The Design and Optimization of a Lighted Kinetic Art Surface Display

(Technical Paper)

An ANT Analysis of Open-Source Development

(STS Paper)

A Thesis Prospectus Submitted to the

Faculty of the School of Engineering and Applied Science
University of Virginia • Charlottesville, Virginia

In Partial Fulfillment of the Requirements of the Degree
Bachelor of Science, School of Engineering

Philip Renkert
Fall, 2019

Technical Project Team Members
Megan Mazzatenta
Jack Purcell
Philip Renkert

On my honor as a University Student, I have neither given nor received unauthorized aid on this assignment as defined by the Honor Guidelines for Thesis-Related Assignments

Introduction

The term “technology” elicits thoughts of hardware, software, and technical design. Yet beneath the surface of everything from cars to laptops to social media lies a complex sociotechnical network. As these technologies develop from concept to fully integrated product, they shape and are shaped by their societal context. The proposed STS research focuses on open source technology development, a relatively new concept that emerged with the invention of the internet and has grown to power much of today’s modern technology (Finley, 2016). Open source is broadly defined as “a philosophy that promotes the free access and distribution of an end product” (Technopedia, n.d.). This philosophy applies to software by opening the source code and to hardware by opening the designs, bill of materials, and assembly instructions. The open source model distributes technological development among a large community and assumes a fluid structure that allows for asynchronous contributions, continuously integrates new features, and prioritizes robust documentation and collaboration (Ibrahim & Warner, 2011). In his book *Two Bits: The Cultural Significance of Free Software*, author Christopher M. Kelty notes how the open source movement has “emerged in tandem with the Internet as both a technical and a social form” (Kelty, 2008). The technical and social aspects of open source are tightly interconnected. The communities that form around open projects are societies themselves, comprised of individuals and organizations with complementary skills and motivations that align to create a product. Modern technologies, however, connect the members of these communities and enable collaboration. Therefore, open source technology development cannot be understood from a solely social or technological perspective, and this understanding is crucial given the rapid growth and potential impact of the open source movement.

The second proposed project integrates technical and social elements and aligns with the tenets of open source hardware. The research team is designing a kinetic art display that users can program to obtain a physical representation of a surface or image. The design consists of a grid of transparent rods with colored lights attached to create an interesting visual effect. A servo motor coupled to each rod actuates its linear motion. On the technical side, the design is optimized to reduce cost, complexity, and manufacturing time. Components that cannot be constructed with common digital manufacturing tools like 3D printers and CNC machines are easily available off-the-shelf. The Parallax, Inc. microcontroller and software which power the design are open source (Parallax Inc, n.d.). These technical elements increase the feasibility of completing a finished project, but they also make it easy for others to replicate, modify, and share the design as per the open-source philosophy. On a social note, the intent of the design is to create a visually appealing piece that demonstrates the potential beauty of mechatronics and inspires viewers to join the community. We hope to demonstrate that STEM education need not be confined to a classroom, as open design principles allow great engineers and makers to develop simultaneously with great products.

Technical Topic

The capstone team, comprised of Megan Mazzatenta, Jack Purcell, and Philip Renkert, identified the need for a kinetic, programmable display that demonstrates mechatronic principles and increases excitement about Mechanical Engineering. From a design perspective, the iterative development of space-efficient and optimized systems is a concept integral to both engineering design and business decisions yet not typically covered in engineering curricula. The kinetic display will emphasize the importance of design optimization for minimizing manufacturing cost and time and create a lasting addition to the Mechanical Engineering (MEC) building.

The kinetic sculpture will consist of many small, lighted units, aptly named Voxels, that translate up and down to form surfaces or display images. The initial concept is inspired by the kinetic sculptures developed for the Bavarian Motor Works (BMW) Museum in Munich (ART+COM Studios, 2018), as well as the Build UP LLC design piece in Dubai (Build UP LLC, 2015). These designs use string-and-pulley systems to lower metal and LED-lit spheres (see Figure 1a), respectively, which move in pre-programmed patterns to create surfaces, waves, and designs. These string-and-pulley systems, though simple and effective, are constrained by several limitations. First, because they rely entirely on gravity to lower the hanging spheres, they only work in one orientation, with the objects hanging beneath the pulley. Furthermore, because the hanging spheres are only constrained in one direction, nothing prevents the spheres from swinging or getting tangled. Therefore, these designs must be isolated from wind and other environmental factors.

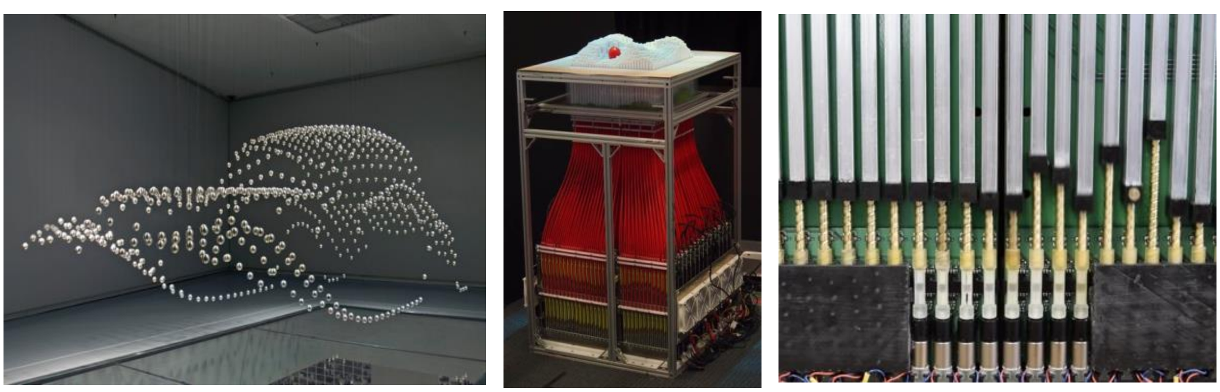

Figure 1(a-c). Representations of prior art for extendable surface tables and suspensions. From left to right: (a)BMW Museum kinetic sculpture with hanging metal spheres, (b)MIT inFORM table with pin-array structure, and (c) Stanford shapeShift table with screw drive.

However, the pin array configuration mitigates many of these issues. Researchers at the Massachusetts Institute of Technology (MIT) pioneered this configuration. Their Tangible Media Group has conceived of numerous designs, including Relief and inFORM (see Figure 1b), which use an array of sliding pins to create a digitally-configurable surface (Follmer, Leithinger, Olwal, Hogge, & Ishii, 2013). Stanford has a similar configuration in their shapeShift project (Siu, Gonzalez, Yuan, Ginsberg, & Follmer, 2018). The MIT design utilizes a compact linear actuator coupled with a slide potentiometer for feedback, while Stanford's design uses a screw drive (see Figure 1c). Both designs are limited by cost, difficulty of assembly, and limited linear travel for each pin.

The team used these limitations to form considerations aimed to develop a similarly-captivating visual effect in the project and chose to design a rigid lighted structure in order to meet the project goals. A rigid structure allows the motor to both raise and lower each unit without relying on gravity for lowering, so the display can theoretically operate the same way when installed in any direction. When compared to string, the rigid structure also discourages students passing through the MEC Building from reaching up to swat at the display. The system must be modular, inexpensive, and easily manufacturable to feasibly build a display of any size and orientation. Accordingly, the team focused on optimizing one unit of the design, and major design decisions were motivated by the objectives of minimizing both cost and manufacturing time while maintaining a captivating visual effect.

Overall, the optimized unit will create a versatile, mystifying visual display that can be expanded or modified with ease. Motion of the lighted acrylic rod will be large and controlled, and incorporating the motor into each unit provides a key advantage in modularity. If one unit breaks, or the user desires a larger display, units can be easily replaced or added on with minimal cost and manufacturing time. The inverted kinetic sculpture will be an attention-grabbing addition to the MEC building, and the team aims to give the display a greater purpose by allowing students to program in their own functions and use the sculpture as a fun way to visualize surfaces, waves, vibrations, etc. The project is set to be completed in the Fall 2019 semester with a budget of 20 dollars per Voxel, and future work on the project will focus on expanding user interaction with the kinetic sculpture.

STS Topic

To understand the open-source development (OSD) technology and its future implications for society, we must first examine its history. As shown in Figure 2, one can trace the roots of OSD back to the mid-20th century, when the advent of computers prompted universities to develop and share software (Longsight, n.d.). This early software, therefore, was entirely open source as universities worked together to harness the power of the computer. In the 1970s, this collaborative development gave rise to Unix, a popular operating system upon which many commercial startups were founded.

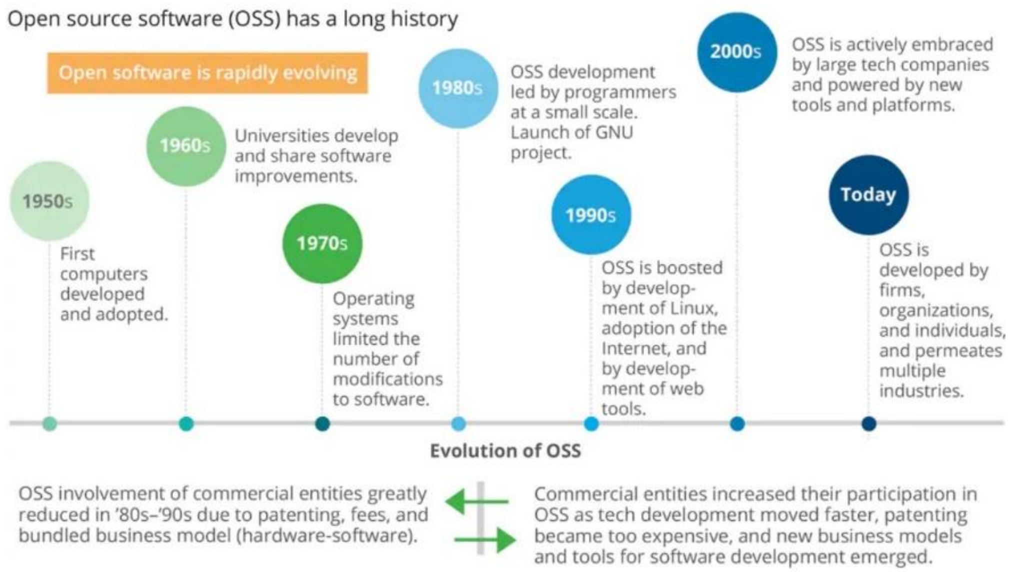

Figure 2. History of Open Source *(Jesus Leal Trujillo, Steve Fromhart, & Val Srinivas, 2017)*.

Open software experienced a resurgence in the 1980s, when Massachusetts Institute of Technology (MIT) researcher Richard Stallmann founded the Free Software Foundation and created a theoretical foundation for open-source software (Alexander Hars, 2002). Stallmann also launched the GNU community development effort, which aimed to develop a free alternative to UNIX. In 1991, Linus Torvalds began his work on the canonical example of open-source software (OSS): the Linux operating system (Welsh, Kaufman, & Dalheimer, 1999). Torvalds combined elements of UNIX and GNU and released his work to the internet community, creating a snowball of development that led to Linux as we know it today. Shortly thereafter, organizations developed to represent open-source communities and advocate and license open-source projects. These organizations also provide definitions of terms surrounding OSD, securing the cogency of open-source licenses. The Open Source Initiative provides ten defining criteria for software distributed with an open-source license:

1. Free Redistribution
2. Source Code
3. Derived Works
4. Integrity of the Author's Source Code
5. No Discrimination Against Persons or Groups
6. No Discrimination Against Fields of Endeavor
7. Distribution of License
8. License Must Not Be Specific to a Product
9. License Must Not Restrict Other Software
10. License Must Be Technology-Neutral

(Open Source Initiative, n.d.)

The lesser known cousin of OSS is open-source hardware (OSH). OSH derives its principles from OSS: "Open source hardware is hardware whose design is made publicly available so that anyone can study, modify, distribute, make, and sell the design or hardware based on that design" (Open Source Hardware Association, 2012). OSH developed naturally from early "makers" and hacking communities that had "long traditions of knowledge-sharing practices" (Gibb, 2014). Like with OSS, the rise of the internet allowed these communities and designs to spread far beyond geographical boundaries. Unlike OSS, OSH requires physical materials to bring community designs into reality. OSH communities encourage designers to incorporate readily-available components and materials and open-source tools and machines to maximize an individual's ability to make the hardware (Open Source Hardware Association, 2012). While OSS gained momentum, advances in digital manufacturing technologies largely mitigated the difficulties of turning digital designs into physical products. Machines like 3D printers and CNC mills "allow for the creation of unique, one-off objects without significant setup time or tooling costs" (Mellis & Buechley, 2011).

Actor Network Theory (ANT) provides an excellent framework to analyze the dynamic and complex networks surrounding open-source development. ANT was largely penned by scholars Michel Callon, Bruno Latour, and John Law, who laid the groundwork for ANT in the late 20th century (Darryl Cressman, 2009). *Learning Theories* provides a concise definition: "a framework and systematic way to consider the infrastructure surrounding technological achievements. Assigns agency to both human and non-human actors" (David L., 2007). Several attributes of ANT make it an ideal candidate for an analysis of OSD. The theory is applicable to the complex networks surrounding OSD, and it easily accommodates human and nonhuman actors through its principle of generalized symmetry: "An actor in ANT is a semiotic definition – an actant – that is something that acts or to which activity is granted by another…an actant can literally be anything provided it is granted to be the source of action" (Latour, 1996). An ANT network is dynamic. Its toolbox contains Translation to examine sociotechnical processes and

Punctualization to collapse static subnetworks into “black boxes” for a clearer view of the larger network (Darryl Cressman, 2009).

One key to successful open-source development is the diverse community of actors participating in the network (Bonaccorsi & Rossi, 2003). The Linux Foundation cites the following human actors: Leaders, Maintainers, Committers, Contributors, and Users (The Linux Foundation, n.d.). The network of an open-source project also consists of many non-human actors. Some are obvious, such as computer hardware and the internet. In a recent presentation, Mark Gisi of Wind River Systems pointed to code repositories such as GitHub and open-source licenses as nonphysical artefacts that play critical roles in open-source networks (Gisi, 2017). OSH projects include these artefacts as well as manufacturing tools, manufacturing locations, and component supply chains. Finally, the nature and structure of the project itself is a central actor in both OSS and OSH project networks. For example, projects must accommodate distributed collaboration so that contributors can asynchronously contribute to small pieces of the project.

Despite its capabilities, ANT applied in isolation would make for an incomplete analysis of OSD. One point of contention is that ANT insists on the agency of nonhumans; some critics maintain that properties such as intentionality or creativity are distinctly human and cannot be attributed to nonhuman actors. In a paper in the journal *Social Studies and Science*, Edwin Sayes points to several writers with this contenting stance. Collins and Yearly argue that “social scientists are ‘not particularly good’ at coming to terms with the competencies of nonhumans and, thus, should leave such an analysis to natural scientists and engineers” (Sayes, 2014). Other writers referenced by Sayes include Khong, Riis, Amsterdamska, and Scaffer (Sayes, 2014). Another criticism of ANT, made by Hans Radder in his “Normative Reflexions on Constructivist

Approaches to Science and Technology," is that it relies heavily on observations and case studies, leading to situations where researchers simply report what they find without recognition of general elements like norms and values (Radder, 1992). David Bloor, a fervent critic of ANT, suggests that ANT neither gives proper weight to non-social elements and processes nor acknowledges their contribution to social arrangements (Bloor, 1999). ANT has also been criticized for being amoral and for failing to account for pre-existing structures or power asymmetries.

The benefits of open-source development and its rapid diffusion into consumer, academic, and industrial spheres warrant further research of the topic. Time has shown that open-source products are superior to closed alternatives in that they offer higher quality and better reliability. Generally, this is attributable to "increased efficiencies the open-source development model offers to large, distributed teams working on major software projects" (Ibrahim & Warner, 2011), as well as benefits from mass scale peer review and collaboration (Pearce, 2014). Open source is growing rapidly and diffusing into industry. Companies like Red Hat, MuleSoft, and MongoDB, all based on open-source projects, are now valued in the millions of dollars and continue to grow (Volpi, 2019). Companies are also adopting open-source projects internally as opposed to "reinventing the wheel" with their own developers or shackling themselves to proprietary business products (Illuminas, 2019). Linux, which started out as a hobby, now powers 67% of all web servers and provides the backbone for Android, televisions, thermostats, and even cars (Finley, 2016). It is imperative that society understand a movement that has gained so much momentum in such a short amount of time. Only by understanding the underlying mechanisms of the open-source system can society learn to predict and control it. A solely technological perspective is insufficient; open-source development is a complex network of sociotechnical actors who interact to create a technology and a movement far greater than the sum of its parts.

Research Question and Methods

The STS research proposes the following research question: "How do heterogeneous actors contribute to the success of open-source projects?" The method will combine network analysis of a case study with documentary research of other open-source projects in order to generalize the results beyond the specific case. A common criticism of ANT is its narrow perspective of the case being analyzed. Integration with surrounding research allows the discussion to "zoom out" from the specific case and view actants and their attributes in the general case of open-source projects.

Though many projects could be chosen for the case analysis, the Arduino project dovetails nicely with the purpose of this research. Arduino is an inexpensive and easily-programmable microcontroller board that makes DIY electronics available to the masses (Kushner, 2011). The project started as a simple educational tool in 2005 has matured to become a stable company (Kushner, 2011). It has a vibrant and growing community that includes not only hackers and hobbyists but also educators, researchers, industry, and derivative products. Because Arduino is an example of OSH, results of the analysis will complement similar studies on OSS. The documentary research used to generalize the results of the case study includes general analyses of open-source networks as well as case studies of specific OSS projects. The former category includes "Open source enters the world of atoms: A statistical analysis of open design" by Kerstin et al., which provides a quantitative study of open design projects (Balka, Raasch, & Herstatt, 2009); *Two Bits: The Cultural Significance of Free Software* by Kelty, which explores the social and cultural implications of open software (Kelty, 2008); and

“Understanding Sustained Participation in Open Source Software Projects” by Fang and Neufeld, which seeks to understand why developers contribute to OSS projects in the long run (Fang & Neufeld, 2009). The latter category includes “Internet, Innovation, and Open Source: Actors in the Network” by Tuomi, which analyzes the Linux community (Tuomi, 2001); as well as “Lost and Gained in Translation: Adoption of Open Source Software Development at Hewlett-Packard” by Melian and Mähring, which looks at the integration of OSS into Hewlett-Packard’s proprietary software model (Melian & Mähring, 2008).

Conclusion

The technical deliverable will consist of a single optimized unit that aligns with the principles of open-source hardware design. When combined in a grid, these units create a dynamic surface that is simple, has a longer linear range than prior art, and need not be isolated from its environment. A detailed technical report outlining the design process, bill of materials, and assembly instructions will accompany the display. The STS deliverable will explore how heterogeneous actors contribute to the success of open-source projects by analyzing the complex networks surrounding them. By applying Actor Network Theory and network analysis to a specific case and comparing those results to analyses of other open-source projects, I intend to identify and describe general actors in a project’s network whose combined effect enables the project’s success. Prior research largely focuses on OSS projects, which have quickly come to dominate the software industry and have a large effect on our daily lives. Should this trend of open-source development continue to grow and diffuse into hardware and other sectors, it is important that society understand its general mechanisms.

References

Alexander Hars, S. O. (2002). Working for Free? Motivations for Participating in Open-Source Projects. *International Journal of Electronic Commerce*, *6*(3), 25–39. https://doi.org/10.1080/10864415.2002.11044241

ART+COM Studios. (2018). Kinetic Sculpture. Retrieved October 21, 2019, from https://artcom.de/en/project/kinetic-sculpture/

Balka, K., Raasch, C., & Herstatt, C. (2009). Open source enters the world of atoms: A statistical analysis of open design. *First Monday*, *14*(11). https://doi.org/10.5210/fm.v14i11.2670

Bloor, D. (1999). Anti-Latour. *Studies in History and Philosophy of Science Part A*, *30*(1), 81–112. https://doi.org/10.1016/s0039-3681(98)00038-7

Bonaccorsi, A., & Rossi, C. (2003). Why Open Source software can succeed. *Research Policy*, *32*(7), 1243–1258. https://doi.org/10.1016/S0048-7333(03)00051-9

Build UP LLC. (2015). *Kinetic Sculpture Timelapse*. Retrieved from https://www.youtube.com/watch?v=ICixCazf6-k

Darryl Cressman. (2009). *A Brief Overview of Actor-Network Theory: Punctualization, Heterogeneous Engineering & Translation*. Retrieved from School of Communication, Simon Fraser University website: http://faculty.georgetown.edu/irvinem/theory/Cressman-ABriefOverviewofANT.pdf

David L. (2007, March 23). Actor-Network Theory (ANT). Retrieved October 13, 2019, from Learning Theories website: https://www.learning-theories.com/actor-network-theory-ant.html

Fang, Y., & Neufeld, D. (2009). Understanding Sustained Participation in Open Source Software Projects. *Journal of Management Information Systems*, *25*(4), 9–50. https://doi.org/10.2753/MIS0742-1222250401

Finley, K. (2016, August 25). Linux Took Over the Web. Now, It’s Taking Over the World. *Wired*. Retrieved from https://www.wired.com/2016/08/linux-took-web-now-taking-world/

Follmer, S., Leithinger, D., Olwal, A., Hogge, A., & Ishii, H. (2013). inFORM: Dynamic Physical Affordances and Constraints Through Shape and Object Actuation. *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology*, 417–426. https://doi.org/10.1145/2501988.2502032

Gibb, A. (2014). *Building Open Source Hardware: DIY Manufacturing for Hackers and Makers*. Retrieved from https://proquest.safaribooksonline.com/9780133373912

Gisi, M. (2017, April). *Leveraging the Open Source Development Model Inside Your Company*. Retrieved from https://www.youtube.com/watch?v=xfMTE74y2Ig&feature=share

Ibrahim, H., & Warner, B. (2011). *Understanding the Open Source Development Model*.

Illuminas. (2019). Enterprise Open Source Report 2019. Retrieved October 17, 2019, from https://www.redhat.com/en/enterprise-open-source-report/2019

Jesus Leal Trujillo, Steve Fromhart, & Val Srinivas. (2017). *The evolution of blockchain technology | Deloitte Insights*. Retrieved from https://www2.deloitte.com/us/en/insights/industry/financial-services/evolution-of-blockchain-github-platform.html

Kelty, C. M. (2008). *Two Bits: The Cultural Significance of Free Software*.

Kushner, D. (2011, October 26). The Making Of Arduino. Retrieved October 22, 2019, from IEEE Spectrum: Technology, Engineering, and Science News website: https://spectrum.ieee.org/geek-life/hands-on/the-making-of-arduino

Latour, B. (1996). On actor-network theory: A few clarifications. *Soziale Welt*, *47*(4), 369–381. Retrieved from JSTOR.

Longsight. (n.d.). History of Open Source. Retrieved October 15, 2019, from https://www.longsight.com/learning-center/history-open-source

Melian, C., & Mähring, M. (2008). Lost and Gained in Translation: Adoption of Open Source Software Development at Hewlett-Packard. In B. Russo, E. Damiani, S. Hissam, B. Lundell, & G. Succi (Eds.), *Open Source Development, Communities and Quality* (Vol. 275, pp. 93–104). https://doi.org/10.1007/978-0-387-09684-1_8

Mellis, D. A., & Buechley, L. (2011). Scaffolding Creativity with Open-source Hardware. *Proceedings of the 8th ACM Conference on Creativity and Cognition*, 373–374. https://doi.org/10.1145/2069618.2069702

Open Source Hardware Association. (2012, May 26). Definition of Open Source Hardware. Retrieved October 15, 2019, from Open Source Hardware Association website: https://www.oshwa.org/definition/

Open Source Initiative. (n.d.). The Open Source Definition | Open Source Initiative. Retrieved October 15, 2019, from https://opensource.org/docs/osd

Parallax Inc. (n.d.). Open Source Designs. Retrieved October 21, 2019, from https://www.parallax.com/support/open-source-designs

Pearce, J. M. (2014). *Quantifying the Value of Open Source Hardware Development* (SSRN Scholarly Paper No. ID 3331131). Retrieved from Social Science Research Network website: https://papers.ssrn.com/abstract=3331131

Radder, H. (1992). Normative Reflexions on Constructivist Approaches to Science and Technology. *Social Studies of Science*, *22*(1), 141–173. https://doi.org/10.1177/0306312792022001009

Sayes, E. (2014). Actor–Network Theory and methodology: Just what does it mean to say that nonhumans have agency? *Social Studies of Science*, *44*(1), 134–149. https://doi.org/10.1177/0306312713511867

Siu, A. F., Gonzalez, E. J., Yuan, S., Ginsberg, J. B., & Follmer, S. (2018). shapeShift: 2D Spatial Manipulation and Self-Actuation of Tabletop Shape Displays for Tangible and Haptic Interaction. *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, 291:1–291:13. https://doi.org/10.1145/3173574.3173865

Technopedia. (n.d.). What is Open Source? Retrieved October 28, 2019, from Techopedia.com website: https://www.techopedia.com/definition/3294/open-source

The Linux Foundation. (n.d.). Open Source Communities. Retrieved October 19, 2019, from The Linux Foundation website: https://www.linuxfoundation.org/resources/open-source-guides/participating-open-source-communities/

Tuomi, I. (2001). Internet, innovation, and open source: Actors in the network. *First Monday*, *6*(1). https://doi.org/10.5210/fm.v6i1.824

Volpi, M. (2019, January 12). How open-source software took over the world. Retrieved October 17, 2019, from TechCrunch website: http://social.techcrunch.com/2019/01/12/how-open-source-software-took-over-the-world/

Welsh, M., Kaufman, L., & Dalheimer, K. (1999). A Brief History of Linux. In *Running Linux, Third Edition* (Third). Retrieved from https://learning.oreilly.com/library/view/running-linux-third/156592469X/ch01s02.html