exercise, we hope that a deeper understanding may be gained of the strengths and weaknesses of the system, as well as the thought processes and engagement of the users in attempting to model a specific model of their own choosing. These saved models are analyzed, based on which mode was used to create them, complexity (based on number of points used), and whether the shape was ‘exploratory’ or ‘intentional’ (i.e., was the end artifact a result of sort of happy accident, or the result of intentional process to create a specific model).

For the first three modeling tasks (but not the freestyle modeling), time to completion (or request to move on) is recorded, along with an outcome code. The outcome is coded according to a set of conditions detailed below in table 4.1, and was developed upon analysis of the recorded video, in an attempt to fit the sorts of repeated behaviors that were in fact observed.

Participants were asked to “think aloud” about their process, difficulties, modeling choices, etc. In the case that the user believed they had correctly modeled the shape (cases C and E2 in table 4.1) they were asked to explain their modeling strategy[1] . Their explanation was videotaped and analyzed based on the coding strategies laid out in “The Importance of Gesture in Children’s Spatial Reasoning”([35], p.1264), laid out in table 4.5 below. The rationale for performing this analysis in based in part on work by Ehrlich, Levine, and Goldin-Meadow [35][81][53], which suggests that the frequency of gesture and relationships between speech and gesture act as a window into the

[1] Cases E1,E3,E4, and I did not provide the grounds from which to ask about modeling strategy and so were not recorded.

Table 4.1: The coding used in analyzing the modeling exercise outcomes, based on observations from video taken during the study.

Category	*Code*	*Definition*
Correct	C	A complete and correct modeling of the shape
Error in recognition	E1	The correct shape was modeled, but the user did not identify it
Error in belief	E2	A belief that the modeled shape has been modeled correctly, when it has not
Error in implementation	E3	User knew shape was incorrect, and gave a correct explanation
Error in strategy	E4	Knew shape was incorrect, and did not know why or gave an incorrect explanation as to why
Error in proportion	EP	The general shape is correct, but the proportions in one or more dimensions is off (e.g. too tall, not wide enough, etc.)
Incomplete	I	Participant ran out of time, gave up, or asked to move on

learning state of the subjects.

Table 4.2: The various coding strategies used in the video analysis of subjects' modeling strategy explanations. Borrowed and adapted from [35].

Category	*Definition*	*SpeechExamples*	*GestureExamples*
Movement	Any indication of movement	"Just slide them together and then it looks like that"	Miming movement with the hands
Perceptual Features	Focus on a particular feature of the model	"Because there is a little bend in here and a point thing here"	pointing to a specific feature on the model
Perceptual Whole	Any indication of seeing the model as a whole	"It looks like an arrow!"	Gesture indicating inclusion of the whole shape
Vague	An expression of strategy that the coder cannot decipher	"Because I looked at that and I looked at the differences"	Waving gestures above the computer o device that do not indicate any specific strategy
Other	Any strategy not listed above	"And here is like half of it. But so and two halves make a whole"	Using the hand to form a straight line through the middle of the whole shape to represent the line of symmetry

The second session is similar to the first, with the subject using the device not used in session one, and with 12 new models. Once modeling on the second device is completed, users will take a second spatial reasoning assessment of an additional ten questions to help gauge if any meaningful difference in spatial reasoning skills has occurred throughout the study.

A slightly modified version of the software was used for the user study, eliminating several of the functions not being evaluated for the sake of presenting a clear interface for the users. The multiple hull modes, spline, load, and save functions (described in Chapter 2) were eliminated, and the rest of the graphical user interface was reorganized and streamlined. We combined the three different .stl export buttons into a single export button that handled all three modes, changed the order of the remaining buttons and made them larger, and made the X,Y, and Z axis markings larger.

4.3.1 Results

This section reports on the results from our study, relaying our findings across both sessions, genders, modeling modes, and spatial reasoning scores in an attempt to tease out what conclusions, if any, we might make about the strengths and weaknesses of our devices as well as how interacting with our devices affected user's spatial reasoning abilities, 3D modeling skills, or congruence between speech and gesture in explaining the cognitive learning state of the user.

4.3.1.1 Modeling Results

In this section we will focus on delivering the results from the modeling exercises - users went through two sessions, modeling 12 shapes each time (4 shapes each using convex hull, path, and minimal spanning tree modes) for a total of 24 exercises. For each modeling task, a result code was recorded per the rubric shown in table 4.1. One user dropped out of the study (user 6) before completing round one, leaving us to report on 19 users, 10 of whom started on the PopCAD and 9 of whom started with the SnapCAD. 16 users completed session 2, 7 girls and 9 boys, split evenly over the two devices (four on each).

Out of the 228 modeling tasks in session one, the group successfully modeled 127, or roughly

Table 4.3: This table gives the subject's age, gender, and number of correctly completed modeling tasks during each of the two sessions.

User	*Age*	*Gender*	*Device*	*S1 Score*	*Device*	*S2 Score*
1	14	M	Pop	11	Snap	9
2	12	F	Snap	6	Pop	8
3	15	M	Pop	6	Snap	3
4	13	M	Snap	4	Pop	5
5	12	M	Pop	5	Snap	6
7	15	M	Pop	12	Snap	11
8	12	F	Snap	4	Pop	8
9	18	M	Pop	8	Snap	3
10	14	F	Pop	12	Snap	10
11	17	M	Snap	9	Pop	12
12	12	M	Snap	4	Pop	7
13	13	M	Pop	9	–	–
14	14	M	Snap	1	Pop	5
15	12	M	Snap	1	–	–
16	13	M	Pop	12	–	–
17	13	F	Pop	11	Snap	9
18	17	F	Snap	8	Pop	12
19	11	F	Pop	4	Snap	3
20	13	F	Snap	0	Pop	5

56%. Those users who started with SnapCAD performed 37 of 108 tasks, or 34%, while those using the PopCAD device completed 90 of 120 tasks correctly, for a success rate of 75%. Girls completed 45 of 84 tasks (54%), while boys correctly completed 82 of 144 tasks (58%). Individual scores ranged from 0 to 12 (perfect), with an overall overage of 6.68 correct shapes per user. Average correct shapes per user was 4.11 for SnapCAD and 9.00 for PopCAD.

In session two, 116 of 192 (60%) tasks were performed correctly, with SnapCAD modelers correctly representing 54 of 96 shapes (56%) and PopCAD modelers completing 62 of 96 shapes, or roughly 65%. Girls completed 55 of 84 tasks (65%) while boys completed 61 of 108 tasks for 56%. Individual scores ranged from 3 to 12 (perfect), with an average of 7.25 correct shapes overall, while the average correct shapes per user was 6.75 for SnapCAD and 7.75 for PopCAD.

Table 4.4: This table shows the number of correct models generated from a given shape, broken down by device, gender, and average modeling time spent on the shape. Session one results only.

Shape	*Total NumberCorrect*	*PopCAD Only*	*SnapCAD Only*	*Average ModelingTime*	*PopCAD Only*	*SnapCAD Only*
Convex Hull 1	8	7	1	6:33	5:25	7:49
Convex Hull 2	13	9	4	4:34	3:03	6:15
Convex Hull 3	8	7	1	5:29	4:33	6:32
Convex Hull 4	11	7	4	4:52	4:31	5:14
Path 1	17	10	7	2:08	1:22	3:00
Path 2	14	9	5	4:39	3:24	6:02
Path 3	8	6	2	6:21	4:52	8:00
Path 4	9	7	2	4:12	1:56	6:43
Tree 1	12	8	4	2:10	1:10	3:17
Tree 2	8	6	2	2:25	1:08	3:51
Tree 3	8	5	3	3:11	1:29	5:04
Tree 4	11	8	3	4:24	3:12	5:43
Total	127	90	37	4:15	3:00	5:38

Table 4.5: Modeling times per user over all modeling exercises in session 1.

Mode	*Combined AverageTime*	*Number Correct*	*PopCAD Only*	*SnapCAD Only*	*Girls*	*Boys*
Convex Hull	21:30:21	40 / 76	30	10		
Path	17:21:57	48 / 76	33	15		
Minimal Spanning Tree	12:11:52	39 / 76	27	12		
Overall	51:04:10	127 / 228	90	37	45	82

4.3.1.2 Mental Transformation Task Results

Subjects were given two sets of 10 mental transformation problems, as discussed previously in the procedure section. The first set was given before the first modeling session, as a sort of pre-assessment. The second set was given after the second modeling session as a post-test. We recorded performance data by session and per user, and present the results broken out by the type of symmetry represented in the shape (unilateral or bilateral) and the type of translation or rotation performed on the shape (direct or diagonal translation, direction or diagonal rotation), meaning that each shape had both a symmetry type and a translation type.

Table 4.6: Mental Transformation Task by Shape Profile and Translation Type

	Bilateral Symmetry	*Unilateral Symmetry*	*Direct Translation*	*Direct Rotation*	*Diagonal Translation*	*Diagonal Rotation*	*Total Correct*
*Session*1	109/120	57/80	54/60	34/40	56/60	22/40	166/200
Girls	43/48	23/32	22/24	13/16	23/24	8/16	66/80
Boys	66/72	34/48	32/36	21/24	33/36	14/24	100/120
*Session*2	72/80	69/80	31/32	39/48	28/32	43/48	141/160
Girls	32/35	30/35	13/14	18/21	12/14	19/21	62/70
Boys	42/45	39/45	18/18	21/17	16/18	24/27	79/90
Combined	181/200	126/160	85/92	73/88	84/92	65/88	307/360
Girls	75/83	53/67	35/38	31/37	35/38	27/37	256/300
Boys	108/117	73/93	50/54	42/51	49/54	38/51	360/420

Table 4.7: Mental Transformation Task Performance Per User.

User	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Set 1	7	10	8	7	9	8	10	9	9	7	9	8	10	9	6	8	10	8	7	7
Set 2	9	9	10	6	10	-	10	10	10	10	9	9	-	6	-	-	10	8	5	10
Total	16	19	18	13	19	-	20	19	19	17	18	17	-	15	-	-	20	16	12	17
Change	+2	-1	+2	-1	+1	-	0	+1	+1	+3	0	+1	-	-3	-	-	0	0	-2	+3

4.3.1.3 Speech and Gesture Coding

4.3.2 Discussion

In order to attempt to judge each shape's complexity, we sought out a previously-defined set of criteria by which to judge "complexity". As it turns out, there is a long and thorough discussion of complexity in relation to **two-dimensional** shapes, starting seemingly with Fred Attneave and Malcolm Arnoult[16][15] in the mid 1950's, who define methods of generating random two-dimensional shapes and examine their physical characteristics in relation to their judged complexity. As it turns out (in [15] as well as others' follow-up work) the "Number of Turns" in the shape was responsible for significant amount (nearly 80% in Attneave's study) of the preceived complexity of a shape. "Number of Turns" is defined as, "the number of maxima (regardless of sign) in one cycle of the function relating curvature to distance along the contour. This function is a series of spikes for any angular shape, and a step-function for any curved shape..." (see pp. 226 of the aforementioned article). Symmetry, angular variability, and squared perimeter over area also had some affect.

However, as it seems unclear to us how one might adapt a "Number of Turns" rating to a true **three dimensional** model. Although many studies claim to have studied complexity in relation to mental transformation tasks, starting with Shepard and Metzler[117], who instead used perspective line drawings, not actual physical models. This had an advantage for the types of mental rotation tasks they were performing (recognition of matching pairs), and similarly set off a wave of studies using the same (or similar) "faux 3D" stimuli[89][118][125].

All of which leads us to determine (as best we way) the complexity of the shapes we presented in the study, as a way of teasing out any correlation between complexity and performance. In lieu of attempting an exact number of turns estimate, we included three criteria: (a) the minimum number of lights necessary to guarantee the correct shape,[2] (b) the number of faces (for convex hull models only), the number of line segments (for path models only), or the number of distinct branches (for tree models only), and (c) a symmetry score based on number of lines of symmetry, from 3 (indicating asymmetry) to 0 (indicating 3 or more lines of symmetry). The scaling for symmetry comes from the belief that indicators (a) and (b) above are more closely aligned with Attneave's "number of turns" metric (being highly correlated to perceived difficulty), while symmetry was much less correlated to complexity (although symmetry did still play a part), so we made the scale as low as possible so that it would weigh less on the overall complexity score of a model. So, for example, a regular octohedron would have 6 points, 8 faces, and a 0 symmetry score for an overall difficulty score of 14. The complexity score of each shape is shown in 4.8 next to the number of times it was modeled correctly. The shapes in each session were of course different, but are labeled

[2] In minimal spanning tree models where a placement of lights results in several possible correct formations, only one of which is the desired shape, we add points necessary to "force" the correct representation.

the same in this table, indicating the order in which they were presented.

Table 4.8: Complexity of Models and Modeling Performance (CH = Convex Hull, P = Path, T = Minimal Spanning Tree)

	CH1	CH2	CH3	CH4	P1	P2	P3	P4	T1	T2	T3	T4
Session 1												
Complexity Score	14	19	19	19	7	18	20	24	9	13	17	17
Performance of 19	8	13	8	11	17	14	8	9	12	8	8	11
Session 2												
Complexity Score	12	20	15	13	14	14	18	17	21	17	16	25
Performance of 16	7	9	11	11	11	13	8	12	7	11	7	9

Chapter 5

Vision

Chapter 6

Future Work

Bibliography

[1] The printed world. http://www.economist.com/node/18114221, 2011.

[2] 123d catch. http://www.123dapp.com/catch, 2013.

[3] Makerbot. http://www.makerbot.com/, 2013.

[4] Ponoko. http://www.ponoko.com/, 2013.

[5] Stratasys. http://www.stratasys.com/, 2013.

[6] Thingiverse. http://www.thingiverse.com/, 2013.

[7] Tinkercad. http://www.tinkercad.com/, 2013.

[8] Dor Abrahamson and Dragan Trninic. Toward an embodied-interaction design framework for mathematical concepts. In Proceedings of the 10th International Conference on Interaction Design and Children, IDC '11, pages 1–10, New York, NY, USA, 2011. ACM.

[9] S. Ananthanarayan, K. Siek, and M. Eisenberg. Towards the Crafting of Personal Health Technologies. Submitted for publication. 2013.

[10] C. Anderson. Makers: The New Industrial Revolution. Random House, 2012.

[11] Alissa N. Antle. Designing tangibles for children: what designers need to know. In CHI '07 Extended Abstracts on Human Factors in Computing Systems, CHI EA '07, pages 2243–2248, New York, NY, USA, 2007. ACM.

[12] Alissa N. Antle, Milena Droumeva, and Daniel Ha. Thinking with hands: an embodied approach to the analysis of children's interaction with computational objects. In CHI '09 Extended Abstracts on Human Factors in Computing Systems, CHI EA '09, pages 4027–4032, New York, NY, USA, 2009. ACM.

[13] Alissa N. Antle and Sijie Wang. Comparing motor-cognitive strategies for spatial problem solving with tangible and multi-touch interfaces. In Proceedings of the 7th International Conference on Tangible, Embedded and Embodied Interaction, TEI '13, pages 65–72, New York, NY, USA, 2013. ACM.

[14] Alissa N. Antle, Alyssa F. Wise, and Kristine Nielsen. Towards utopia: designing tangibles for learning. In Proceedings of the 10th International Conference on Interaction Design and Children, IDC '11, pages 11–20, New York, NY, USA, 2011. ACM.

[15] Fred Attneave. Physical determinants of the judged complexity of shapes. Journal of Experimental Psychology, 53(4):221, 1957.

[16] Fred Attneave and Malcolm D Arnoult. The quantitative study of shape and pattern perception. Psychological bulletin, 53(6):452, 1956.

[17] Saskia Bakker, Alissa N Antle, and Elise Van Den Hoven. Embodied metaphors in tangible interaction design. Personal and Ubiquitous Computing, 16(4):433–449, 2012.

[18] Saskia Bakker, Elise van den Hoven, and Alissa N Antle. Moso tangibles: evaluating embodied learning. In Proceedings of the fifth international conference on Tangible, embedded, and embodied interaction, pages 85–92. ACM, 2011.

[19] Nicholas A Basbanes. On Paper: The Everything of Its Two-Thousand-Year History. Random House LLC, 2013.

[20] RQRQ Berry, G Bull, C Browning, C Thomas, K Starkweather, and J Aylor. Preliminary considerations regarding use of digital fabrication to incorporate engineering design principles in elementary mathematics education. Contemporary Issues in Technology and Teacher Education, 10(2):167–172, 2010.

[21] Allen Bevans, Ying-Ting Hsiao, and Alissa Antle. Supporting children's creativity through tangible user interfaces. In CHI '11 Extended Abstracts on Human Factors in Computing Systems, CHI EA '11, pages 1741–1746, New York, NY, USA, 2011. ACM.

[22] R Breckinridge Church and Susan Goldin-Meadow. The mismatch between gesture and speech as an index of transitional knowledge. Cognition, 23(1):43–71, 1986.

[23] Jack Breen, Robert Nottrot, and Martijn Stellingwerff. Tangible virtualityperceptions of computer-aided and physical modelling. Automation in Construction, 12(6):649–653, 2003.

[24] Erwin H Brinkmann. Programed instruction as a technique for improving spatial visualization. Journal of Applied Psychology, 50(2):179, 1966.

[25] Norman Brosterman, Kiyoshi Togashi, and Eric Himmel. Inventing kindergarten. HN Abrams New York, NY, 1997.

[26] Craig Brown and Amy Hurst. Viztouch: automatically generated tactile visualizations of coordinate spaces. In Proceedings of the Sixth International Conference on Tangible, Embedded and Embodied Interaction, TEI '12, pages 131–138, New York, NY, USA, 2012. ACM.

[27] Leah Buechley and Michael Eisenberg. The lilypad arduino: Toward wearable engineering for everyone. Pervasive Computing, IEEE, 7(2):12–15, 2008.

[28] Leah Buechley, Mike Eisenberg, Jaime Catchen, and Ali Crockett. The lilypad arduino: using computational textiles to investigate engagement, aesthetics, and diversity in computer science education. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '08, pages 423–432, New York, NY, USA, 2008. ACM.

[29] Leah Buechley, Sue Hendrix, and Mike Eisenberg. Paints, paper, and programs: first steps toward the computational sketchbook. In Proceedings of the 3rd International Conference on Tangible and Embedded Interaction, pages 9–12. ACM, 2009.

[30] Leah Buechley, David Mellis, Hannah Perner-Wilson, Emily Lovell, and Bonifaz Kaufmann. Living wall. In Proceedings of the international conference on Multimedia - MM '10, page 1401, New York, New York, USA, October 2010. ACM Press.

[31] Leah Buechley and Hannah Perner-Wilson. Crafting technology. ACM Transactions on Computer-Human Interaction, 19(3):1–21, October 2012.

[32] A. Clark. Being There: Putting Brain, Body, and World Together Again. A Bradford book. MIT Press, 1998.

[33] Marcelo Coelho, L Hall, J Berzowska, and P Maes. Pulp-based computing: a framework for building computers out of paper. ...Human Factors in Computing ..., pages 3527–3528, 2009.

[34] Matthew Conway, Steve Audia, Tommy Burnette, Dennis Cosgrove, and Kevin Christiansen. Alice: lessons learned from building a 3d system for novices. In Proceedings of the SIGCHI conference on Human Factors in Computing Systems, CHI '00, pages 486–493, New York, NY, USA, 2000. ACM.

[35] Stacy B Ehrlich, Susan C Levine, and Susan Goldin-Meadow. The importance of gesture in children's spatial reasoning. Developmental psychology, 42(6):1259, 2006.

[36] M. Eisenberg, A. Eisenberg, S. Hendrix, G. Blauvelt, D. Butter, J. Garcia, R. Lewis, and T. Nielsen. As we may print: new directions in output devices and computational crafts for children. In Proceedings of the 2003 conference on Interaction design and children, IDC '03, pages 31–39, New York, NY, USA, 2003. ACM.

[37] Michael Eisenberg. Pervasive fabrication: Making construction ubiquitous in education. In Proceedings of the Fifth IEEE International Conference on Pervasive Computing and Communications Workshops, PERCOMW '07, pages 193–198, Washington, DC, USA, 2007. IEEE Computer Society.

[38] Michael Eisenberg. Educational fabrication, in and out of the classroom. In Society for Information Technology & Teacher Education International Conference, volume 2011, pages 884–891, 2011.

[39] Michael Eisenberg, Nwanua Elumeze, Michael MacFerrin, and Leah Buechley. Children's programming, reconsidered: settings, stuff, and surfaces. In Proceedings of the 8th International Conference on Interaction Design and Children, IDC '09, pages 1–8, New York, NY, USA, 2009. ACM.

[40] Michael Eisenberg, Wendy Mackay, Allison Druin, Sheila Lehman, and Mitchel Resnick. Real meets virtual: blending real-world artifacts with computational media. In Conference Companion on Human Factors in Computing Systems, CHI '96, pages 159–160, New York, NY, USA, 1996. ACM.

[41] Michael Eisenberg, Ann Nishioka, and M. E. Schreiner. Helping users think in three dimensions: steps toward incorporating spatial cognition in user modelling. In Proceedings of the 2nd international conference on Intelligent user interfaces, IUI '97, pages 113–120, New York, NY, USA, 1997. ACM.

[42] Fab@Home. www.fabathome.org. 2011.

[43] Desktop Factory. www.desktopfactory.com/. 2011.

[44] ES Ferguson. Engineering and the Mind's Eye. The MIT Press, 1992.

[45] Thomas Fischer and Wing Lau. Marble track music sequencers for children. In Proceedings of the 2006 conference on Interaction design and children, IDC '06, pages 141–144, New York, NY, USA, 2006. ACM.

[46] Mark D Folk and R Duncan Luce. Effects of stimulus complexity on mental rotation rate of polygons. Journal of Experimental Psychology: Human Perception and Performance, 13(3):395, 1987.

[47] Sean Follmer and Hiroshi Ishii. Kidcad: digitally remixing toys through tangible tools. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '12, pages 2401–2410, New York, NY, USA, 2012. ACM.

[48] Sean Follmer, Micah Johnson, Edward Adelson, and Hiroshi Ishii. deform: an interactive malleable surface for capturing 2.5d arbitrary objects, tools and touch. In Proceedings of the 24th annual ACM symposium on User interface software and technology, UIST '11, pages 527–536, New York, NY, USA, 2011. ACM.

[49] F. Froebel and J. Jarvis. The Education of Man. A. Lovell Company, New York, 1886.

[50] Neil Gershenfeld. Fab: The Coming Revolution on Your Desktop–from Personal Computers to Personal Fabrication. Basic Books, Inc., New York, NY, USA, 2007.

[51] Wooi Boon Goh, L. L. Chamara Kasun, Fitriani, Jacquelyn Tan, and Wei Shou. The i-cube: design considerations for block-based digital manipulatives and their applications. In Proceedings of the Designing Interactive Systems Conference, DIS '12, pages 398–407, New York, NY, USA, 2012. ACM.

[52] H. Nusbaum S. D. Delly Goldin-Meadow, S. and S. Wagner. Explaining math: Gesturing lightens the load. Psychological Science, pages pp. 516–22, 1991.

[53] Susan Goldin-Meadow. Hearing gesture: How our hands help us think. Harvard University Press, 2005.

[54] Ira Greenberg. Processing: creative coding and computational art. Apress, 2007.

[55] Mark D. Gross and Michael Eisenberg. Why toys shouldn't work "like magic": Children's technology and the values of construction and control. In Proceedings of the The First IEEE International Workshop on Digital Game and Intelligent Toy Enhanced Learning, DIGITEL '07, pages 25–32, Washington, DC, USA, 2007. IEEE Computer Society.

[56] Elizabeth G Hainstock. The Essential Montessori. New American Library, 1978.

[57] B. Hart. Will 3d printing change the world? http://www.forbes.com/sites/gcaptain/2012/03/06/will-3d-prin 2012.

[58] George W Hart. Procedural generation of sculptural forms. Bridges, 2008:209, 2008.

[59] Susan Lee Hendrix. Popup Workshop: Computationally Enhanced Paper Engineering for Children. PhD thesis, University of Colorado, 2008.

[60] Mark Howison, Dragan Trninic, Daniel Reinholz, and Dor Abrahamson. The mathematical imagery trainer: from embodied interaction to conceptual learning. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pages 1989–1998. ACM, 2011.

[61] http://507movements.com/. 507 mechanical movements. 2013.

[62] Yingdan Huang and Michael Eisenberg. Easigami: virtual creation by physical folding. In Proceedings of the Sixth International Conference on Tangible, Embedded and Embodied Interaction, TEI '12, pages 41–48, New York, NY, USA, 2012. ACM.

[63] Yingdan Huang, Mark D Gross, Ellen Yi-Luen Do, and Mike Eisenberg. Easigami: a reconfigurable folded-sheet tui. In Proceedings of the 3rd International Conference on Tangible and Embedded Interaction, TEI '09, pages 107–112, New York, NY, USA, 2009. ACM.

[64] Amy Hurst and Shaun Kane. Making "making" accessible. In Proceedings of the 12th International Conference on Interaction Design and Children, IDC '13, pages 635–638, New York, NY, USA, 2013. ACM.

[65] Barbel Inhelder and Jean Piaget. The Growth of Logical Thinking from Childhood to Adolescence : An Essay on the Construction of Formal Operational Structures. Basic, 1958.

[66] Instructables. http://instructables.com.

[67] Hiroshi Ishii. Tangible bits: beyond pixels. In Proceedings of the 2nd international conference on Tangible and embedded interaction, TEI '08, pages xv–xxv, New York, NY, USA, 2008. ACM.

[68] Hiroshi Ishii and Brygg Ullmer. Tangible bits: towards seamless interfaces between people, bits and atoms. In Proceedings of the ACM SIGCHI Conference on Human factors in computing systems, CHI '97, pages 234–241, New York, NY, USA, 1997. ACM.

[69] Sam Jacoby and Leah Buechley. Drawing the electric. In Proceedings of the 12th International Conference on Interaction Design and Children - IDC '13, pages 265–268, New York, New York, USA, June 2013. ACM Press.

[70] Gabe Johnson, Mark Gross, Ellen Yi-Luen Do, and Jason Hong. Sketch it, make it: sketching precise drawings for laser cutting. In CHI '12 Extended Abstracts on Human Factors in Computing Systems, CHI EA '12, pages 1079–1082, New York, NY, USA, 2012. ACM.

[71] Samuel Johnson and AnnMarie P. Thomas. Squishy circuits: a tangible medium for electronics education. In CHI '10 Extended Abstracts on Human Factors in Computing Systems, CHI EA '10, pages 4099–4104, New York, NY, USA, 2010. ACM.

[72] Yasmin B. Kafai, Kylie A. Peppler, Quinn Burke, Michael Moore, and Diane Glosson. Frbel's forgotten gift: textile construction kits as pathways into play, design and computation. In Proceedings of the 9th International Conference on Interaction Design and Children, IDC '10, pages 214–217, New York, NY, USA, 2010. ACM.

[73] Yoshihiro Kawahara, Steve Hodges, Benjamin S Cook, and Gregory D Abowd. Instant Inkjet Circuits : Lab-based Inkjet Printing to Support Rapid Prototyping of UbiComp Devices. pages 363–372, 2013.

[74] Yoshihiro Kawahara, Steve Hodges, Benjamin S. Cook, Cheng Zhang, and Gregory D. Abowd. Instant inkjet circuits. In Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing - UbiComp '13, page 363, New York, New York, USA, September 2013. ACM Press.

[75] Scott R. Klemmer, Björn Hartmann, and Leila Takayama. How bodies matter: five themes for interaction design. In Proceedings of the 6th conference on Designing Interactive systems, DIS '06, pages 140–149, New York, NY, USA, 2006. ACM.

[76] Naoya Koizumi, Kentaro Yasu, Angela Liu, Maki Sugimoto, and Masahiko Inami. Animated paper. Computers in Entertainment, 8(2):1, December 2010.

[77] Dennis Krannich, Bernard Robben, and Sabrina Wilske. Digital fabrication for educational contexts. In Proceedings of the 11th International Conference on Interaction Design and Children, IDC '12, pages 375–376, New York, NY, USA, 2012. ACM.

[78] G. Lakoff and R. Nuñez. Where Mathematics Come From: How The Embodied Mind Brings Mathematics Into Being. Basic Books, Inc., 2001.

[79] Ben Leduc-Mills and Michael Eisenberg. The ucube: a child-friendly device for introductory three-dimensional design. In Proceedings of the 10th International Conference on Interaction Design and Children, IDC '11, pages 72–80, New York, NY, USA, 2011. ACM.

[80] Ben Leduc-Mills, Halley Profita, and Michael Eisenberg. "seeing solids" via patterns of light: evaluating a tangible 3d-input device. In Proceedings of the 11th International Conference on Interaction Design and Children, IDC '12, pages 377–380, New York, NY, USA, 2012. ACM.

[81] Susan C Levine, Janellen Huttenlocher, Amy Taylor, and Adela Langrock. Early sex differences in spatial skill. Developmental psychology, 35(4):940, 1999.

[82] Hod Lipson and Melba Kurman. Factory@ home: The emerging economy of personal fabrication. A report commissioned by the US Office of Science and Technology Policy, 2010.

[83] Hod Lipson, Francis C Moon, Jimmy Hai, and Carlo Paventi. 3-d printing the history of mechanisms. Journal of Mechanical Design, 127:1029, 2005.

[84] Evan Malone and Hod Lipson. Fab@ home: the personal desktop fabricator kit. Rapid Prototyping Journal, 13(4):245–255, 2007.

[85] James H Mathewson. Visual-spatial thinking: An aspect of science overlooked by educators. Science Education, 83(1):33–54, 1999.

[86] Timothy S. McNerney. From turtles to tangible programming bricks: explorations in physical language design. Personal Ubiquitous Comput., 8(5):326–337, September 2004.

[87] Arduino Mega. arduino.cc/en/main/arduinoboardmega2560. 2011.

[88] David A. Mellis, Sam Jacoby, Leah Buechley, Hannah Perner-Wilson, and Jie Qi. Microcontrollers as material: crafting circuits with paper, conductive ink, electronic components, and an "untoolkit". In Proceedings of the 7th International Conference on Tangible, Embedded and Embodied Interaction, TEI '13, pages 83–90, New York, NY, USA, 2013. ACM.

[89] Jacqueline Metzler and Roger N Shepard. Transformational studies of the internal representation of three-dimensional objects. 1974.

[90] Jason Mickelson and Wendy Ju. Math propulsion: engaging math learners through embodied performance & visualization. In Proceedings of the fifth international conference on Tangible, embedded, and embodied interaction, pages 101–108. ACM, 2011.

[91] Maria Montessori. The Montessori Method. Frederick Stokes Co., New York, NY, 1912.

[92] Catarina Mota. The rise of personal fabrication. In Proceedings of the 8th ACM conference on Creativity and cognition, pages 279–288. ACM, 2011.

[93] Stefanie Mueller, Bastian Kruck, and Patrick Baudisch. LaserOrigami. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems - CHI '13, page 2585, New York, New York, USA, April 2013. ACM Press.

[94] Stefanie Mueller, Pedro Lopes, and Patrick Baudisch. Interactive construction: interactive fabrication of functional mechanical devices. In Proceedings of the 25th annual ACM symposium on User interface software and technology, UIST '12, pages 599–606, New York, NY, USA, 2012. ACM.

[95] Ricardo Nemirovsky, Cornelia Tierney, and Tracey Wright. Body motion and graphing. Cognition and instruction, 16(2):119–172, 1998.

[96] Nora Newcombe and Janellen Huttenlocher. Making space: The development of spatial representation and reasoning. The MIT Press, 2003.

[97] Brian N. O'Connell. The development of the paperbots robotics kit for inexpensive robotics education activities for elementary students. Master's thesis, Tufts University, 2013.

[98] S. Papert. Mindstorms: Children, Computers, and Powerful Ideas. Basic Books, Inc., 1980.

[99] Amanda J. Parkes, Hayes Solos Raffle, and Hiroshi Ishii. Topobo in the wild: longitudinal evaluations of educators appropriating a tangible interface. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '08, pages 1129–1138, New York, NY, USA, 2008. ACM.

[100] Jean Piaget and Barbel Inhelder. The childs conception of space (fj langdon & jl lunzer, trans.). New York, 1967.

[101] Processing. www.processing.org. 2011.

[102] The RepRap Project. reprap.org/wiki/main_page. 2011.

[103] Jie Qi and Leah Buechley. Electronic popables: exploring paper-based computing through an interactive pop-up book. In Proceedings of the fourth international conference on Tangible, embedded, and embodied interaction, TEI '10, pages 121–128, New York, NY, USA, 2010. ACM.

[104] Jie Qi and Leah Buechley. Animating paper using shape memory alloys. In Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems - CHI '12, page 749, New York, New York, USA, May 2012. ACM Press.

[105] Hayes Solos Raffle. Sculpting behavior: a tangible language for hands-on play and learning. PhD thesis, MIT, Cambridge, MA, USA, 2008. AAI0821001.

[106] Hayes Solos Raffle, Amanda J. Parkes, and Hiroshi Ishii. Topobo: a constructive assembly system with kinetic memory. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '04, pages 647–654, New York, NY, USA, 2004. ACM.

[107] Betty Repacholi and Alison Gopnik. Early reasoning about desires: Evidence from 14- and 18-month-olds. In Developmental Psychology, pages 12–21, 1997.

[108] Mitchel Resnick. All i really need to know (about creative thinking) i learned (by studying how children learn) in kindergarten. In Proceedings of the 6th ACM SIGCHI conference on Creativity & cognition, C&C '07, pages 1–6, New York, NY, USA, 2007. ACM.

[109] Mitchel Resnick, Fred Martin, Robert Berg, Rick Borovoy, Vanessa Colella, Kwin Kramer, and Brian Silverman. Digital manipulatives: new toys to think with. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '98, pages 281–287, New York, NY, USA, 1998. ACM Press/Addison-Wesley Publishing Co.

[110] Mitchel Resnick and Brian Silverman. Some reflections on designing construction kits for kids. In Proceedings of the 2005 conference on Interaction design and children, IDC '05, pages 117–122, New York, NY, USA, 2005. ACM.

[111] Rhino. www.rhino3d.com. 2011.

[112] Eric Rosenbaum and Jay Silver. Makey makey: improvising tangible and nature-based user interfaces. In Proceedings of the Sixth International Conference on Tangible, Embedded and Embodied Interaction, TEI '12, pages 367–370, New York, NY, USA, 2012. ACM.

[113] Greg Saul, Cheng Xu, and Mark D. Gross. Interactive paper devices. In Proceedings of the fourth international conference on Tangible, embedded, and embodied interaction - TEI '10, page 205, New York, New York, USA, January 2010. ACM Press.

[114] Eric Schweikardt, Nwanua Elumeze, Mike Eisenberg, and Mark D Gross. A tangible construction kit for exploring graph theory. In Proceedings of the 3rd International Conference on Tangible and Embedded Interaction, TEI '09, pages 373–376, New York, NY, USA, 2009. ACM.

[115] Eric Schweikardt and Mark D. Gross. roblocks: a robotic construction kit for mathematics and science education. In Proceedings of the 8th international conference on Multimodal interfaces, ICMI '06, pages 72–75, New York, NY, USA, 2006. ACM.

[116] Shapeways. www.shapeways.com. 2011.

[117] RN Shepard and Jacqueline Metzler. Mental rotation of three-dimensional objects. 1971.

[118] Shenna Shepard and Douglas Metzler. Mental rotation: Effects of dimensionality of objects and type of task. Journal of Experimental Psychology: Human Perception and Performance, 14(1):3, 1988.

[119] Francis T Siemankowski and Franklin C MacKnight. Spatial cognition, a success prognosticator in college science courses. 1971.

[120] Arnan Sipitakiat and Nusarin Nusen. Robo-blocks: designing debugging abilities in a tangible programming system for early primary school children. In Proceedings of the 11th International Conference on Interaction Design and Children, IDC '12, pages 98–105, New York, NY, USA, 2012. ACM.

[121] Google SketchUp. sketchup.google.com/. 2011.

[122] Solidways. www.solidworks.com. 2011.

[123] J. P. Spencer, M. Clearfield, D. Corbetta, B. Ulrich, P. Buchanan, and G. Schoner. Moving toward a grand theory of development: In memory of esther thelen. In Child Development, pages 1521–1538, 2006.

[124] Cristina Sylla, Pedro Branco, Sérgio Gonçalves, Clara Coutinho, and Paulo Brito. t-books. In Proceedings of the 11th International Conference on Interaction Design and Children - IDC '12, page 323, New York, New York, USA, June 2012. ACM Press.

[125] Steven G Vandenberg and Allan R Kuse. Mental rotations, a group test of three-dimensional spatial visualization. Perceptual and motor skills, 47(2):599–604, 1978.

[126] Danli Wang, Cheng Zhang, and Hongan Wang. T-maze: a tangible programming tool for children. In Proceedings of the 10th International Conference on Interaction Design and Children, IDC '11, pages 127–135, New York, NY, USA, 2011. ACM.

[127] Ryoichi Watanabe, Yuichi Itoh, Masatsugu Asai, Yoshifumi Kitamura, Fumio Kishino, and Hideo Kikuchi. The soul of activecube: implementing a flexible, multimodal, three-dimensional spatial tangible interface. Comput. Entertain., 2(4):15–15, October 2004.

[128] Michael Philetus Weller, Ellen Yi-Luen Do, and Mark D Gross. Escape machine: teaching computational thinking with a tangible state machine game. In Proceedings of the 7th international conference on Interaction design and children, IDC '08, pages 282–289, New York, NY, USA, 2008. ACM.

[129] Michael Philetus Weller, Mark D. Gross, and Ellen Yi-Luen Do. Tangible sketching in 3d with posey. In CHI '09 Extended Abstracts on Human Factors in Computing Systems, CHI EA '09, pages 3193–3198, New York, NY, USA, 2009. ACM.

[130] Karl D.D. Willis, Juncong Lin, Jun Mitani, and Takeo Igarashi. Spatial sketch: bridging between movement & fabrication. In Proceedings of the fourth international conference on Tangible, embedded, and embodied interaction, TEI '10, pages 5–12, New York, NY, USA, 2010. ACM.

[131] Karl D.D. Willis, Cheng Xu, Kuan-Ju Wu, Golan Levin, and Mark D. Gross. Interactive fabrication: new interfaces for digital fabrication. In Proceedings of the fifth international conference on Tangible, embedded, and embodied interaction, TEI '11, pages 69–72, New York, NY, USA, 2011. ACM.

[132] Margaret Wilson. Six views of embodied cognition. Psychonomic Bulletin Review, 9(4):625–636, 2002.

[133] Lesley Xie, Alissa N. Antle, and Nima Motamedi. Are tangibles more fun?: comparing children's enjoyment and engagement using physical, graphical and tangible user interfaces. In Proceedings of the 2nd international conference on Tangible and embedded interaction, TEI '08, pages 191–198, New York, NY, USA, 2008. ACM.

[134] Kening Zhu, Hideaki Nii, Owen Noel Newton Fernando, and Adrian David Cheok. Selective inductive powering system for paper computing. In Proceedings of the 8th International Conference on Advances in Computer Entertainment Technology - ACE '11, page 1, New York, New York, USA, November 2011. ACM Press.

[135] Amit Zoran and Joseph A. Paradiso. Freed: a freehand digital sculpting tool. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '13, pages 2613–2616, New York, NY, USA, 2013. ACM.

[136] Oren Zuckerman, Saeed Arida, and Mitchel Resnick. Extending tangible interfaces for education: digital montessori-inspired manipulatives. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '05, pages 859–868, New York, NY, USA, 2005. ACM.

Appendix A

Weird Exam Answers

About appendices: Each appendix follow the same page-numbering rules as a regular chapter; the first page of a (multi-page) appendix is not numbered. By the way, the following are supposedly authentic answers to English GCSE exams!

(1) The Greeks were a highly sculptured people, and without them we wouldn't have history. The Greeks also had myths. A myth is a female moth.

(2) Actually, Homer was not written by Homer but by another man of that name.

(3) Socrates was a famous Greek teacher who went around giving people advice. They killed him. Socrates died from an overdose of wedlock. After his death, his career suffered a dramatic decline.

(4) Julius Caesar extinguished himself on the battlefields of Gaul. The Ides of March murdered him because they thought he was going to be made king. Dying, he gasped out: Tee hee, Brutus.

(5) Nero was a cruel tyranny who would torture his subjects by playing the fiddle to them.

(6) In midevil times most people were alliterate. The greatest writer of the futile ages was Chaucer, who wrote many poems and verses and also wrote literature.

(7) Another story was William Tell, who shot an arrow through an apple while standing on his sons head.

(8) Writing at the same time as Shakespeare was Miguel Cervantes. He wrote Donkey Hote. The next great author was John Milton. Milton wrote Paradise Lost. Then his wife died and he wrote Paradise Regained.

(9) During the Renaissance America began. Christopher Columbus was a great navigator who discovered America while cursing about the Atlantic. His ships were called the Nina, the Pinta, and the Santa Fe.

(10) Gravity was invented by Issac Walton. It is chiefly noticeable in the autumn when the apples are falling off the trees.

(11) Johann Bach wrote a great many musical compositions and had a large number of children. In between he practiced on an old spinster which he kept up in his attic. Bach died from 1750 to the present. Bach was the most famous composer in the world and so was Handel. Handel was half German half Italian and half English. He was very large.

(12) Soon the Constitution of the United States was adopted to secure domestic hostility. Under the constitution the people enjoyed the right to keep bare arms.

(13) The sun never set on the British Empire because the British Empire is In the East and the sun sets in the West.

(14) Louis Pasteur discovered a cure for rabbis. Charles Darwin was a naturalist who wrote the Organ of the Species. Madman Curie discovered radio. And Karl Marx became one of the Marx brothers.

Appendix B

Ode to Spot

(Data, Stardate 1403827) (A one-page chapter — page must be numbered!) Throughout the ages, from Keats to Giorchamo, poets have composed "odes" to individuals who have had a profound effect upon their lives. In keeping with that tradition I have written my next poem . . . in honor of my cat. I call it. . . Ode. . . to Spot. (Shot of Geordi and Worf in audience, looking mystified at each other.)

Felus cattus, is your taxonomic nomenclature
an endothermic quadruped, carnivorous by nature?
Your visual, olfactory, and auditory senses
contribute to your hunting skills, and natural defenses.
I find myself intrigued by your sub-vocal oscillations,
a singular development of cat communications
that obviates your basic hedonistic predilection
for a rhythmic stroking of your fur to demonstrate affection.
A tail is quite essential for your acrobatic talents;
you would not be so agile if you lacked its counterbalance.
And when not being utilized to aid in locomotion,
It often serves to illustrate the state of your emotion.

(Commander Riker begins to applaud, until a glance from Counselor Troi brings him to a halt.) Commander Riker, you have anticipated my denouement. However, the sentiment is appreciated. I will continue.

O Spot, the complex levels of behavior you display
connote a fairly well-developed cognitive array.
And though you are not sentient, Spot, and do not comprehend
I nonetheless consider you a true and valued friend.