Blucher Design Proceedings
Dezembro de 2014, Volume 1, Número 8
www.proceedings.blucher.com.br/evento/sigradi2014

Physical Computing without the Computing: Small Responsive Prototypes

Joshua Vermillion
School of Architecture
University of Nevada Las Vegas, United States
josh.vermillion@unlv.edu

Abstract

This paper outlines a framework for better understanding the appropriate skills and roles of design students as developed by the author for a one-week short course on the topic of physical computing and design. Severe time constraints forced an examination of how to introduce physical computing to students with novice understandings of these systems and how they work, while maintaining expectations to prototype and produce full-scale spatial installations. This framework allowed the short-course students to deliver focused and well-crafted self-assembling lattice prototypes, while integrating the complexities of human interaction, spatial effects, fabrication, detailing, and prototyping.

Keywords: Physical computing; Self-assembly; Responsive environments; Design education; Digital fabrication

Introduction

The built environment is rich with opportunities for embedding and integrating digital technologies to create responsive and adaptable systems. Physical computing, a term coined by Tom Igoe and Dan O'Sullivan of New York University, refers to these sorts of systems that can sense, interpret this data computationally and, in response, physically change (2004). The technical implementation of these systems requires a broad range of skills that span multiple knowledge domains—design, engineering, mechanics, programming and computer science, robotics, mathematics, electronics—just to name some. There are many examples of design teams negotiating these interdisciplinary challenges and deploying them to create responsive prototypes with at various scales and for various effect, for instance ceilings that move and change color (Senagala and Vermillion, 2009), walls that sense and emit light (Buente and Perry, 2013), panels that fold in shape according to sound and acoustic qualities (Thune, et al, 2012), or even completely immersive environments with parts that change, move, and reconfigure both locally and systemically (Beesley, 2012).

Assuming that the design of the built environment will increasingly integrate physical computing systems, does our architectural repertoire of skills and knowledge need to be adjusted to meet these challenges? In particular, how do we educate and prepare architecture professionals for this future of physically active and interactive environments? Will we be professional architects, and also have to be professional programmers, engineers, and electricians? This paper outlines a framework for better understanding the appropriate skills and roles of design students as developed by the author for a one-week short course on the topic of physical computing and design.

The process of designing a short course forced an examination of how to introduce physical computing to students with novice understandings of these systems and how they work. With expectations to prototype and produce a full-scale spatial installation, with one week from start to finish, we gave much thought on how to use this short amount of time and still deliver a focused and well-crafted outcome. This situation forced a very pragmatic response: to avoid all of the time-consuming technical training involved with sensing, computation, actuation, and instead, spend time designing with a focus on human interaction, spatial effect, and part-to-whole relationships. Lately, the architectural discourse seems very concerned with the technical specifics of how physical computing systems work and these are certainly important skills to learn about. However, for beginners, it's more important to ask why we would use these systems, in other words, to what end are these systems put to good use? Often, when introduced to physical computing in courses, students focus on the challenging lower-level technical skills, many times at the expense of the larger design purposes. In some cases, students will compromise on the design ideas solely because of limitations with their ability to write the correct script or to solve the electronics. As these systems scale up, the complexity of deploying them multiplies, and projects rely heavily on teams that bring multiple, interdisciplinary skills and expertise to the table.

Professionally trained designers are in the unique position to orchestrate and integrate systems into a poetic "whole" that is more than the sum of it lower-level parts. A designer's role is to integrate and this is where design adds value to an otherwise reductive process of breaking large problems into smaller problems and then simply solving them. Designers are uniquely aware of human needs as they relate to space, scale, effect, aesthetics, and experience. These topics and concerns should be at the forefront of the students' efforts when discussing physical computing in design, rather than devoting most time to overcoming technical glitches with scripting or electronics. In other words, can we introduce physical computing, without the computing?

Methods

The students were posed with the question: how to design an for interactivity without embedding sensing and computing? The applied projects for this short course involved the design and fabrication of self-assembling spatial lattices. The students worked in small teams and leveraged algorithmic design methods to generate lattice systems. Parallel to the systemic designs, the teams fabricated prototypes and mock-ups in order to work out the component shapes and details.

Lattices

Spatial lattice systems were studied during the short course in order to provide structure and focus to the design problem. Technical definitions of lattice vary between disciplines such as mathematics, biology, crystallography, and engineering. For the purposes of this paper, a lattice describes a three-dimensional array of struts and joints in a repetitive pattern. Lattice systems have inherent advantages that aligned neatly with the objective of creating installation-scaled prototypes, namely structural capacity with even distribution of loads, redundancy of parts that provide a systemic robustness, and repeatable patterns of modules (Chilton 2000).

Algorithmic thinking

Repetition was crucial to the students' understanding of lattice systems and the design problem at hand. Small teams developed simple physical components and devised simple rules for an algorithmic approach to understanding part-to-whole relationships as lattice systems were developed. These algorithmic recipes were tested and ran repeatedly with modeling and parametric design software in order to understand which results were more or less predictable (Figures 1, 2).

In parallel, physical prototyping was carried out, deploying the algorithms step-by-step to ensure that each system's structural and assembly logics worked. These methods were important for devising more or less complex systems from very simple parts and rules. These methods also enhanced the students' understanding of algorithmic systems, dealing with complexity and ambiguity, and exploring the shared dynamic between a designer and the computer (Terzidis, 2006).

Prototyping and feedback

This framework excludes the prototyping of active systems with sensors, wires, microcontrollers, motors, and the like (at least to start with novice students). But as mentioned earlier, iterative physical prototyping was incredibly valuable for experimenting, testing, and simulating lattice components and the resulting systems. In this sense, fabrication (which the students were skilled at) became a useful vehicle for testing parts-to-systems aggregation, details, weight, scale, and effects. Because of extreme time limitations, we focused on repetition of simple components, a limited material palette, and extremely simple electronic components, such as coin cell batteries, copper tape, LEDs. "Tinkering" with more sophisticated components of physical computing systems can come later, and as a result, these systems would be deployed to serve design that is focused on human-user experience.

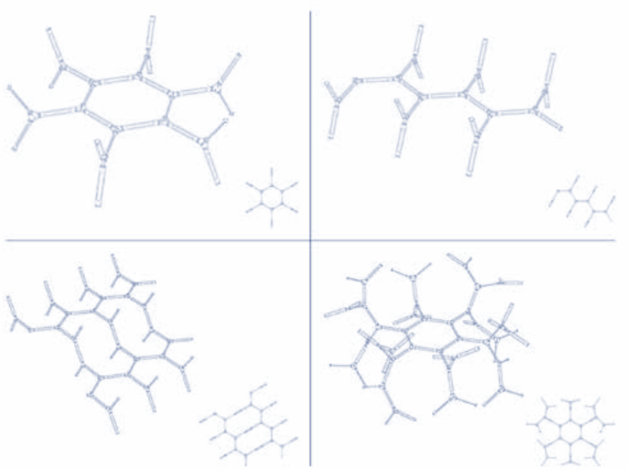

Figure 1: Algorithmic studies of initial lattice systems.

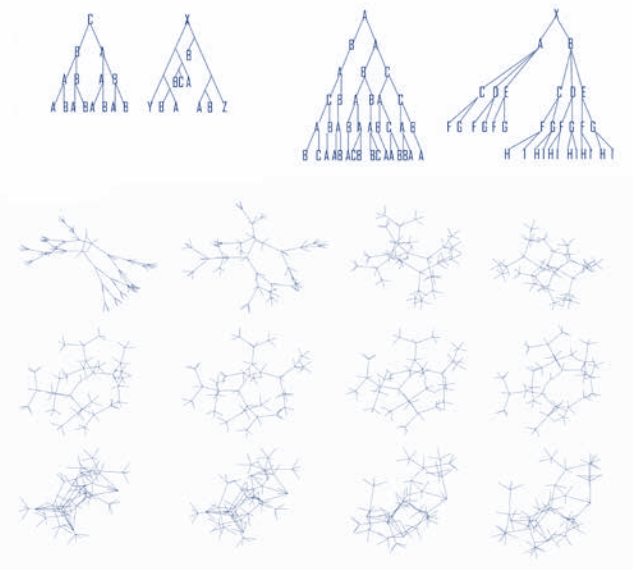

Figure 2: Algorithmic studies for a branching L-system lattice.

Self assembly

From the numerous physical prototypes that were developed, it was decided to experiment with self-assembly—the ability of parts to organize into systems without external directions. Skylar Tibbits' work at MIT on self-assembly in design was a useful resource to better understand how such systems could work and provided the inspiration for using magnets to develop interlocking details for assembly and organization (Tibbits, 2012; Tibbits and Falvello, 2013). Efforts with prototyping focused on the design problem of creating order from disorder through human interaction with the installation. In other words, actuation was achieved through a widely available source of energy—namely human motion. An observer applying energy to one part of the installation creates a disturbance in the field of components that leads to a chain reaction as magnetized components snap together in particular configurations encouraged by the polar directions of the magnets. As components snap together these connections close primitive circuits (made from copper tape) that turn on LED lights, rendering a real-time visual of the phenomenon (Figure 3). This focus on self-assembling lattices allowed the students to experiment with interactive systems without the added challenge of learning more complex skills in engineering, programming, or electrical systems.

Results

The final prototypes, full-sized installations in a gallery space, partly worked but had many problems for further troubleshooting. Integrating self-assembling components into the lattice systems presented many challenges, and these challenges seemingly increased as the scale of each system increased. Nevertheless, when reflecting on the week's events with the students, it was clear that much was learned in a very compressed amount of time, and there were well-crafted artifacts to demonstrate the cumulative effort.

The first installation was derived from an aggregation of acrylic linear struts and tri-directional joints. Each potential joint was magnetized for quick interlocking assembly while suspended in a triangulated grid. The struts were introduced (one-by-one) to form a three-dimensional lattice system (Figures 4, 5).

A second installation was generated algorithmically from a branching variation of a Lindenmayer system. Comprised of acrylic bifurcating joints and struts, the final prototypes physically demonstrated the variability that can be achieved using self-similar component systems (Figure 6).

Discussion

The short course and projects detailed above certainly had limitations in time, and therefore, what could be accomplished. However, these limitations became important constraints that forced a prioritization of the design of human experience, interaction, and effect over mechanical and electrical systems, programming, and other technical issues outside the traditional domain of architectural knowledge. The apparent value of this framework extends beyond its highly situational origins. These circumstances have led to a fundamental rethinking of my own pedagogical aims when teaching students who are new to the topic of physical computing—from teaching and practicing highly technical skills—to focusing on ideas for interactions between humans and the built environment (regardless of whether physical computing systems are deployed, or not). Certainly, there is value in learning some of the more specialized skills in working with code, electronics, and mechanical systems. But there is even more value in being able to work effectively with other professionals with these skills (engineers, programmers, electricians), and within such interdisciplinary teams, architects and other professional designers provide the most value by integrating all of these skills and expertise to shape meaningful and memorable spatial and interactive experiences.

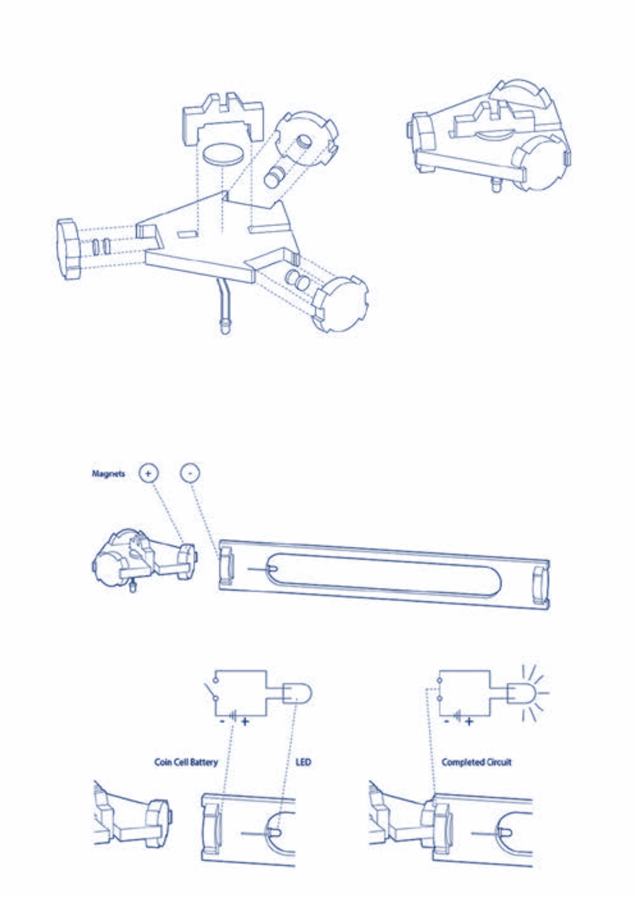

Figure 3: Joint, magnet, and battery circuit schematic details.

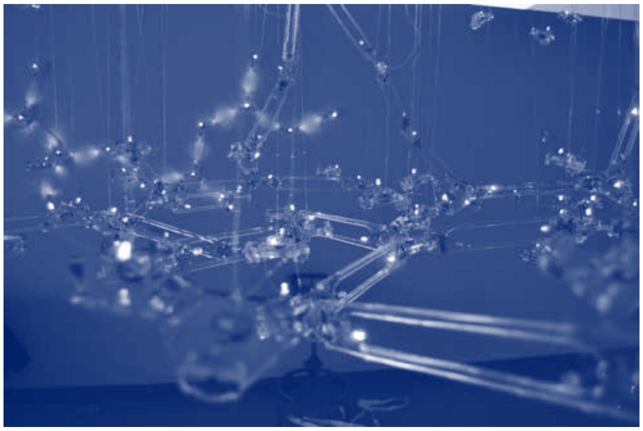

Figure 4: Close-up of first lattice installation.

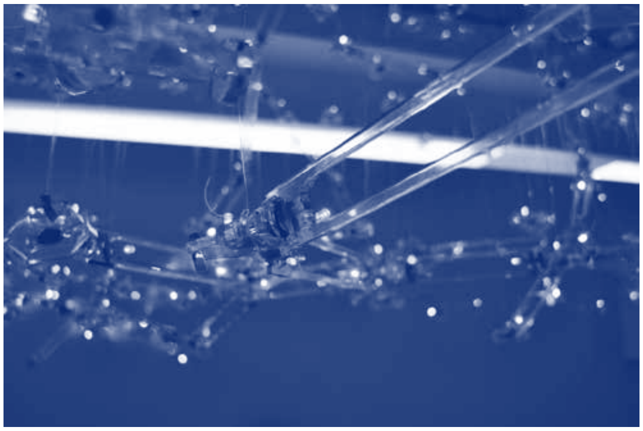

Figure 5: Details from first lattice installation once deployed.

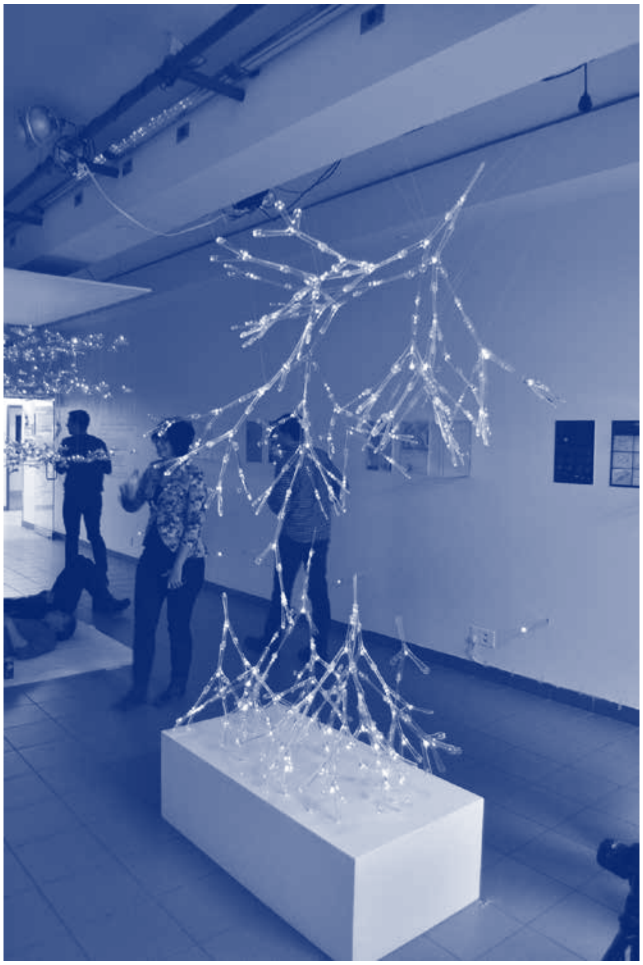

Figure 6: Second lattice installation based on Lindenmayer system.

Acknowledgments

The work shown was produced in a four-day workshop conducted by the author as part of the annual Taylor Seminar in the Faculty of Environmental Design at the University of Calgary. Many thanks are given to assistant professor Jason Johnson and the students that participated: Mehrdad Amjadi, Michael Chu, Nic Dykstra, Meysam Ehsanian, Daniel Farid, Alyssa Haas, Kendra Kusick, Joanna Long-Tieu, Matt Marrotto, Jamie Lynne McFadyen, MacKenzie Nixon, Obinna Martins, Shane Oleksiuk, Sadaf Rabbani, Matt Stewart, and Sabrina Vastag. Assistance with prototyping and graphics was provided by Ludwing Vaca, graduate assistant at UNLV School of Architecture.

References

Beesley, P. (2012). Soil and Protoplasm: Designing the Hylozoic Ground Component System". in B. Sheil (Ed.), Manufacturing the Bespoke: Making and Prototyping Architecture, Architectural Design Reader (pp. 102-119). London, UK: Wiley.

Buente, A., Perry, K. (2013). "Temporal Synapse". PROJECTiONE. Retrieved from http://projectione.com/temporal-synapse/.

Chilton, J. (2000). Space Grid Structures. Oxford: Architectural Press.

Igoe, T., O'Sullivan, D. (2004). Physical Computing. Boston, MA: Thomson.

Senagala, M., Vermillion, J. (2009). An Inconvenient Studio. Proceedings of the 29th Annual Conference of the Association for Computer Aided Design in Architecture (pp. 287-290). Chicago, 22-25 October.

Terzidis, K. (2006). Algorithmic Architecture. Oxford: Architectural Press.

Thune, G., Velikov, K., O'Malley, M., Sauve, L. (2012). The Agency of Responsive Envelopes: Interaction, Politics and Interconnected Systems. International Journal of Architectural Computing, Vol. 10, No. 3, 377-400.

Tibbits, S. (2012). Design to Self-assembly. Material Computation: Higher Integration in Morphogenetic Design, Architectural Design 82, 68-73.

Tibbits, S., Falvello, A. (2013). BioMolecular, Chiral and Irregular Self-Assemblies. Proceedings of the 33rd Annual Conference of the Association for Computer Aided Design in Architecture (pp. 267-268). Cambridge, ON. 24-26 October.