Danielle Calle
CINE-GT 1807
Instructor Nicole Martin
December 14, 2018

Disentangling the Retiary:
A Laurie Spiegel Software Preservation and Web Archiving Project

[Figure 1] Examples of early computer art by Laurie Spiegel, developed at Bell Laboratories, 1974-1975.[1]

For composer and computer programmer Laurie Spiegel, computers can be seen like a new kind of folk instrument. As a folk musician herself, Spiegel was motivated by the desire to open creativity and expression to all people, and she envisioned computers as facilitating this through exponential and infinite ways. A pioneer of many worlds and forms, Spiegel was invested in unlocking and demystifying creative processes through her life's work. Disentangling the Retiary *is an homage to an artist who has been emblematic of the vast technological advancements to creative industries in the last five decades. This digital preservation assessment serves to honor her career and work and enable new pathways for its longevity into the future.*

Laurie Spiegel is an electronic musician and computer programmer most known for her algorithmic compositions such as *The Expanding Universe* and *Unseen Worlds*. In the 1970s, while her music was travelling through space,[2] Spiegel counted herself among the few artists experimenting with the new possibilities afforded by computers. Spiegel spent formative years at Bell Laboratories pioneering early computer graphics [Figure 1], digital synthesizers,[3] and hybrid analog and digital computing systems like the GROOVE.[4] Her more than fifty-year career is representative of the exponential technological changes that have influenced art practice as a

[1] http://retiary.org/ls/btl/ls_btl_art.html
[2] https://www.theverge.com/2012/12/7/3740280/laurie-spiegel-profile-pitchfork
[3] https://www.youtube.com/watch?v=NChqEEz31eE
[4] http://retiary.org/ls/btl/groove_quick_description.html

whole. In contrast to her highly lauded music career, her website, *Retiary.org*, illuminates another side to the artist's process. *Retiary*, which has been maintained since the late 1990s, is a repository for her writings on media theory and opinions on the development of the technology most relevant to her artistic practices. A particularly important aspect to the site is the availability of two software programs Spiegel authored in the mid 1980s: Music Mouse and MIDI Terminal. The following assessment aims to capture and document Spiegel's life and work as represented through this Retiary Network. Retiary, the word for a Roman gladiator that fights with a steel net, embodies for the artist a "weblike" or "netlike" approach to her thought and practice.[5] To honor the intentionality with which Spiegel has imbued her personal site, *Disentangling the Retiary* aims to serve as a digital preservation project for an underappreciated corner of the web, and hopes to unpack the many treasures held therein. The project will delineate the ways this website reflects the tendencies of the early web through its active discourse and critique of the technologies made available at the time.

In order to fill in this much-needed gap, the following assessment will highlight preliminary findings in capturing and documenting the assets Spiegel has made available online. These include two websites belonging to the Retiary Network—*Retiary Ramblings* and *Laurie Pages*—and the two aforementioned software programs. It will also consider proper strategies to verifying the validity of these files, and inject within this the appropriate historicity to contextualize the moment of its creation, that being the early days of the Internet. In highlighting and reflecting on the digital preservation challenges inherent to representing works that originate

[5] https://web.archive.org/web/19981202073123/http://retiary.org:80/

in the digital realm, the project will track appropriate next steps for a future, more comprehensive plan.

A Web and Two Knots—*Laurie Pages* and *Retiary Ramblings*

In the late 1990s, statements like "Welcome to My Homepage!" became emblematic of Web 1.0, and sites like the now (almost) defunct GeoCities cemented websites as a form of personal expression and not merely for companies or businesses. The activity of creating personal archives for the public set a precedent for the social media networks of today. In perusing through the assembly of posts featured on *One Terabyte of Kilobyte Age*,[6] a website dedicated to disinterring archived GeoCities pages, its accompanying oral history project helps orient one to the emotional logic and at times arbitrary decisions early web users would make in crafting their personal sites.[7] In the absence of web spaces dedicated to specific functions (LinkedIn as a portfolio of work, Instagram as a catch-all for images) personal websites were a locus for creative output to the layman and professional alike. Laurie Spiegel's websites can be seen as representative of this time period.

The larger organizational structure of the site is referred to as the Retiary Network, and further broken down into two distinct subentities—*Laurie Pages* and *Retiary Ramblings*. The former is a showcase of Spiegel's career, while the latter is more nebulous in its purpose, but offers a playful philosophical conduit for Spiegel's interests and hobbies. To learn of the web archiving practices available today, two methods were used: Webrecorder, a free web archiving

[6] http://blog.geocities.institute/
[7] Exemplary interview GeoCities Oral History. Interview #2 http://blog.geocities.institute/

tool developed by Rhizome in 2014,[8] and 'wget,' a command line function that was developed in 1996.[9] Because of the vastness of this site (it proved to be more meandering and tangled than expected), a small sample was considered, the "Writings" section to *Laurie Pages*. As a case study, this segment of the website still reveals important facets to the web archiving process. Following this preliminary capture, Webrecorder was used to open the WARC files to compare the two methods employed.

Webrecorder

Before beginning this endeavor, a spreadsheet was created to document the links categorized under 'Writings' which amounted to a total of 37 pages, not considering outlinks held within each page.[10] After the spreadsheet was created, the URLs were tracked upon the activation of a Webrecorder "session," the name given to the web-recording process in the software. Outlinks proved to be the most problematic issue for the website as a whole. Many of them led to websites that are no longer active, though thankfully to the Internet Archive's Wayback Machine, various captures throughout the years were available to "patch" into the record. A common issue with patching is knowing which archived website to implement. For the "Ex Post Facto OMS" article for example, its outlink to the site *Opcode.com* has over 305 Wayback Machine captures from 1996 and onwards.[11] Choosing the most appropriate one depends on the time period that is being targeted for representation in the collection. If a version of the site from 1998 is chosen and subsequently included in the overall procedure, is this an

[8] https://blog.supdigital.org/saving-the-internet-an-interview-with-webrecorders-dragan-espenchied/
[9] https://www.howtogeek.com/281663/how-to-use-wget-the-ultimate-command-line-downloading-tool/
[10] Google Spreadsheet here.
[11] https://web.archive.org/web/19970212235719/http://www.opcode.com/

accurate portrayal of the website in 2018? In figuring out this quandary, it became apparent the inherent value to documentation. A positive feature of Webrecorder is the ability to create lists, which can provide a way to add more nuance to the record. These lists can therefore be assigned to a certain year of the website's life, and placed in chronological order, adding a historical perspective to the overall process.

In viewing the WARC file via the Webrecorder Player, the website appeared and worked as expected. Webrecorder Player by design is made to mimic the look and feel of a web browser, so the targeted 37 pages were made viewable with relative ease, and there were no surprises as to how the content was displayed. One small issue was apparent in the metadata. Through a side panel called "Pages" one can navigate captured pages from each session. Though 37 were catalogued in the spreadsheet, 81 appeared in the navigation menu. Upon further inspection, it became clear that Webrecorder, when navigating naturally through a website, creates redundant copies of pages if backward or forward navigation is used. To have a more neatly defined set of pages, one would have to create a session for each URL (as captured in the spreadsheet) and collate them through the "patch" function in Webrecorder.

wget

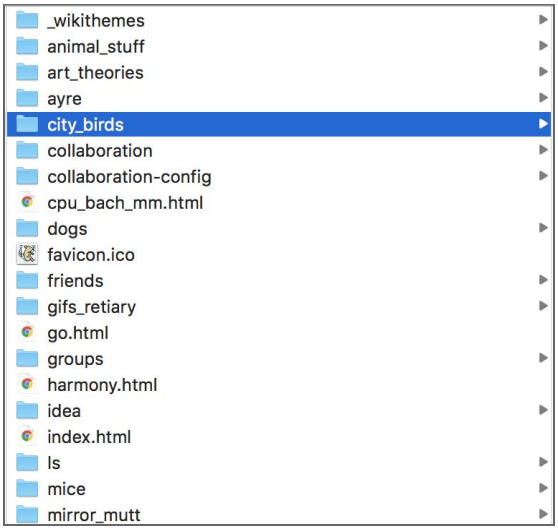
The usage of wget revealed another facet to the web-recording process, one that can be useful if creating records within a limited time frame. The following command was used to capture "Writings":

```
wget -e robots=off -r -l 1 -p --waitretry 5 --timeout 60 --tries 5 --wait 1
  "http://retiary.org/ls/writings.html" --warc-file=lauriespiegel_writings
```

Once the capture was complete, the file was assessed using Webrecorder Player. Instead of the 37 pages accounted for in the spreadsheet, 39 pages were visible in the application. A

comparison of the spreadsheet to the capture was once again warranted. A reason for the additional pages is the way in which the landing page to *Laurie Pages* is arranged. It is divided in three parts—the index and two frames (one left, one right). Whereas the spreadsheet had only tracked the URL for the index (the ones for the other frames were not visible when naturally browsing the site) the automated process employed by wget accounted for two additional "frames" that make up this page [Figure 2]. Despite this being a surprise, it was informative to realize that the website's landing page was divided in such a way—three frames, each with a separate URL, overimposed one another to propel the site's navigation.

[Figure 2] Additional "frames" in Webrecorder.



[Figure 3] Organizational logic to Laurie Pages.

Overall, wget provided a cleaner capture with fewer, distinct pages. More importantly, it has the potential to highlight elements that are not immediately apparent. An earlier attempt to use wget created folders for the capture that can be seen in Figure 3. This exposed internal organizational structures that might not have been visible otherwise, and a personal logic to how Spiegel categorizes her work. For any archivist, this is valuable insight into a work's custodial history and the collecting strategies that were utilized.

Web Archiving Takeaways

There are pros and cons to using Webrecorder and wget. While in many ways wget affords a cleaner and more expedient process, Webrecorder does allow the ability to patch in older, defunct websites. To fully assess the site, the spreadsheet deserves more time to encompass the entirety of *Laurie Pages* and *Retiary Ramblings.* A combination will need to be used in order to account for the many changes the website has experienced. While Webrecorder is more time-consuming than wget, it has the potential to be more exact if taking the proper measures. Overall, documentation in the form of a spreadsheet proved essential to ensuring that all of the correct links were accounted for.

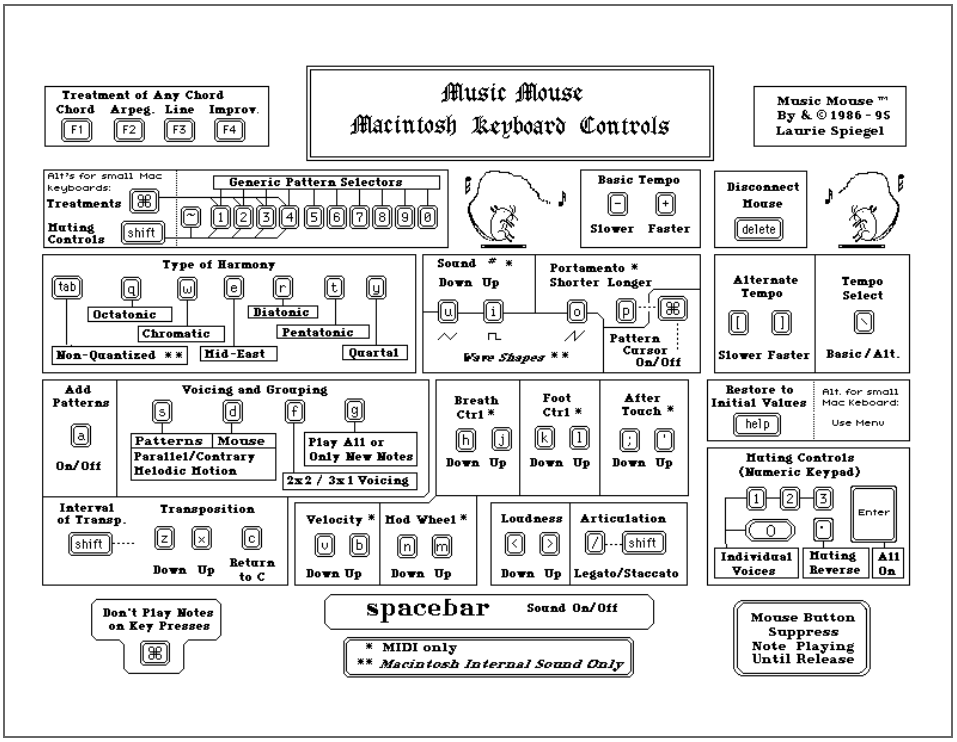
Emulation Strategies for Music Mouse

Music Mouse: An Intelligent Instrument originally debuted in 1986 and was written for Macintosh, Atari, and Amiga computers. As its name suggests, it was developed as a "pointer" or guide to service the unassuming creative, simply wanting to explore their inner ear.[12] Music Mouse, in addition to relying on hardware peripherals, allowed for the configuration of MIDI instrument input. MIDI Terminal, which can be understood as a complementary software to Music Mouse, was therefore made to visualize, in human-readable terms, communication between a MIDI instrument and a computer. Because no MIDI instruments were available at the time of the assessment, the focus was made on Music Mouse instead.

The basic functions of Music Mouse are meticulously explained in the user manual, with an accompanying keyboard map [Figure 4]. As shown by this image, keyboard-mouse

[12] Why Music Mouse? An Introduction http://retiary.org/ls/progs/mm_manual/mouse_manual.html

coordination is required to create music. The image of the map must also be displayed on a computer monitor or printed physically in order to recall the functions, though through practice it is possible to learn these by memory, just as one would with any other musical instrument.



[Figure 4] Music Mouse Keyboard Map, http://retiary.org/ls/progs/mm_manual/mouse_manual.html

Reviving Music Mouse was achieved using two different strategies: software-for-hardware emulation and two legacy computer environments.

Software-for-Hardware: SheepShaver

Sheepshaver is an open-source software-for-hardware program that mimics classic Macintosh environments (versions Mac OS 7.5.2 through 9.0.4),[13] and is free to download via *Emaculation.com*. For the first trial of Music Mouse, the emulation was used primarily to get an idea of how the program functions. Almost immediately, the differences were apparent between

[13] https://www.emaculation.com/doku.php/sheepshaver

a modern keyboard layout (using a 2017 13-inch MacBook Pro) and the one designated by the Music Mouse Keyboard Map. Contemporary computers and their peripherals have changed considerably since the 1990s. Most notably, keyboards have been truncated to enable sleeker, more economic design. The program makes use of keys like “Help” that are no longer available today, and entire commands allowed by function keys require a slightly more cumbersome approach to activate using the ‘fn’ key to enable the function keys row. In addition to this, for reasons that are not immediately apparent, the emulator did not successfully generate memorized patterns in the software when pressing the proper commands (toggling the ‘a’ key). It is possible that at the command line level, keyboards have changed in the way they are programmed into computers to reflect the change in design.

Hardships in Hardware

The hardware installation took place using two different machines and on three separate occasions. The first computer was a Power Mac G5 using the “Classic Environment” and the second was a Power Macintosh G3 Desktop running Macintosh 8.5.1 [Figure 5,6]. The first trial on the Power Mac G5 allowed for a more natural interaction with Music Mouse, but even here the keyboard was too modern for the intended use of the program’s instructions. In keeping with the mismatched functions displayed by the emulator, the keyboard appeared confused by the commands. The ‘q’ ‘w’ and ‘e’ keys, which all relate to the harmonic adjustments one can make in the program, were offset by one character. In other words the ‘q’ became a ‘w,’ the “w’ an ‘e,’ and so forth. To gain further clarity on the basic functions of the program, two more attempts were made on a Power Macintosh G3, this time with another slew of complicated results.

On the first day using the Power Macintosh G3, the software was burned onto a DVD disc, but it was quickly realized that the computer could not read this carrier. Instead, another copy of the software was made onto a CD, and then loaded into the computer. After a successful installation, the program booted up. However, once again the keyboard that accompanied this computer was not ideal for the application since it had an array of missing keys, and most notably, the function keys row. An additional attempt was made on a separate occasion to use a more compatible keyboard, but even this brought about more issues. The keyboard for this second attempt could not simultaneously connect into a mouse, and seeing that the computer only had one ADB port [Figure 6], the attempt proved limiting. Despite this, the multiple unsuccessful attempts revealed hardware components to be an integral component to software preservation.

[Figure 5] Power Macintosh G3 Desktop.

[Figure 6] Apple Desktop Bus (ADB) port.[14]

[14] https://deskthority.net/wiki/Apple_Desktop_Bus

Software Preservation Takeaways

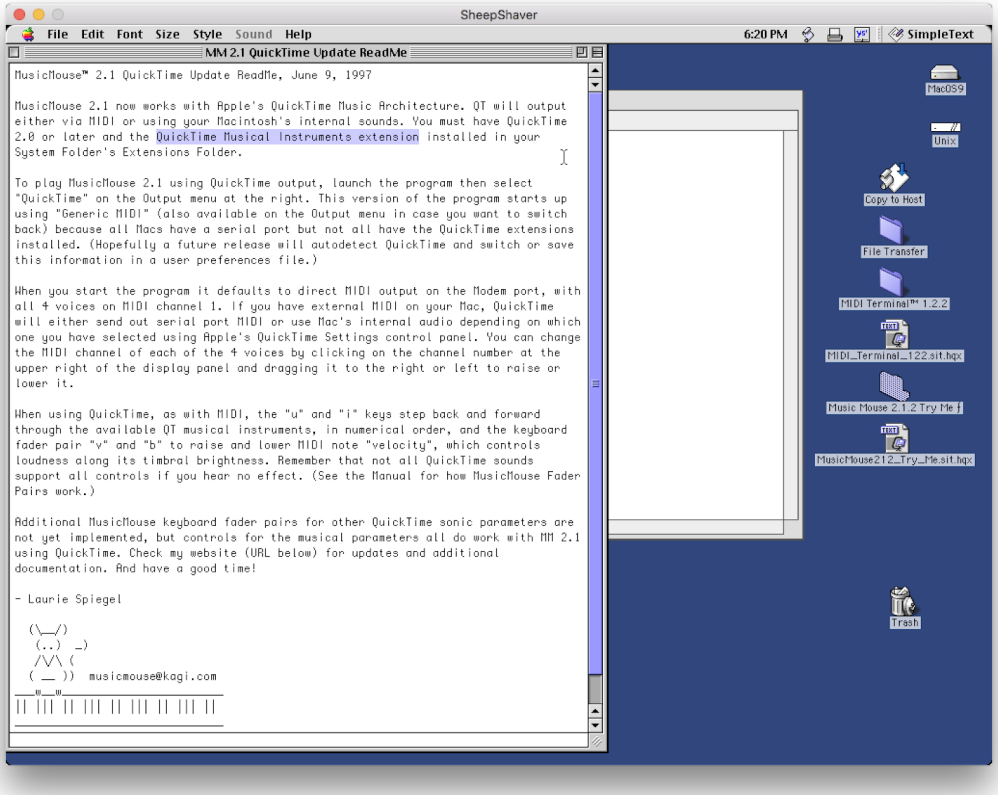
Software and hardware emulation offer important access points to the revival of programs such as Music Mouse, but the order in which it is done is crucial. For the purposes of getting the program to run as quickly as possible, without legacy hardware immediately available, the first method used was software-for-hardware emulation. In hindsight, it was apparent that the process could have benefitted having a correct understanding of the program's capabilities through its native environment, mainly because the program was designed around hardware peripherals (legacy keyboard and mouse). The preservation project further merits the acquisition of proper peripherals to more smoothly run the program.

In terms of documentation, it is important to record examples of the output sounds, images, or text whenever possible. Relying on memory, it was noted that the program sounded different on each system. This prompted an investigation into the Quicktime Musical Instruments extension that is mentioned in the 'ReadMe' files for the software [Figure 7]. More options should also be explored via the emulators available online such as Basilisk II which can run older versions of the Macintosh operating system. Overall, the process revealed important factors to preservation issues and the continued troubleshooting that must be accomplished in order to have a more complete assessment.

Next Steps

Laurie Spiegel's body of work, as evidenced by the assets made available on her website, are a good case study for the inherent issues digital materials face in the early Internet age. As noted by David Rosenthal's *Digital Preservation Systems*, the two arenas that have been

assessed—web archiving and software preservation—present unique issues that are not always easy to remedy. To treat them, they require a specialized and customizable focus. For the risks of software preservation, it is apparent through this preliminary assessment the many bugs one might encounter, pointing to software and hardware obsolescence when a program is not updated to accommodate changes to modern computing environments. The website, too, reflects common issues to web archiving—the failure of network services, the lack of persistent URLs, and the transient nature of the Internet. Though there are many rabbit holes that are difficult to emerge from, it is clear that such preservation efforts require time, consistent documentation, and resources to tackle to the best ability.



[Figure 7] Screengrab of MM 2.1 QuickTime Update ReadMe.