Digital Studies / Le champ numérique

Burrell, Andrew. 2023. "Towards Accessible, Open, and Ethical Design-Led Digital Humanities Visualization Projects on the Web," in "DH Unbound 2022, Selected Papers," ed. Barbara Bordalejo, Roopika Risam, and Emmanuel Château-Dutier, special issue. *Digital Studies/Le champ numérique* 13(3): 1–14. DOI: https://doi.org/10.16995/dscn.9597.

Open Library of Humanities

Towards Accessible, Open, and Ethical Design-Led Digital Humanities Visualization Projects on the Web

Andrew Burrell, Faculty of Design, Architecture, and Building, University of Technology Sydney, AU, andrew.burrell@uts.edu.au

This paper is an exploration of the design and material considerations for creating design-led digital humanities information visualizations using open and accessible web-based tools. It considers how we might approach or even re-approach the high-level material infrastructures of the web (network/code/browser) to facilitate these goals of openness and accessibility—what could perhaps be called a series of meta-considerations when developing visualization projects for the web. It emphasizes code as a material of design—a material that is argued should not always be the exclusive domain of software developers and engineers but can instead be an open and accessible material for, in this case, building dynamic immersive information visualizations. The aim of thinking of code as material in this way is to allow for the possibility of web-based projects being easily accessed and extended by others. It uses the Glossopticon VR visualization as an example throughout.

Cet article est une exploration des considérations de conception et de matériel pour créer des visualisations d'information sur les humanités numériques dirigées par la conception en utilisant des outils Web ouverts et accessibles. Il examine comment nous pourrions aborder ou même réapprocher les infrastructures matérielles de haut niveau du web (réseau/code/navigateur) pour faciliter ces objectifs d'ouverture et d'accessibilité—ce que l'on pourrait peut-être appeler une série de méta-considérations lors de l'élaboration de projets de visualisation pour le web. Il met l'accent sur le code en tant que matériau de conception—un matériau qui, selon nous, ne devrait pas toujours être le domaine exclusif des développeurs de logiciels et des ingénieurs, mais qui peut au contraire être un matériau ouvert et accessible pour, dans ce cas, construire des visualisations d'informations dynamiques et immersives. L'objectif de considérer le code comme un matériau de cette manière est de permettre aux projets basés sur le web d'être facilement accessibles et étendus par d'autres. La visualisation Glossopticon VR est utilisée comme exemple tout au long de l'ouvrage.

Introduction

This paper is an exploration of the design and material considerations for creating design-led digital humanities information visualizations using open and accessible web-based tools. It considers how we might approach or even re-approach the high-level material infrastructures of the web (network/code/browser) to facilitate these goals of openness and accessibility—what could perhaps be called a series of meta-considerations when developing visualization projects for the web. It emphasises code as a material of design—a material that is argued should not always be the exclusive domain of software developers and engineers but can instead be an open and accessible material for, in this case, building dynamic immersive information visualizations. The aim of thinking of code as material, in this way, is to allow for the possibility of web-based projects being easily accessed and extended by others.

This is particularly true of web-based projects which have a primary use case as tool for research and information visualization—and where users have a stake in the potential for expansion of the visualization to meet their own needs—though this should not be read as the only use case for where it is relevant. This paper also asks what the process of developing design-led digital humanities visualizations on the web (as practiced by design/digital humanities professionals) can learn from earlier (what might now be called web.10), ways of approaching web projects, as well as what might be learnt from alternate, activist, creative or other outlier led communities of practice in the web/code space.

In many ways, this reverses the scholar's burden presented by N. Katherine Hayles when she states, "The necessity for executable code creates new requirements for digital literacy. Not every scholar in digital humanities needs to be an expert programmer, but to produce high-quality work, scholars certainly need to know how to talk to those who are programmers" (Hayles 2012, 42). This reversal suggests that through the way they write and present their code, programmers and coders (who are potentially also scholars) can create material outcomes that can speak directly to the digitally literate.

While this paper deals directly with front-end web-based projects, many of the considerations are transposable to other projects developed with code as a primary material.

Glossopticon

This paper will use the "Glossopticon" project as a case study. The Glossopticon VR project is a project by Andrew Burrell (virtual environment and data visualization design/development, University of Technology Sydney), Rachel Hendery (conception

and linguistics, Western Sydney University), and Nick Thieberger (linguistics and data, Melbourne University). The project was funded by the Centre of Excellence for Language Dynamics (COEDL), through a Transdisciplinary and Innovation Grant. This paper centres its material underpinnings. The interested reader will find a broader description of this project and its aims in Burrell, Hendery, and Thieberger (Burrell, Hendery, and Thieberger 2019) and on the project website (http://glossopticon.com).

Glossopticon (see **Figure 1**) is a virtual reality-based information visualization system that presents archived heritage audio and related metadata in an immersive spatial environment. A central aim of Glossopticon is to visualize (and sonify) the PARADISEC collection (PARADISEC 2022), thus creating a new access point into an archival database, which holds records of 1,202 languages of the Pacific region, many of which include audio recordings. There are close to 9,700 hours of recordings in PARADISEC, and from these recordings, we extracted short snippets with links back to the full content in the existing PARADISEC web portal. The process of creating these snippets and some of the implications of choosing to extract these using a semi-automated process is described in the paper "Nura Yaman ('Country Speaks'): Language, People and Place in Serious Games" (Burrell, Hendery, and Hromek 2021).

Figure 1: Glossopticon VR Visualization—Game engine version.

Glossopticon provides a means for both expert and non-expert users of the visualization to explore this vast amount of archival material by spatially locating

(in a virtual environment) a combination of metadata from the catalogue and related audio snippets. This metadata was also supplemented with data drawn from the "Open Language Archives Community" (OLAC) visualizer, which contains information about how much archival material linguists have for any given language (OLAC 2011).

Glossopticon has taken on a number of different forms. Initially it was developed in the Unity (https://unity.com/) game engine and then as a web-based visualizations for sharing information with a wider audience—as a toolset for researchers to work with visualizing linguistic data spatially. These two versions of the project represent two different approaches to design-led, digital humanities visualization—the polished complete platform and the more open-ended and slightly "messier" platform inviting a user to interact not just with the visualization, but with the underlying code as material of the platform.

Glossopticon has recently been redeveloped (2021–2022), taking into mind the central considerations of this paper—openness, accessibility, and attention to the ethics of technology. It is to this redeveloped version of the project that this paper will speak to as a case study. This redevelopment is focused on an audience who has an investment in the information being visualized and who want to be able to interrogate the platform itself in order to understand how the visualization has been created and hence validate (or otherwise) the visualization against their own domain knowledge.

This taking into mind of openness, accessibility, and attention to the ethics does not offer full solutions, but rather an approach to working towards rediscovering the web as a platform that allows the end audience to also be part of a dynamic framework that allows them to take the role of participants in a dynamic network. This role is very much one that was imagined in the web's early days, for example, as can been seen in Berners-Lee's statement: "The web is more a social creation than a technical one. I designed it for a social effect—to help people work together—and not as a technical toy. The ultimate goal of the Web is to support and improve our weblike existence in the world" (Berners-Lee and Fischetti 1999, 123). In order for this to be the case, users of the web cannot just be consumers of information, but creators, aggregators, and curators of information as content. While an extended discussion on this point cannot be made here, it is important to note that this social and creative user of the early web is a very different proposition from what we now have with platform based social media.

Development ecosystems

One of the main reasons for developing the web-based version of Glossopticon is that the version developed in the game engine tended to be a closed system, a finished product, a piece of software that was excellent for a user who simply wanted to view

the data, but not necessarily manipulate and validate the visualization and the data it contained or even import data of their own. In doing this, Glossopticon attempts also to address the fact that there is a significant barrier to entry in using a game engine, which does not need to exist in a web-based project. In order to do this well, however, it takes some consideration of how we make a web project open and accessible for it to actually be so. Simply put, one of the central aims of the considerations presented here is that a web-based project can be examined, interacted with, and manipulated in a web browser in both its "front facing outcome" form and in the form of its material construction through code.

Developing web-based applications for design-led, digital humanities-based research has become increasingly and overly complex. The practice of working with web code (as a material underpinning of project outcomes) is often impenetrable and exists within layers of frameworks, jargon, and gatekeeping of "the right way" of doing something. This will be discussed in detail in the discussion to follow. For now, it is sufficient to point out that contemporary trends in web development are a long way from early visions of user-accessible, human-readable approaches to coding for the web. It is important to note that HTML (the markup language of the web), as developed by Berners-Lee, was by original design not intended to be "hand coded." In fact, the original intention was for more of a WSIWIG form of editing of the web. Berners-Lee makes this point:

> I never intended HTML source code (the stuff with the angle brackets) to be seen by users. A browser/editor would let a user simply view or edit the language of a page of hypertext, as if he were using a word processor. The idea of asking people to write the angle brackets by hand was to me, and I assumed to many, as unacceptable as asking one to prepare a Microsoft Word document by writing out its binary coded format. But the human readability of HTML was an unexpected boon. To my surprise, people quickly became familiar with the tags and started writing their own HTML documents directly. (Berners-Lee and Fischetti 1999, 42)

The point being that we have moved a long way from this original human-readable simplicity, and, from an end-user point of view, contemporary approaches to web development often add layers of unnecessary over-complexification. How we might work against this is an ongoing and central concern of this author as they continue to design, implement, and share immersive visualizations on the web and beyond. In the context of this paper, it is useful to ask for whom the coding for the web is over-complex and impenetrable (a software developer would surely disagree with this statement). The audience being spoken of here is an end user who is interested and invested in

understanding both the material being visualized and in how it is visualized—the previously mentioned digitally literate scholar. This user requires transparency and the ability to scrutinize and interrogate what they are seeing, as both an outcome and the material and conceptual processes that went into creating this outcome.

If we frame this question of who this end user is along the lines of what Nancy Mauro-Flude and Yoko Akama call "reimagined Internet futures" (Mauro-Flude and Akama 2022), then we can start to think about what sort of things we should be taking into mind when developing visualizations for researchers on web platforms and beyond. "Reimagined Internet futures" require us to take a wider look at how and why we develop for the web and how we treat and consider the material of the web (network/code/browser) with the attention to detail we would afford any other material practice.

Four considerations

While there are many aspects to consider, this discussion will examine four and consider this to be part of a much larger ongoing conversation. These are (1) access, (2) documentation, (3) frameworks, and (4) ecological. Each of these is interconnected and cannot stand in isolation but is worth exploring individually as part of a larger, more complex whole.

(1) Access

There are four main vectors I would like to consider for accessing web code. These are (in the order in which they will be discussed) open web repositories, the ability to view the source of web content, the licencing of web-based projects, and accessibility of content to diverse audiences. In these four vectors, it can be seen that access to content and access to the material/code delivering this content are tightly linked.

Access: Repositories

There are many arguments as to why we may choose or not choose a particular platform to act as a repository for a project's code. In the case of Glossopticon, GitHub, a service owned and operated by Microsoft, was chosen. The ownership is mentioned here as this is a consideration when making decisions about how a project may be considered externally from the associations it makes with the wider "tech-world," such as making a considered decision to host with Microsoft. Prior to its purchase of GitHub in 2018, Microsoft had traditionally been seen as a champion of closed-source, proprietary software, while GitHub was always seen to champion open-source software practices. This association also brings up other considerations, including the ethical (or otherwise) practices of the platform being chosen. For example, Sasha Costanza-Chock reminds us

of developer dissatisfaction with Microsoft's political contracts in the USA (Costanza-Chock 2020), and, in doing so, questions the ethics of hosting code with GitHub. As with the ultimate decision to host the Glossopticon project on GitHub, the digitally, critically aware practitioner needs to weigh pros and cons and make a decision that is right for their project. Other possible platforms that provide similar functionality include Gitlab (often hosted by academic institutions) and Atlassian's BitBucket.

The reason for this choice is twofold, and this is the case for any of the platforms mentioned. The first is ease of access. The code on GitHub is easily accessed by anyone to freely download and use, and the in-built "social coding" options allow for these users to easily provide feedback or contributions to the project.

Secondly the nature of GIT (the version control system underlying these platforms) means that when a user "clones" the code that is hosted online by the platform, they have literally created a version of the code—their own version—on their local computer. This helps to decentralize the code, ensuring that the project is not relying on the whims of a corporation or institution for its ongoing availability. What is described here is using these platforms as both archive and distribution method for the sharing of code, as well as allowing users downloadable access to develop and extend projects for their own needs.

Access: View source

The second type of code accessibility is one that is a little more endangered these days, and one that up until recently has been a mainstay affordance of the web—that of right-clicking and viewing the page source. This has been a feature of web browsers from the web's origins, when the tools to both read and write web pages were transparently available in browsers. For example, Netscape Communicator (which is available to download from Internet Archive [Netscape Communications Corporation 1999]) and some other versions of early web browsers had in-built HTML editors (Version Museum 2023). Writing code for the web was considered an integral part of using the web. Access to and ability to manipulate the underlying source code for displaying content on the web should be kept front of mind when developing for design-led digital humanities projects aspiring towards open, accessible, and transparent web-based tools. The call to action here is to write code and create web-based documents with the intention that a user should be encouraged to inspect and even manipulate the source in the browser.

Access here is as much about the way the code is written as it is about the "magic" of being able to right-click and view that code in the browser. As users of the web, we have little control over the way browsers implement standards and provide access to code. Despite these things often being championed as open systems (W3C 2022), what we

do have control over is the way we keep our code human-readable and -writable. The languages of the web (HTML, CSS, and JAVASCRIPT) can be very easy to understand when they are viewed. They can also be written or post-processed in ways that purposefully obfuscate or complicate. In some ways, this is a result of trying to reduce the size of files that we are serving over the internet, and there are legitimate reasons to "minify" web code, though in many cases it can be argued that the difference is minimal and outweighed by the gains of accessible and readable code—in particular, if you consider that far larger savings in file size can be made in how we choose to treat the images, videos, and other rich media we serve over the web, which are considerations often treated as secondary or not at all. By thinking about the machine readability and the human readability of the code we write, we can make right-click and view source a useful and reliable tool for an engaged end user.

Even if an end user does not want to change the code or work with it for their own means, as an interested, digitally literate party, they may still want to read and validate the code, so keeping it human-readable is central to this.

Access: License

The licensing of the codebase of a project is an important decision. An in-depth discussion of the nuances of particular open (or otherwise) licenses is not appropriate here. The interested reader will find a useful introduction on the Choose a License webpage (Choose a License 2023). Platforms like GitHub make it easy to apply one of many licenses to code hosted on the platform. The license chosen for Glossopticon is the "GNU General Public License v2.0 (GPL-2.0)," which is a very permissive license. Licenses such as GPL-2.0 are primarily written from an engineering and IT perspective, and while they are transferable to code written for humanities-based projects, they may not always fit the ethical or conceptual positioning of a project.

Wendy Liu provides a critical take on some of the nuances of how open-source licenses may operate in a way that seems counterintuitive to what we may think of as free and accessible software (Liu 2018), indicating that we should not necessarily rely on established methods to be accessible and open. Because of this, there has been some interest in proposing alternate models to these licenses, such as "The Anti-Capitalist Software Licence" (Nasser and Pipkin 2023); the "Hippocratic License 3.0 (HL3): An Ethical License for Open Source Communities," developed by the Organization for Ethical Source in partnership with Corporate Accountability Lab (The Hippocratic License 2021); or the "Imagine, Open Source Software License for Peace and Nonviolence" (Crispin [2019] 2020). While the nature of these from a legal

perspective has been questioned (for example, in the Twitter thread associated with Crispin's proposed license [Crispin 2019] or the FAQ attached to the HL3 license), they raise important questions about how, why, and to whom we chose to give rights from a conceptual and ethical standpoint. Practitioners in design and the digital humanities may consider the use of a Creative Commons license appropriate for sharing their code-based projects. This is, however, discouraged by Creative Commons themselves, as Creative Commons' licenses "do not contain specific terms about the distribution of source code, which is often important to ensuring the free reuse and modifiability of software" (Creative Commons 2023).

Access: Diverse audiences

Designing for the web and the subsequent coding of web experiences has in recent years come under the spotlight for developing best practices for providing access for diverse audiences—from general design principles such as legibility, typographic choice, contrast, font sizing, and colour blindness considerations to web-specific considerations such as providing alt-texts on images. These developed best practices do not need repeating here, and practitioners are encouraged to seek out guidance to develop their own strategies for best practice for their intended audiences. Interestingly, Shannon Finnegan and Aimi Hamraie discuss this point in their conversation in *Extra Bold*, where they advocate for a more nuanced and creative approach to accessibility above and beyond standards. Finnegan states that "often, people seek consultation in search of concrete guidelines. But cut-and-dried instructions for alt-text haven't been working out yet" (Finnegan and Hamraie, 2021, 47), indicating that working collectively with audiences to develop a responsible and responsive approach to designing access for diverse audiences is a desired approach here.

(2) Documentation

Good documentation of web-based code projects can often fall behind in the process of trying to bring complex projects to completion in line with deadlines and other demands. Documentation can also be let slide when a project is released, and the compulsion is to move on to the next thing. For this author, this often falls into a state of "do as I say, not as I do." Regardless, it is the position of this paper that good documentation should be considered part of the process and preferably built into timelines. While best practice can be undermined by the realities of project timelines and funding constraints, if we consider good documentation to be an integral part of the material of code—as part of the process and outcome—then we can prioritize it as requisite.

Documentation: Code comments

In-line documentation with comments is one of the primary ways of documenting code-based projects, and I would say perhaps the most important. Commenting code is the process of adding "plain language," human-readable text into code (comments are completely ignored by the machine, in this case, the browser rendering content to the screen). There are a number of approaches to writing comments in code, and the one proposed here, which is used in Glossopticon, is based on an ethos of "creative coding." In this approach, we make use of commenting as much as possible, both as notes to oneself and as a means of explaining the way the code works to anyone who is interested and has accessed it via the browser inspector or from the repository (see **Figure 2**). It is an often-repeated mantra when teaching design students to code that comments are like little gifts to your future self. Comments also help build structure, and if we think about writing code as an act of writing (Reas and Fry 2014), we might see that we can create chapter headings and short synopses of what is to follow in the comments we provide.

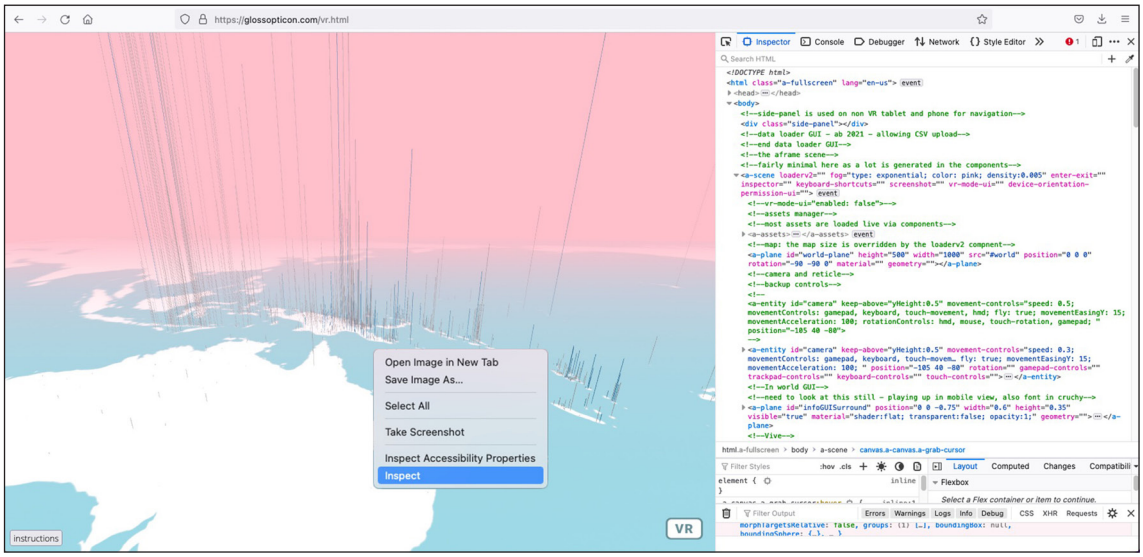

Figure 2: Inspecting the code—Glossopticon WebVR version.

Documentation: Function and variable names

In addition to comments, the names we give functions, variables, classes, and other named code structures can further enhance human readability and documentation. This is something that newer ways of working with the web has removed, in particular, when working with popular frameworks such as React. A function is a small named piece of code that, as its name suggests, carries out a particular function. Functions are used to modularize code and in web development are mostly found in JavaScript. As an example,

a function name in the Glossopticon project lets a digitally literate audience know what its "function" is. The function named `getXYpos()` returns the X and Y positions on a cartesian plane based on latitude and longitude inputs. When combined with the comments describing it—`returns the x,y coordinate of a lat long position from the database`—a human reader of the code has a clear indication of what it does.

Documentation: Website as container

The website as container for a project also acts as a point of documentation and an entry point into a project and is often the space we use to invite a user into a project. Glossopticon provides one way of doing this as a single scroll webpage framing the project, providing links to the project's papers and impact, instructions for use, an invitation to contribute to the code, and, of course, links to the project proper—the interactive visualization. Links are also provided to the GitHub repository where the code itself can be found, and an associated "read me" file discusses the finer point of working with the code as a tool to tailor the visualization *as tool* to an end user's needs.

We can provide framing for a project in the repository for a user who wishes to perhaps extend on or validate a project at a deeper level, and it is also possible to use the repository itself as a main landing space or primary web presence for the project. This includes information on how the user can download, implement, and use the code that is shared with them. It also includes other important information about the license under which the code is shared.

(3) Framework

The third consideration is the choice of frameworks made in creating a project. In thinking about the how and why of choosing a framework for a project, it is important to ask if the framework is abstracting or simplifying lower levels of web code that would otherwise be cumbersome to learn and work with, and/or do they provide an off-the-shelf solution to something that would otherwise mean spending a large amount of time solving? A second, equally important question is whether the framework is licensed in a way that is compatible with the license I wish my project to be available under and is in line with the ethical framing and position of the project.

The main framework used in Glossopticon is AFRAME, a web VR framework that abstracts quite a complex layer of the web—that of WEB-GL—while at the same time continuing to give us access to the underlying layers. AFRAME itself sits on top of the THREE.JS framework, which is in turn is an abstraction of WEB-GL. The framework therefore simplifies and improves a steep learning curve rather than replaces one with another.

(4) Ecological

This may be the most difficult of factors to control, but in the face of global climate catastrophe, it may be the most important. There are many practical steps that can be taken to counter the ecological impact of web-based projects. These impacts are often invisible and often go unacknowledged. In particular, the ecological impact of web server infrastructure often goes unseen. The website of Low-Tech Magazine (Low-Tech Magazine 2023) provides a lot of useful contexts here, as well as providing a number of possible solutions. One solution offered is the fact that the site is hosted on a solar-powered mini server, with "do it yourself" guidelines to creating your own. This solution is obviously not scalable, but it does visualize the invisible distinctly through what could be called a discursive design object (Tharp and Tharp 2018), where its function is as much to generate discourse towards solutions, as to provide a solution in and of itself. More pragmatic solutions are also offered, including reducing the file size of rich media elements, as well as taking into close consideration when and where these elements are even necessary.

Glossopticon purposefully keeps the audio snippets it presents as short and compressed as possible in order to reduce the footprint in both server storage space and file transfer when used in the visualization. It also considers carefully its use of images and other rich media files in both the visualization proper and the project website. This is an ongoing effort as part of the Glossopticon project, and the author attempts to address the impact on climate change of their design-led visualization and creative practice that relies heavily on these often-unseen technical infrastructures.

Conclusion

In the words of Winnie Soon and Geoff Cox, we must "consider programming to be a dynamic cultural practice and phenomenon, a way of thinking and doing in the world, and a means of understanding some of the complex procedures that underwrite and constitute our lived realities, in order to act upon those realities" (Soon and Cox 2020, 14). This paper provides some ways to engage our project's stakeholders in this "dynamic cultural practice" and "way of thinking and doing in the world." It is intended to provide some insight into an ongoing discussion that is in a constant state of development and redevelopment, and as an invite and an opening up of this discussion to design practitioners and digitally literate digital humanities scholars, with an ongoing aim to form a robust way forward for thinking about accessible, open, and ethical design-led digital humanities visualization projects on the web and beyond—visualizations that, as a result, are transparent, interoperable, and reproducible.

Competing interests

The author has no competing interests to declare.

Contributions

Editorial

Special Guest Issue Editors

Barbara Bordalejo, University of Lethbridge, Canada
Emmanuel Château-Dutier, University of Montreal, Canada
Roopika Risam, Dartmouth College, United States

Copy and Production Editor

Christa Avram, The Journal Incubator, University of Lethbridge, Canada

References

Berners-Lee, Tim, and Mark Fischetti. 1999. *Weaving the Web: The Original Design and Ultimate Destiny of the World Wide Web by Its Inventor*. San Francisco: Harper San Francisco.

Burrell, Andrew, Rachel Hendery, and Danièle Hromek. 2021. "Nura Yaman ('Country Speaks'): Language, People and Place in Serious Games." *Leonardo Electronic Almanac* 23(1): 1–27. Accessed October 10, 2023. http://hdl.handle.net/10453/155122.

Burrell, Andrew, Rachel Hendery, and Nick Thieberger. 2019. "Glossopticon: Visualizing Archival Data." In *2019 23rd International Conference in Information Visualization – Part II (IV-2 2019)*, edited by Theodor G. Wyeld, Ebad Banissi, Anna Ursyn, Mark Bannatyne, Nuno Datia, and Muhammad Sarfraz, 100–103. Accessed November 7, 2023. DOI: https://doi.org/10.1109/IV-2.2019.00029.

Choose a License. 2023. "Choose an Open Source License." Accessed October 11. https://choosealicense.com/.

Costanza-Chock, Sasha. 2020. *Design Justice: Community-Led Practices to Build the Worlds We Need*. Cambridge, MA: The MIT Press.

Creative Commons. 2023. "Frequently Asked Questions." Accessed October 11. https://creativecommons.org/faq/#can-i-apply-a-creative-commons-license-to-software.

Crispin, Sterling. (@sterlingcrispin). (2019) 2020. "Imagine License." GitHub. Accessed October 7, 2023. https://github.com/sterlingcrispin/imaginelicense/blob/0cac99f6c9010d6d54749929cfc16d91357d0d62/LICENSE.md.

———. (@sterlingcrispin). 2019. "Excited to share a first draft of an open source software license that would explicitly oppose violent usages of the software. If you have any feedback or suggestions please share or make a pull request. Thank you #opensourcesoftware #nonviolence. https://github.com/sterlingcrispin/imaginelicense/blob/master/LICENSE.md." Twitter, April 8, 6:30 p.m. Accessed October 11, 2023. https://twitter.com/sterlingcrispin/status/1115411627849863168.

Finnegan, Shannon, and Aimi Hamraie. 2021. "Voices: Shannon Finnegan and Aimi Hamraie (Conversation Moderated by Emily Watlington)." In *Extra Bold: A Feminist, Inclusive, Anti-Racist,*

Non-Binary Field Guide for Graphic Designers, edited by Ellen Lupton, Farah Kafei, Jennifer Tobias, Josh A. Halstead, Kaleena Sales, Leslie Xia, and Valentina Vergara, 44–47. New York: Princeton Architectural Press.

Hayles, N. Katherine. 2012. *How We Think: Digital Media and Contemporary Technogenesis*. Chicago: University of Chicago Press.

Liu, Wendy. 2018. "Freedom Isn't Free." *Logic Magazine*, 5. Accessed October 7, 2023. https://logicmag.io/failure/freedom-isnt-free/.

Low-Tech Magazine. 2023. "Low-Tech Magazine." Accessed October 11. https://solar.lowtechmagazine.com.

Mauro-Flude, Nancy, and Yoko Akama. 2022. "A Feminist Server Stack: Co-Designing Feminist Web Servers to Reimagine Internet Futures." *CoDesign* 18(1): 48–62. Accessed November 7, 2023. DOI: https://doi.org/10.1080/15710882.2021.2021243.

Nasser, Ramsey, and Everest Pipkin. 2023. "The Anti-Capitalist Software License." Accessed October 11. https://anticapitalist.software/.

Netscape Communications Corporation. 1999. "Netscape Communicator 4.6 (En-Gb) (99124)." Accessed October 7, 2023. http://archive.org/details/netscapecommunicator4.6en-gb99124.

OLAC (Open Language Archives Community). 2011. "Home." Accessed October 11, 2023. http://www.language-archives.org/.

PARADISEC (Pacific and Regional Archive for Digital Sources in Endangered Cultures). 2023. "Safeguarding Research in Australia's Region." Accessed October 11. https://www.paradisec.org.au/.

Reas, Casey, and Ben Fry. 2014. *Processing: A Programming Handbook for Visual Designers and Artists*. 2nd ed. Cambridge, MA: The MIT Press.

Soon, Winnie, and Geoff Cox. 2020. *Aesthetic Programming, A Handbook of Software Studies*. London: Open Humanities Press.

Tharp, Bruce M., and Stephanie M. Tharp. 2018. *Discursive Design: Critical, Speculative, and Alternative Things*. Cambridge, MA: The MIT Press.

The Hippocratic License. 2021. "Hippocratic License 3.0 (HL3): An Ethical License for Open Source Communities." Accessed October 11, 2023. https://firstdonoharm.dev/.

Version Museum. 2023. "Visual Design Evolution of Netscape Navigator." Accessed October 11. https://www.versionmuseum.com/history-of/netscape-browser.

W3C (World Wide Web Consortium). 2023. "Standards." Accessed October 11. https://www.w3.org/standards/.