Do You Even Noise?

Sumanth Srinivasan
MS Electrical Engineering
Tandon School of Engineering
New York University
sumanths@nyu.edu

ABSTRACT

Do You Even Noise? is an installation that attempts to allow the user to embody a mathematical abstraction. Drawing from both communication models and media studies, the project uses audio as a medium of communication, where the user is allowed to play the role of Noise, that disrupts coded information while it is transmitted from the a software synthesizer to the speakers in the form of digital signals. In doing so, the software also exposes some of a browser's inherent limitations in encoding and decoding audio while exploring its aesthetic potential.

1. INTRODUCTION

Since the inception of language and early symbolic communication, humans have rather instinctively embodied the roles of an encoder and decoder in modern systems of information transfer.

Through various generations of abstraction, we have progressively found new ways to encode ideas and expressions using various symbols in our environment and trained ourselves to ignore noise – an event that we attribute to the environmental changes affecting information transfer either partially or completely out of our control.

Both Information Theory and Literary Media criticism are credited with enormous amounts of research and organization of various challenges in communication, most remarkable ones being those of Claude Shannon [1] and Jacques Derrida [2].

2. THE COMMUNICATION MODEL

Widely regarded as the father of Information Theory, Claude Shannon was among the first to separate the semantic aspects of communication from the technological problems that surround it. In doing so, he treated transmission of information between machines as a problem completely different from transmission of meaning between human beings. He wrote:

"The fundamental problem of communication is that of reproducing at one point either exactly or approximately a message selected at another point. Frequently the messages have meaning..." [1]

Shannon's communication model, in terms of Signal and Noise, divides the problem of information transfer into five blocks. The transmission begins at a *source,* which produces the message. An *encoder* turns this message into signals suitable for transmission, before sending it via a *channel.* A *receiver decodes* the message from the signal and delivers it to the destination. This model also includes an additional, perhaps a crucial, component – *Noise*, which is an external disruptive factor in the system. Historically, noise has been viewed as an errant factor leading to imprecision in the output of any system. Shannon borrowed this idea into the world of signal processing in an attempt to quantify the kinds of materiality that caused unexplained variations in the message as seen at the destination.

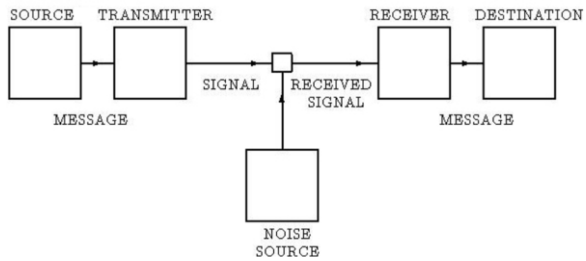

Figure 1. Shannon's mathematical model of communication [3]

Besides an external noise source, Shannon also talks about *entropy* – the kind of noise that comes with uncertainty between the *symbols* used for communication and the states of the medium that transport it. Since communication – particularly in signal processing – can be viewed as a series of state changes, where each state (or set of states) corresponds to a given aspect of the message, the thermodynamic and probabilistic limitations of the system may in itself introduce another type of noise.

3. NOISE IN MEDIA CULTURES

While Information Theory was driving towards a streamlined understanding of communication, post-structural thinkers drew from its mathematical abstractions to understand communication of meaning by questioning analogies such as intention/sending and interpretation/receiving.

Jacques Derrida [2], emphasized on the temporal uncertainty of the sender's intention or the receiver's interpretation of any given message. In the context of post-structural ideas that were already challenging conventional notions about meaning, Derrida pointed out that a given symbolic representation may go on to signify a number of different messages and ideas over time. Hence communication was viewed as the process of negotiating semantic ambiguity but not to overcome it.

There is a convergence in the ideas of both Shannon and Derrida in that all systems of communication are subject to certain amount of noise (either as external disruptions or internal ambiguities), and that this creates variability in the receiver's understanding of the message. [4]

4. INFORMATION AND NOISE ART

While the model described by Shannon is fairly deterministic, it's predictability is undermined by the addition of noise, which reveals itself over the information in the form of 'noise artifacts'. The type of artifacts that manifest greatly depends on the form of information, the deliberate processes of *encoding* and *decoding*, as well as the *corruption* of the signal while in transmission.

In *Glitch Moment(um),* Rosa Menkman [5] talks about how the post-procedural nature of glitch-art reveals the machine's internal techné and enables critical sensory experience to take place around materials, ideologies and structures that shape an information system.

Numerous artists have incorporated noise and disruption in their work. Most notable of them is Jodi, a Dutch/Belgium art duo that focuses on making subversive glitch art, which battle against the general hegemonic flows of proprietary media systems. By not attaching explanations to their work besides overturning generic expectations that are attached to specific media, Jodi have created semantic ambiguity that directs attention towards the medium by deliberately leaving it to malfunction and output improbable information. This is seen in their *UNTITLED GAME*, a series of 11 modifications to the visuals and control mechanics of the popular first-person shooter *Quake 1*.

Another example is et al. – an artist's collective based in New Zealand that seek to enfold information and noise with the structures of a conventional gallery exhibition and raise the possibility that in transforming the structures and materials through which they travel, information and noise have left important traces that can be harnessed by an art installation. In their 2006 installation *maintenance of social solidarity-instance 5*, et al. provide a space with an information overload across audio and visual media, where the uncertainty of coherent message turns every piece of tangible information into a component of the collective noise [6].

5. DESIGN AND CORE MECHANIC

Do You Even Noise? is setup in the form of an installation that allows the user to listen to audio over headphones and interact with the information system using a keyboard.

The underlying software makes use of tone.js [7], a javascript framework built over the browser's WebAudio API [8], which suitably complements the modular nature of Shannon's communication model. Working entirely on a browser, the software uses the native abilities of the browser and library to code musical notes into frequencies and into sound, treating various intermediate stages as the *channel* and hence a testing ground for disruption possibilities.

In emphasizing on the interactive nature of the system, the installation makes sure that every entity involved has sufficient agency to cause a state change in another. A visual counterpart of the system was built using p5.js to provide feedback of a user's involvement.

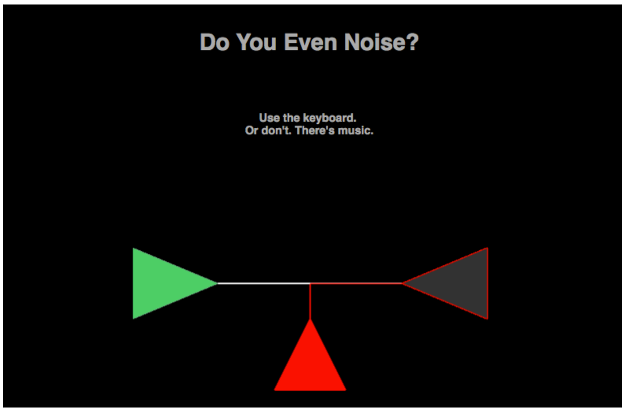

Figure 2. A screenshot of the installation featuring a minimalist representation of Shannon's model, with the Noise module dynamically appearing upon user interrupt via the keyboard.

5.1 Audio Mechanic

The *source* is modeled within the javascript by constructing a `Tone.MonoSynth()` object that takes a set of musical notes from an array and outputs it as frequencies with additional parameters such as `oscillator.type`, `envelope` and `volume`. The array of notes functions as a *message* in this system, where each corresponds to a specific tone, together forming a melody as interpreted by the listener, who plays the role of a *receiver* by listening to the music from headphones.

The *Noise* module is modeled using the keyboard interrupt functions `keyTyped()` and `keyReleased()` in the p5.js library, which detect a user's interaction with the keyboard and call from a bank of routing functions that introduce noise in the signal as it goes from the synthesizer object to the system's audio driver. Every time the user holds down an ASCII key on the keyboard, the script triggers a random callback function from the bank. This randomization leaves the user in a state of uncertainty about the nature of disruption that their interaction might result in. The original synthesizer output is routed back to the audio drivers once a key release is detected.

Since both the *Receiver* and *Noise* being embodied by the same user was very likely in an installation setting, a major design problem was to allow the user to experience the audio output and the effect of their interruptions in the system separately. Initial iterations included allowing keyboard interrupts to permanently noise the audio, and building on the ambiguity between signal and noise.

When no keyboard interrupt is detected for a specified period of time, the script triggers a looping melody, emulating a perfectly functioning information system. A keyboard interrupt will halt the synthesizer at the current note, until the channel is clear again. While halting of transmission in the event of noise is not a common occurrence in real-world communication systems, this mechanism has been emulated to highlight the internal feedback system that allows every entity in the installation to affect the others.

The bank of functions use a variety of popular audio signal processing techniques (includes delay, envelope shaping, amplitude quantization and filtering), but by setting parameters beyond the generally accepted musical aesthetic. However, though disruptive in nature, the functions contain the potential of providing aesthetic relief when considered with different sensibilities of music and sound – leaving it to the iterative interpretations of the user.

Certain callback functions also have temporally varying parameters that are a function of oscillating signals, which add another layer of uncertainty to the nature of disruption. An example callback function is as follows:

```
function() { //FUNCTION 6
        var split = new Tone.Split();
        var noise = new Tone.Filter().toMaster();
        synth = new Tone.SimpleSynth().connect(split);
        var lfo = new Tone.LFO("2n", 10,
5000).start();
        lfo.connect(noise.frequency);
        split.right.connect(noise);
        split.left.toMaster();
    }
```

Some functions, such as delay with high feedback, leave a longer impact on the audio, and continue to be audible even after the user has released the key. Successive key presses hence result in a unique accumulation of such artifacts that eventually overpower the actual signal.

5.2 Visual Feedback

To emphasize on the role that the user is playing in the system, the web page running the script features a minimalist representation of Shannon's mathematical model. The page shows two triangles, tacitly representing the *Source* and the *Receiver,* which change colors in tandem, highlighting a normal course of information transfer.

When the user interacts with the keyboard, a third module, representing *Noise* appears on the canvas (Fig. 2), and its effect on the signal chain is symbolized using a red-colored channel that goes on to the *Receiver.*

6. OBSERVATIONS AND ANALYSIS

6.1 Two Stages of Decoding

During development, the initial model followed assumed the software synthesizer as the *encoder* and system's native audio drivers as the *decoder*. Hence, most of the disruption capabilities built into the software were centered on modifying or re-routing data from the synthesizer to the browser output.

Upon repeated testing, it was discovered that there was an additional layer of decoding that the browser performed to convert the output of javascript into audio before sending it to the audio driver.

The presence of this additional layer manifests in an interesting scenario, where the noising occurs outside the scope of the javascript decoder, hence allowing the script to function in a perfect feedback loop – playing a melody oblivious to the distortion occurring before the audio reaches the listener. It was seen that this occurred when keys were pressed in rapid succession.

6.2 Browser Fingerprint

Since this new noise artifact outside of the script was due to a browser malfunction, it served as motivation to test other browsers with the same condition.

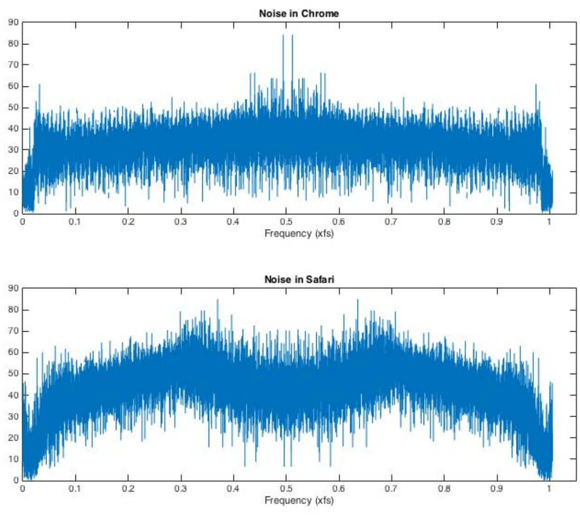

Figure 3. Plots of frequency response after buffer overflow in each browser. (Amplitude in dB, fs = 44100Hz)

The script was run in Google Chrome and Apple Safari and interrupted sufficiently in each case until the key buffer overflowed and broke the audio output. Clean and noisy audio outputs from the browser were recorded in each case and a frequency response relating the clean and noisy audio was computed by dividing their FFT values. This analysis was performed using a `Matlab` script.

It was found that each browser has a unique frequency response when it comes to the distortion outside the script. Fig. 3 shows the two frequency plots.

While Google Chrome has a fairly uniform response, there are peaks in the high frequency range, suggesting clipping or amplitude bit-quantization. In case of Apple Safari, it is seen that there is a rise in the mid-frequency ranges and then a drop as it approaches the Nyquist rate (22050Hz). There is also a peak in the lower frequency regions, which leads to the speculation that something to the effect of down sampling occurs in Safari. Understanding the exact process of this glitch in a browser is worthy of exploration.

7. CONCLUSIONS AND FUTURE WORK

This paper outlines the design process, iterative methods and signal processing tools involved in building the browser-specific installation, *Do You Even Noise?* It also discusses the motivations behind using mathematical and literary models of information systems as inspiration for artistic ends. Additionally, the paper quantifies some of the discoveries made about the inherent limitations of a web browser in its function as an encoder and decoder.

Further work may include research on the audio capabilities of various browsers and narrowing on crucial processes involved in their decoding algorithms that may be obscured by its otherwise perfect functioning. This will provide more conclusive audio fingerprints of software decoders – information that may be valuable with the increasing reliance on network transmission for production and consumption of digital media.

8. REFERENCES

[1] Shannon, C. E. (1948). A Mathematical Theory of Communication. *Bell System Technical Journal*, *27*(3), 379–423. http://doi.org/10.1002/j.1538-7305.1948.tb01338.x

[2] Jacques Derrida. (1977). *Limited Inc*. *Context*. http://doi.org/10.1007/s13398-014-0173-7.2

[3] *A Brief History of Information Theory* - https://cs.stanford.edu/people/eroberts/courses/soco/projects/1999-00/information-theory/history.html.

[4] Ward, C. G., (2010) *Stock Imagery, Filler Content, Semantic Ambiguity*

[5] Menkman, R. (2011). *The glitch moment (um)*. *networkcultures.org*. Retrieved from http://networkcultures.org/blog/publication/no-04-the-glitch-momentum-rosa-menkman/

[6] Ballard, S. (2007). Information, Noise and et al. *M/C Journal*, *10*(5). Retrieved from http://journal.media-culture.org.au/0710/02-ballard.php

[7] Mann, Y(2014). *Interactive Music with Tone.js, Web Audio Conference, Paris, France, 2014*

[8] Web Audio API: W3C Editor's Draft 13 May 2016. http://webaudio.github.io. Accessed: 2016-05-13.