University of Louisville

ThinkIR: The University of Louisville's Institutional Repository

1-2022

Clickbait.

Roger Knight
University of Louisville

CLICKBAIT

By

Roger Knight
B.A., University of Evansville, 2007

A Thesis
Submitted to the Faculty of the
School of Music of the University of Louisville
In Partial Fulfillment of the Requirements
For the Degree of

Master of Music
in Music Composition

School of Music
University of Louisville
Louisville, KY

May 2022

CLICKBAIT

By

Roger Knight
B.A., University of Evansville, 2007

A Thesis Approved on

April 27, 2022

by the following Thesis Committee:

_____
Dr. Krzysztof Wolek

_____
Dr. John Ritz

_____
Dr. Kirsten Carithers

ACKNOWLEDGEMENTS

To my wife, for unwavering support and patience.

To my children, for endless, intangible gifts.

To my parents, for love and support at all times.

To Dr. Krzysztof Wolek, for asking, "What is the piece?"

To Dr. John Ritz, for constant mentorship, encouragement, and trust.

To Dr. Kirsten Carithers, for valuable guidance as a member of my thesis committee.

To my colleagues and friends, Isaac Barzso, Tanner Jones, Rron Karahoda, Emery Miles, and Isaac Smith, for regularly inspiration and guidance, belief and confidence in my work, pushing me beyond my expectations, and helping me process the last two years.

Thank you.

ABSTRACT

CLICKBAIT

Roger Knight

April 1, 2022

Clickbait is a surround-sound, audio, video, and audience participation work for live electronics that connects its listeners to the music through the use of a variety of interconnected technologies. The topic of internet clickbait was the initial inspiration that grew into a personal and cultural question. Technological mechanics of live web-based services and data-mining informed those themes further. The interactive nature of internet clickbait drove efforts to establish a deeper connection between the audience and the music. To develop connections between people, applications, and the piece, Node.js, Google Cloud Services, VoiceRSS, and Max/MSP/Jitter video tools were used. This underpins a composition of unique, computer-based music created with modular synthesis, Spitfire Audio's software instruments, and Max's MIDI capabilities showcasing explorations of effects processing using Max's audio tools, Circuit Music Labs' Petra, and IRCAM's SuperVP and Spat5. The implementations of these technologies were in service of a central, conceptual theme allowing Clickbait to transcend beyond the normal limits of an ordinary audio work.

TABLE OF CONTENTS

LIST OF FIGURES

INTRODUCTION

Clickbait is a 21-minute piece of audio and video programming employing indeterminate, audience participation elements. The work is custom software, written primarily in Max/MSP/Jitter (version 8) and Node.js featuring live manipulation of audio and video using digital signal processing, live control of various synthesis methods, and fixed media. At performances of the work, listeners are invited to interact with the piece through a website on any personal device. Data entered by participants are brought into the system, directly controlling the piece while connected web services provide dynamic, A.I.-based text analysis and text-to-speech audio generation. This results in a unique form of internet-connected, A.I.-influenced, crowdsourced, electronic chance music that seeks to explore themes of communication technology, internet connectedness, data-mining, social networks, artificial intelligence, advertising, attention, distraction, addiction, and anxiety. This document will explore the piece in two main sections: the conceptual inspirations and goals, and the technological mechanics and processes.

CHAPTER I:

INITIAL INSPIRATIONS, CHALLENGES, AND GOALS

Internet Clickbait

The most important conceptual seed for the creation of this piece is the phenomenon of internet clickbait. The term, clickbait, describes deceptive and often sensational text, image, and/or video links designed to exploit a person's curiosity and manipulate them into clicking through to a new webpage. The linked content is universally vapid ephemera, often dubious, that exists only to host online advertisements that either sell products or harvest data. If a person is successfully 'baited', the content gets a 'click' and either the advertisements generate revenue for the hosting website, or the data-miner pries some personal data from the user. These types of links are probably familiar to anyone who has ever used a web browser or social network, but they often contain enticing teasers such as "3 Reasons Why…", "How to Achieve Results Using This One Weird Trick", or "… and You Won't Believe What Happens Next!". These taglines are often tied to sensational imagery and promises of exciting reveals (see Figure 1). When this topic first presented itself, the most interesting aspect was the moment in a person's brain when they decide to click, especially if that person knows the link is a sham. What does a sensational indulgence such as internet clickbait give to us? What does it take from us? What are we giving up? The early concept was to ask, with the audience, why we turn to vices that we know are without substance.
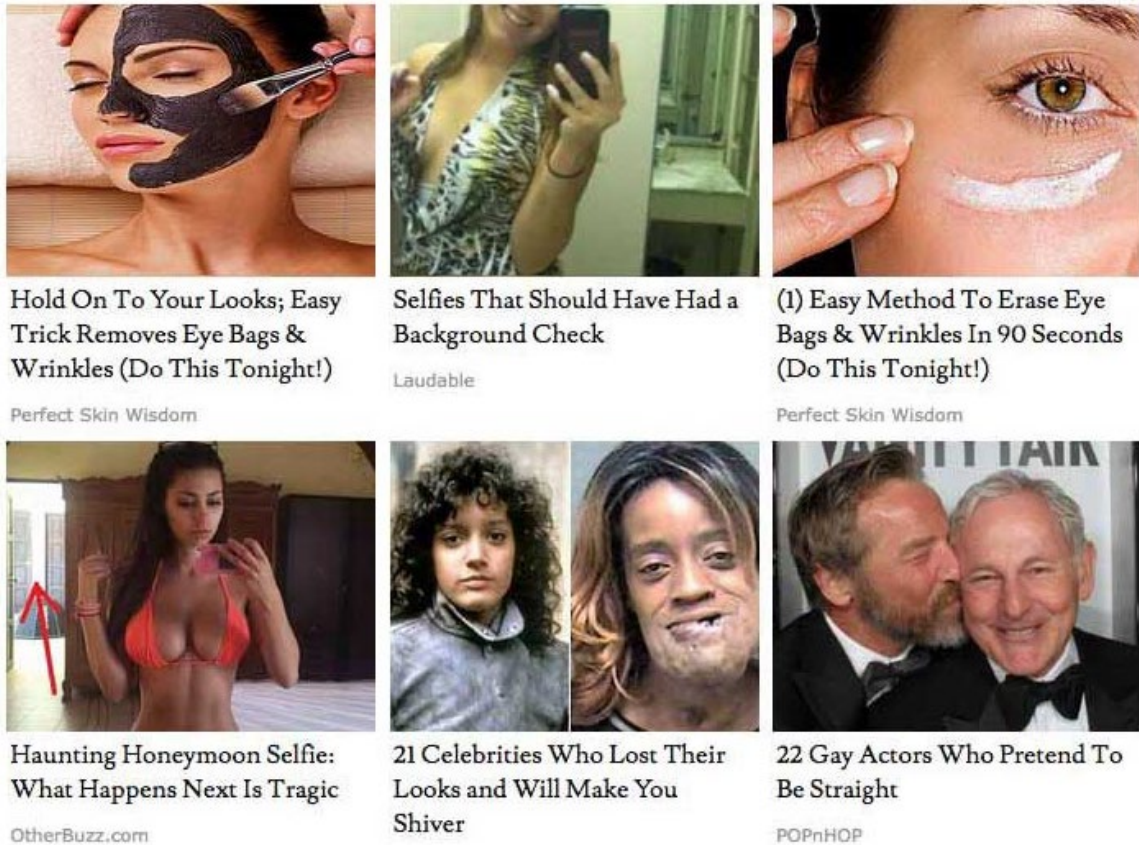
Figure 1. Internet Clickbait

<div align="center">Twitch Plays</div>

Another significant inspirational jumping-off point for this piece was a social

experiment turned cultural phenomenon called "Twitch Plays Pokémon", a popular

channel on Twitch.com, a video game streaming site.[1] In this stream, the host and creator

set up a system which gave a live, active, and notoriously unruly Twitch chat room the

only direct command of the beloved 1996 Gameboy game, Pokémon Red.[2] The game's

control was handed over, virtually, to the anonymous masses of the internet. Through an

---

[1] Twitch. "Twitch Plays Pokémon." Twitch. 12 February 2014. www.twitch.com/twitchplayspokemon.
[2] Nintendo. *Pokémon Red*. Nintendo. Gameboy. 1996.

IRC (Internet Relay Chat) bot written by an anonymous user in Python, the online

audience could participate directly in the stream they were watching by typing particular

commands that corresponded to the full set of controls for the game (up, down, left, right,

start, select, a, b). These chat messages were extracted and immediately input into the

emulator on which the game was being hosted while the commands were displayed

visually in a framed overlay on the stream using a JavaScript web app (see Figure 2).

Figure 2. Twitch Plays Pokémon

When this experiment first launched, no one knew if it would be successful in

either gaining viewers or even progressing in the game. As it turned out, it made for a

fascinating sociological study and a compelling spectator sport. Many would watch and

wonder if the Twitch chat room's hyperactive mob mentality could overcome itself to

guide Pokémon's precarious avatar, Red, through the labyrinthine world, lengthy battles,

and deep menus. Would the limitations of a fully open and internet-connected system

mean that defeat was inevitable, or could a game-ending victory be achieved? Early on, it seemed any and all progress would be foiled by constant "trolls" and trigger-happy participants. Often, nefarious chat users would spam either "down" commands, sending Red off of a ledge, or "start" commands to force the game into its main menus, prohibiting any progress. Despite this, through a combination of massive user coordination, strategy, innovation, and some creator-imposed input-throttling on certain commands, progress was starting to be made.

After two weeks of learning and improving, the participants had finally inched their way through the challenge and completed the game in its entirely with over 55 million total views and an estimated 1.16 million total participants, obliterating the world record for the most participants on a single-player online video game.[3] Twitch Plays Pokémon had solidified its place in internet and gaming pop-culture history and opened the door for countless other "Twitch Plays" derivatives, including more complex chat systems such voting, interactive layers, and hosting of multiple networked games. The earliest concepts of my thesis project had everything to do with this fascinating type of technology and crowdsourcing system and a plan emerged to bring it into the realm of electronic concert music.

Orchestrating the Inaudible

Devising systems and relinquishing control has been a core focus of my work at the University of Louisville, where I've expanded my knowledge with a survey of

[3] Guinness World Records. "Most users to input a command to play a live streamed videogame" Guinness World Records. April 1, 2022. https://www.guinnessworldrecords.com/world-records/most-participants-on-a-single-player-online-videogame/

aleatory and non-linear music as well as generative music and system or process-based music. Notably, composer John Zorn's socio-musical philosophy that he explored in his famous game pieces was of particular interest.[4] The idea that a social dynamic can be a musical element that is equally as important or possibly more important than any other musical component is a compelling concept with which there was much grappling with the creation of *Clickbait*.

Ultimately, there were many divergent ideas which gave the project a wide variety of possible directions (see Figure 3). The most challenging aspect of the work was grappling with how to turn the amorphous collection of ideas that grew around the initial categories of "crowdsourcing" and "internet-connected" and condense them into a rock-solid project robust enough for significant development.

Figure 3. Early Conceptual Word Cloud

---

[4] John Brackett. 2010. "Some Notes on John Zorn's Cobra." *American Music* 28 (1): 44–75.

The structural heart of the piece evolved to be two parts: a technological connection between the audience and the piece via the internet; and the act of the audience engaging in this connection. It was clearly important to make it evident to the listener that their interactions were affecting the music, but devising a full and complex system that mapped every input from the audience onto a meaningful and recognizable musical element proved to be an impossible task. In fact, it became clear that it was not important to create and maintain direct mechanical connections for all of the audience's interactions. Instead, a relationship could be established early in the piece, and represented later in any variety of ways. Furthermore, the idea of an audience making choices that they believe may have an impact, when in fact they do not, ended up becoming crucially connected *Clickbait*'s central themes.

The plan would be to coerce the audience to look at and interact with their phones during the piece's presentation while making a statement about the internet and about the audience looking at their phone during the piece. The idea was to play with people's attention and manipulate their expectations by representing the farcical nature of our modern internet landscape in a musical structure, thereby 'baiting' the audience. If the audience's assumptions about mechanics of the connection between them and the music is uncertain or incorrect, then the piece's themes are reinforced even further. In this way, the music that has been orchestrated is not <u>for</u> a distracted, disillusioned, digitally connected audience, but <u>of</u> a distracted, disillusioned, digitally connected audience.

CHAPTER II:

STRUCTURE AND TECHNOLOGY

Framework and Automation

Initial ideas for the crucial audience participation element involved a custom website with interactive graphics, timed events, pop-up ads, monetization, voting, and a bespoke micro-social-media network. This very broad approach needed to be scaled down in order to made a clear and effective artistic statement as well as manageably fit within the time constraints and overall scope of a master's thesis project. Based on previous experience in this program, complexity simply isn't necessary to make successful or significant compositions. Ultimately, the decision was made to simplify and utilize ubiquitous and reliable web technologies such as Google. This allowed more time to be focused on other aspects of the composition, technologies, and research during development.

The macro view of the piece reveals a linear form with six sections composed of fixed media, live electronics, or both with three main instances of audience participation. The decision was made to keep the form linear and controllable and utilize the audience for a kind of solo instrumentation that would sit on top of musical elements that lay underneath.

The overall framework was devised and created within Max/MSP/Jitter,[5] placing the main controls and layout on the front page of the application (see Figure 4). While the functionality of Max was greatly extended using myriad third-party tools, the main audio structure resides solely in Max and is found in the blue section on the right. At its core is a custom digital audio matrix mixer with seven outputs and eleven inputs. The seven outputs lead to both the main output and six effect sends while the eleven inputs come from six generator modules as well as five effect returns. This mixer was implemented using the native Max object [matrix~][6] and controlled with [matrixctrl] set in "dial mode" with a cell range of 128 to allow fine control of all audio routing and mixing in the program from a single object. Additionally, a custom cue system was created using [pattr] to exclusively control all aspects of the mixer at every cue.

Figure 4. Application Layout

5 Cycling '74. "What is Max?." Cycling '74. April 1, 2022. www.cycling74.com/products/max.
6 All references to native Max/MSP/Jitter objects are denoted with square brackets

This system allows a full mixer arrangement to be recalled at any moment with the smooth, time-based interpolations between cues, making the composition process for transitions between routing setups incredibly flexible. There are 36 cues for this [pattr] originating from 24 distinct cue modules that are triggered automatically from a global transport system. These cue modules were intentionally kept in a separate system from the [pattr] that control audio routing because they also contain all other controls and triggers for every aspect of the sound and interactive elements throughout the piece (see Figure 5).
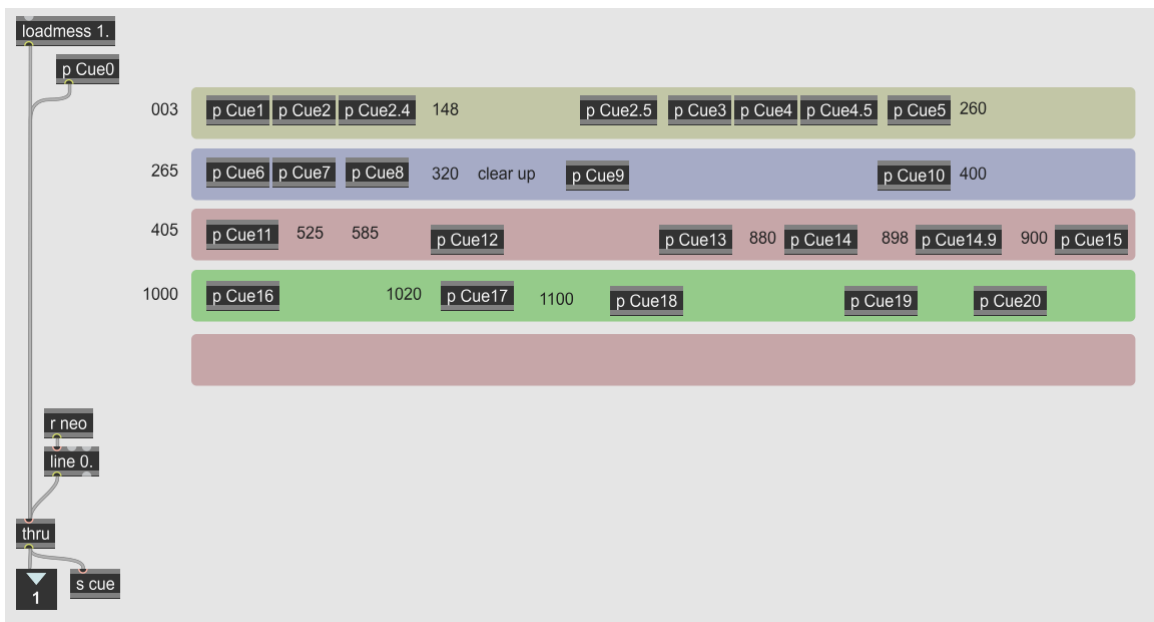
Figure 5. Main Cue System

There is also another smaller, discrete cue system which was created for the video automation (see Figure 6). Having all of these systems separated was helpful in creating a clear division between mixing automation, audio automation, and video automation, but it also required additional attention in managing the three different systems.

Figure 6. Video Cue System

Generators

The bedrock components of most of musical underlayment in the piece originates

from the subpatches *Chords, Strings,* and *Hum*.[7] *Chords* is generator module composed

of algorithmic MIDI commands[8] that drive an array of four virtual instruments (VSTs)[9]

from Spitfire Audio's Labs series: Textural Pads, developed by Mark Summerell by

sampling bass guitar, kalimba, cello, vocals, and synthesizers, and processing them with

granular effects; and Monochord in three variations (Strum, Bow, and Paddle Detune), a

collection sampled from Stephen Warbeck's unique microtonal instrument.[10] These two

instruments have opposing effects whereby one lulls listeners into a hypnotic comfort and

the other serves as a more visceral and grating interruptive element.

---

[7] All references to custom Max modules are denoted with italics.

[8] Musical Instrument Digital Interface: A digital communication protocol that carries musical information such as pitch, duration, velocity, and clock

[9] Virtual Studio Technology: An audio plug-in software interface that integrates software synthesizers and effects units into digital audio workstations.

[10] Spitfire Audio. "Free Labs." Spitfire Audio. April 1, 2022. labs.spitfireaudio.com.

The algorithms for these instruments have randomization and will never play exactly the same thing twice, but will also never be too different due the high level of control built into them. Textural Pads creates semi-randomly chords that are sometimes pan-diatonic, sometimes very chromatic. These play at regular time and duration intervals, with rotating options for amplitude and expression envelopes, which yields constantly moving variations of intensity. The range and set of available pitches for these semi-random chords is highly controlled and orchestrated throughout the piece (see Figure 7). From 17:55 to 20:00 the pitch range can be heard as it decreases and lowers over time, leaving all but the most rumbling, low pads.
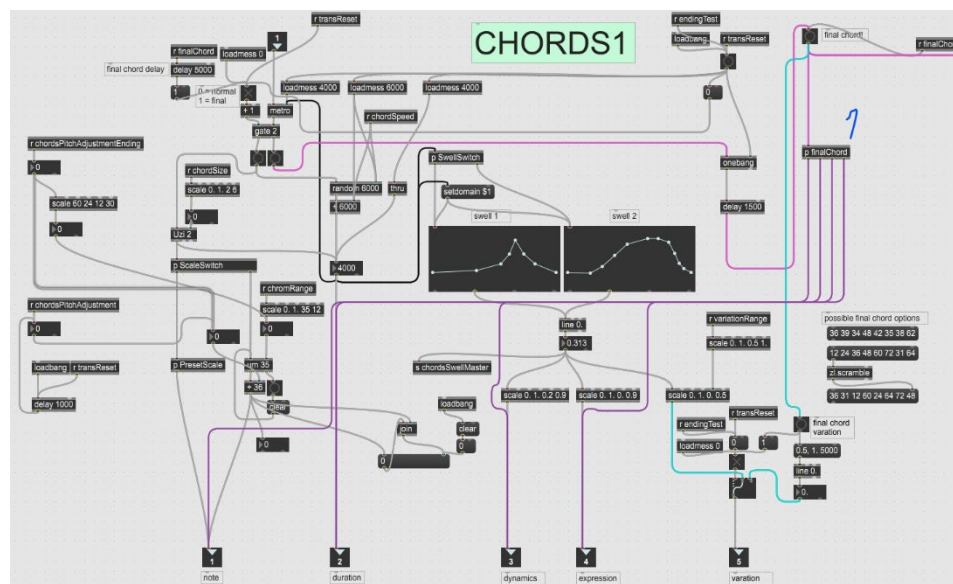

Figure 7. Textural Pads Algorithms

The Monochord instrument was chosen for its dry and harsh color that contrast the Textural Pads. The pitches of the instrument are again semi-random, selected from the same pool as the Textural Pads, orchestrated by the cue system. Strummed and bowed versions can be heard clearly from 10:40 to 11:20.

Strings is very similar to *Chords* in that it is utilizing VSTs from Spitfire Audio controlled by custom algorithms. The instrument used is the highly detailed, sample-based BBC Symphony Orchestra, which provided access to between 16 and 20 different articulations with multiple recordings each for each instrument of the orchestra. The *Strings* module only uses the string section of the orchestra, but has a variety of different algorithms controlling the creation of musical material. A key element of the computer-based control was switching very frequently between short articulations in the basses and celli while the violas and violins are being guided by sharp swells using either sul ponticello, tremolo sul ponticello, or tremolo con sordino. These algorithms can be heard at 12:20 – 15:00.

Looking into the *Hum* module reveals a simple audio player for four-channel fixed media, converting from stereo to multi-channel when needed. This player was one of the first things to be implemented and helped immensely in defining the proportions of the piece and establishing guide posts for all other material and automation.

Some of these fixed media elements were created using hardware modular synthesis. Many minutes of recordings were made in an improvisatory manner with certain goals in mind. Sections of these recordings were processed and edited to sit in place within Clickbait. A key feature of these modular sections, heard clearly at 0:00 – 2:00, is the use of the Mimeophon delay module from modular synthesizer manufacturer, Make Noise.[11] This device is a stereo, multi-zone audio repeater programmed by Tom Erbe of Soundhack which allows a high level of control over feedback and different tone colors in a stereo delay line (see Figure 8).

---

[11] Make Noise Co. "Make Noise Mimeophon." Make Noise. www.makenoisemusic.com/modules/mimeophon.
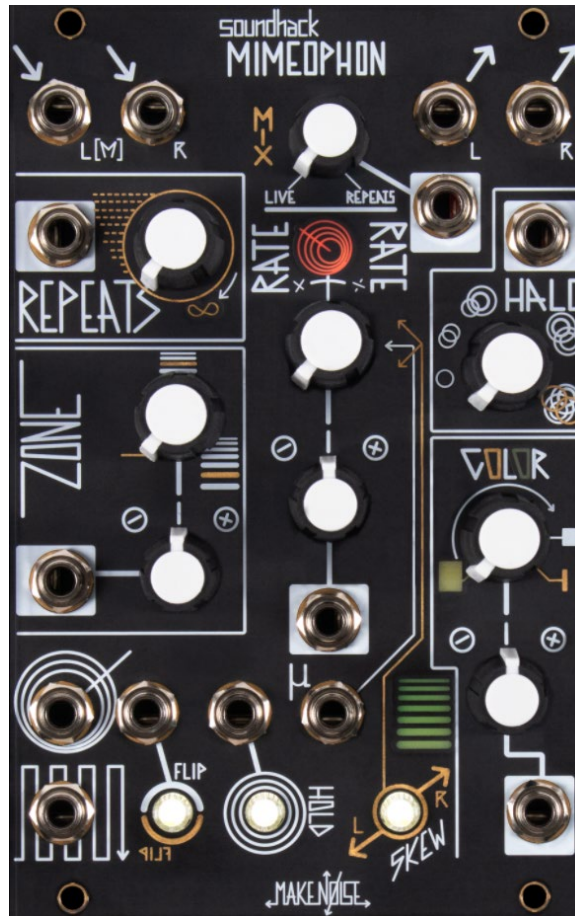
Figure 8. Mimeophon by Make Noise

The other fixed media material used in *Hum* was created using an original feedback tool built by the author in 2020 within Max as a part of a surround-sound, delay study referred to as the *Madness Engine* (see Figure 9). This is a four-channel audio processor utilizing daisy-chained, rotational delay lines with complex modulation of the integrated gain control as well as delay times and a simple user interface to warp the very nature of the modulation mappings. This material can be heard clearly from 5:40 – 6:35. Both of the feedback methods mentioned above strongly feature precarious feedback control and occasional lack of control, which emerged as a thematic element throughout the piece. This connection speaks to the internet throughout its history being a wild

cacophony of feedback, despite often being presented as either organized or fully controlled.
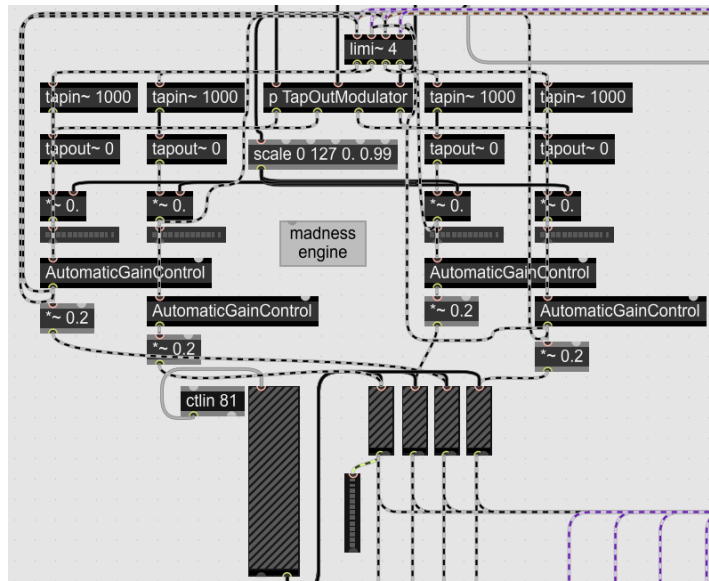
Figure 9. Madness Engine

Crunch is another generator that is simply an audio player, but with more nuanced modulation and playback manipulation of the audio. This player is built more for specific sounds functioning and presenting more as solo instruments and transitional material than the background pad structure of audio being created in *Hum*. Source material here is original recordings of mechanical gears, failing hard drives, and a 56kbs modem connection. Gears and hard drives can be heard at 4:07 – 4:33 and repeating waves of the modem sound with evolving speed and pitch manipulation can be heard at 19:14 – 20:25.

Notification is a generator that uses an implementation of the native Max object [poly~] to play random notification sounds from various popular devices and networks such as iPhone, Android, Facebook, and Google, to name a few. The [poly~] scheme draws from a folder of audio files and plays each file spatially at a random azimuth and

distance. These sounds happen at rates ranging from once every ten seconds to 20 per second, either distracting the audience with the sound of their digital lives or immersing them in it.

<div align="center">Voicing the Internet</div>

The last remaining generator to discuss is in fact the crux of the piece. The *Voice* module is in charge of playing back audio created from the text entered by the audience in the interactive components of the piece. However, the generator itself is very simple and similar to *Notification* in that it plays audio files from a folder and spatializes them. The function of producing the audio files lies in the green section of the main patch layout (see Figure 3). The main networking structure is two instances of [node.script], a Max object that allows control of a local Node.js[12] process from within Max. Node is an asynchronous event-driven JavaScript runtime designed for network applications, which provides Max access to the outside world, namely web services such as Google.

Outlining the details of an example of the flow of time and information visualizes the logic in this process (see Figure 10). When the piece reaches an elapsed time of 65 seconds, a change in the video programming is triggered, presenting the first QR code to the audience. Audience members have 85 seconds until an elapsed time of 150 seconds is reached and another change in the video programming is triggered, removing the QR code. When the audience visits the link associated with the first QR code, they are presented with a Google Forms quiz with three questions, responses of which are logged into a Google Sheets spreadsheet. At an elapsed time of 150 seconds, Cue 3 triggers

---

[12] OpenJS Foundation. "Node.js." Node.js. April 1 2022. www.nodejs.org.

changes in the mixer and audio routing, some effect automation, and sends a message to a

JavaScript application running in a [node.script] instance to use Google's API (Advanced

Programming Interface) to poll a particular spreadsheet for the contents of column "C"

from row "2" until it reaches an empty row (see Appendix A). This gives the script all of

the responses to the third question from the first quiz, which it loads into a message and

sends the responses, as a list of individual strings of text, back to Max. A moment after

this happens, Cue 3 also starts a module called *voiceInternet* within Max which, at a

given rate ranging between once per ten seconds and 20 per second, selects a random

item from the list of responses, and sends it as a message to a separate JavaScript

application running in a separate [node.script] instance that sends this text back out of

Max to a web service called VoiceRSS via its own API (VoiceRSS.com). VoiceRSS is a

web service that allows conversion of textual content to high quality speech in the form

of audio files. The custom script that sends the text to the VoiceRSS service is bundled

with a download command as well as randomization of various parameters that result in

"speaker personalities" made up of different vocal qualities, ages, and nationalities (see

Appendix B). The resultant file is stored in a subfolder within the main project directory

where Max can access it. A moment after this, the file is played and spatialized through

the *Voice* generator module while Sheets.js continues to poll the spreadsheet every ten

seconds for additional submissions that may have been added late.
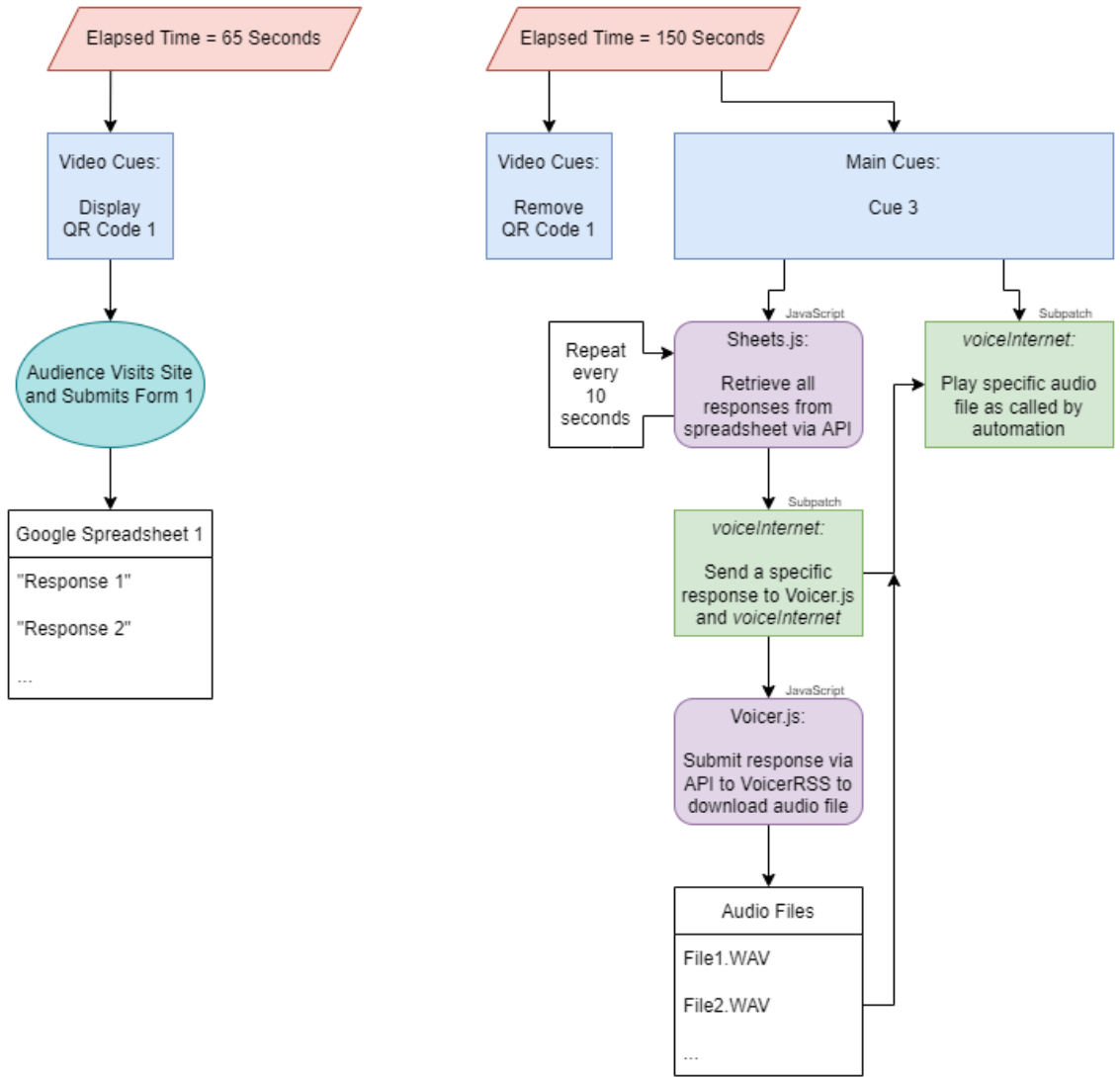
Figure 10. Audience Participation Example Flow Chart

To review, the audience sees a QR code and fills out an online survey. These
responses fill a spreadsheet and Max uses scripts running in Node to access that
spreadsheet and retrieve its data. Max then uses another Node script to send a portion of
that data to VoiceRSS and download audio files to be played. Finally, Max plays and
spatializes the files per cue automation. This all happens three times during the piece with
three distinct quizzes and automation.

In orchestrating these moments, considerations were made for many things: what questions populate the particular quiz; what music the QR code was interrupting; how long listeners had been sitting in the previous section; how long the QR code is visible on screen; how long before the application can poll the spreadsheet to attempt to retrieve data; into what music does the quiz section lead; and are the culminations of these orchestrations in service of the statements being made by the totality of piece. The full, collected spreadsheets remain accessible and provide some interesting insights into this piece (see Appendix C). Ideally, listeners, through their interaction with the quizzes and time with long periods of stagnant musical stasis, are faced with a question about their relationship with the internet and its evolution.

<div align="center">Effects</div>

In addition to the six generators creating sound, there are five effects that are tasked with processing sound in different ways through the piece. The reverb that was used is a flexible pre-built tool pulled from a software suite developed by IRCAM (Institut de Recherche et Coordination Acoustique/Musique) called Spat5.[13] At different times in the piece, the reverb parameters are either automated in very dynamic ways or left alone to simply create an acoustic space and sound enhancement. Reverb "warble" and "overflow and snap shut" moments populate the section between 12:10 and 15:00, which required extensively integrated automation to be created (see Figure 11). Late in the process of preparing the work for a concert setting presentation, a control was added to the main patch layout that provided a single, quick adjustment of all of the reverb

---

[13] Institut de Recherche et Coordination Acoustique/Musique. "IRCAM." IRCAM. April 1, 2022. www.ircam.fr.

automation which allows *Clickbait*'s reverb effects to be "tuned" to accommodate the difference in the acoustics of various performance environments.
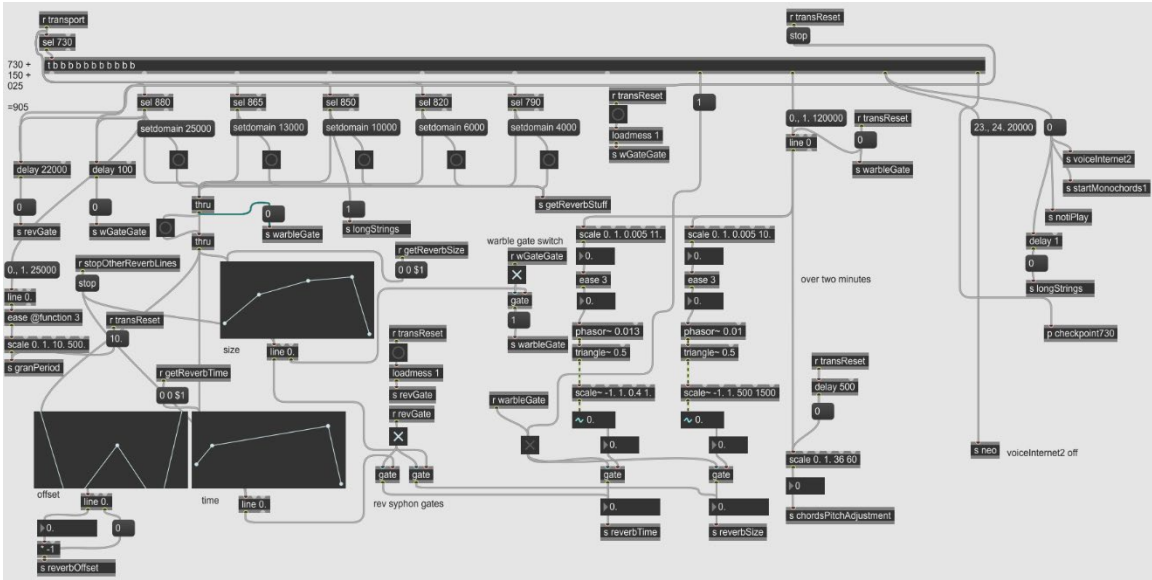
Figure 11. Selection of Reverb Automation

Gran1 and *Gran2* are both live granulation processors and are built from two different granulation algorithms. The first is a new dynamic granulation-spatialization tool included in the aforementioned Spat5 software suite. This effect cuts and distributes mono audio streams to a multichannel audio output. The second granulation tool was LiveCloud from the Petra collection developed by Matthias W. Müller at Circuit Music Labs.[14] LiveCloud works similarly, but has a simpler control scheme than its Spat5 counterpart. The decision to use two different systems was based on separate musical needs and the organizational benefit that comes from having separate modules. Petra was chosen for its ease and familiarity, whereas Spat5 was chosen as a research topic.

---

[14] Matthias Müller. "Circuit.Music.Labs." Circuit Music Labs. April 1, 2022. https://cmlabs.wordpress.com

The live stutter effects created in *Stutter* were created using custom logic and a simple, native Max object called [stutter] which keeps a history of an incoming signal and can call back portions of that signal and replay them at a specified rate. The algorithm driving the semi-random modification of the original signal was directly inspired by the now-classic vocal processing that is applied to the character of Shodan in the 1999 video game *System Shock 2* (see Figure 12), an action game about a sentient A.I. who has gone rogue.[15] It was thematically important to replicate this specific process accurately and connect our audience's voices to an early-internet-era understanding of artificial intelligence help bend the piece toward a conceptual connection to the broader categories of internet past, present, and future. This aural, future-retro processing can be heard applied to voices from 3:30 – 4:20, and to instruments from 15:05 – 16:10.

Figure 12. System Shock 2's Sentient A.I., Shodan

---

[15] Irrational Games Looking Glass Studios. System Shock 2. Electronic Arts. PC. 1999.

Finally, the *CrossSynth* effect has two inputs and one output. Through the mixer, the effect is passed a 'source' signal and a 'filter' signal and within the subpatch, these two signals are combined in another tool called SuperVP (Super Phase Vocoder), from a different team at IRCAM. This high quality, active phase vocoder allows for extremely adjustable, real-time transformation of the 'source' signal based on the 'filter' signal. While the piece utilizes this effect in a variety of ways, the clearest example can be heard at 16:48 – 19:10, when our audience's voices are convolved through cross-synthesis with modern notification sounds. Here, the voices are heard clearly, keeping their dynamic and transient character, but they have taken on the timbral character of the digital notification sounds that have become so connected to our lives in every way. At this point in this piece, our digital lives have begun to even be connected to the sound of our voices.

Video

While distributing QR codes to an audience could have been accomplished in any number of ways, video playback with integrated, time-aligned automation was developed. This provided a straightforward coordination with the linear flow of the piece, which ensured everything would line up predictably and yielded a significant added benefit of having another sensory layer of the piece. This layer was orchestrated along with the aural and interactive components in an effort to envelop the audience in the work.

The colored screens were created as static images in Adobe Photoshop and depict simple, framed rectangles that fill the screen. The changes in color help to outline new sections and also provide visual cues to the audience that there is a new QR code present

that can be accessed. The changes serve as a sneaky score that the audience may not realize they are following.

The goal with the design of these images was to harken back to the monochromatic monitor displays of the early computer era. The zoom effect that bookends the piece is highly inspired by Pierre Jodlowski's *Outerspace* for trombone, video, and electronics.[16] In that piece, Jodlowski viscerally explores video screens, their borders, the space held within them, perspective, and movement, including a particular effect where a sudden zoom into a video screen brings the audience to a new section of the music, including a drastically different video and lighting component (see Figure 13). This moment, and the piece at large, was extremely influential in the design of *Clickbait*'s video aspects.

Figure 13. Pierre Jodlowski's *Outerspace*

16 Jodlowski, Pierre, composer. *Outerspace*. Pierre Jodlowski Publishing, 2018.

Finally, a custom visual-distortion effect created in Jitter, Max's native video processing toolset, was added to the frame images at four major moments in the piece. The effect rises smoothly, adding a slight, billowing wobble to the outer frames of the rectangles, and dissipates again. At first, the effect is extremely small and may go unnoticed by the audience, but each time the distortion returns, it is more drastic and noticeable. In the final moments of the piece, the effect doesn't stop rising and continues to undulate in newly dramatic fashion. Simultaneously, the zoom effect that began the piece comes back and smoothly pushes the audience's perspective into the frame. Musical material at this point is a repeating and continually stretching sample of a 56kbs modem dialup procedure which aligns with the visual distortion and zoom effect vectors, serving as a recapitulation of the gray frame in the opening (see Figure 14). All of this together takes listeners further into the piece and takes the conceptual aspects of the piece further into ideas of the arc of the internet across history and into the future, and importantly, our connection to that arc.
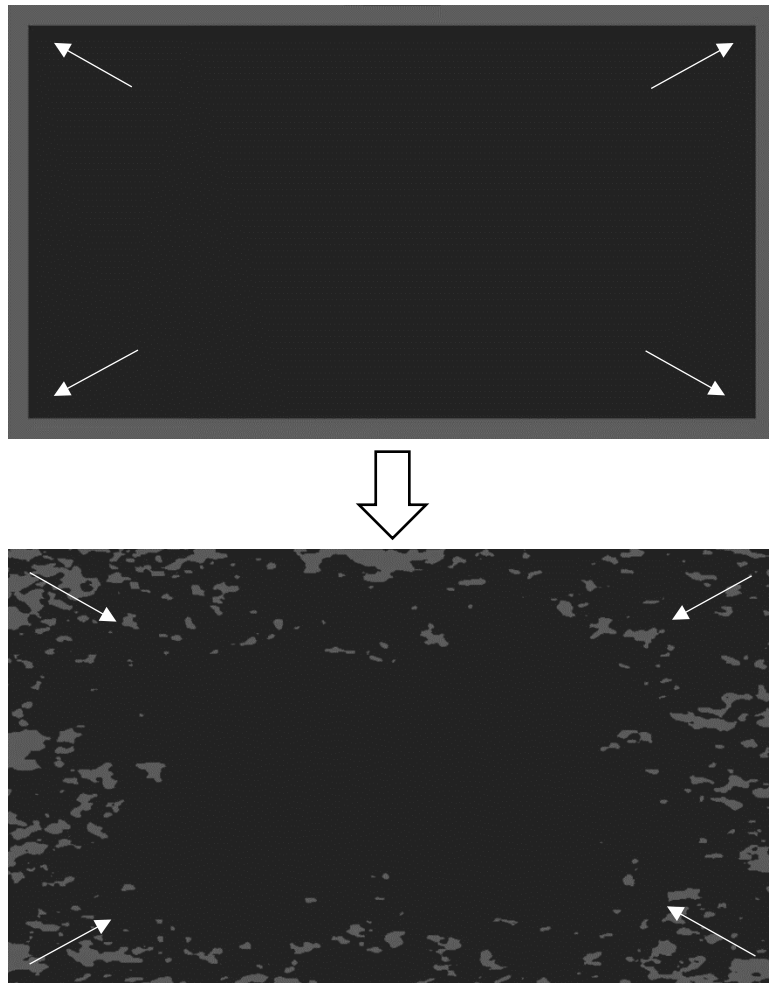
Figure 14. *Clickbait* Video Start and Ending Comparison

SUMMARY

Clickbait involved many tools in its creation, but the conceptual aspects always guided development. Crowdsourcing in the spirit of collective gaming events added a captivating feature to a typical concert experience and entrusted some aspects of the piece to the audience. The insatiable curiosity of the human mind was featured in the design of the composition. However, whether or not this inquisitiveness was utilized, rewarded, or ignored by the piece may remain a mystery to the audience. This empowerment and partial deception of the audience runs through the composition of attention and distraction. Participants experienced direct yet unclear feedback from their interactions while their focus was manipulated and orchestrated as a musical element. Video effects were used to cue the audience's behavior. These signals indicated times to relax, take action, or question. While this work could have sounded similar if created as fixed media in a traditional audio workflow, it would have neglected the exploration of human psychology and sociology woven by a dynamic interplay that necessitated the implementation of intersecting technologies. *Clickbait* sits comfortably in the cross section between many media and concepts, which allows the work to powerful convey a unique, trans-aural, electronic arts experience.

REFERENCES

Brackett, John Lowell. 2010. "Some Notes on John Zorn's *Cobra*." *American Music* 28

    (1): 44–75.

Cycling '74. "What is Max?." Cycling '74. April 1, 2022.

    www.cycling74.com/products/max.

Guinness World Records. "Most users to input a command to play a live streamed

    videogame" Guinness World Records. April 1, 2022.

    https://www.guinnessworldrecords.com/world-records/most-participants-on-a-

    single-player-online-videogame/

Institut de Recherche et Coordination Acoustique/Musique. "IRCAM." IRCAM. April 1,

    2022. www.ircam.fr.

Irrational Games Looking Glass Studios. *System Shock 2*. Electronic Arts. PC. 1999.

Jodlowski, Pierre, composer. *Outerspace*. Pierre Jodlowski Publishing, 2018.

Knight, Roger, composer. *Clickbait*. 2022.

    https://www.youtube.com/watch?v=EueaP9sZSQA

Make Noise Co. "Make Noise Mimeophon." Make Noise.

    www.makenoisemusic.com/modules/mimeophon.

Nintendo. *Pokemon Red*. Nintendo. Gameboy. 1996.

OpenJS Foundation. "Node.js." Node.js. April 1 2022. www.nodejs.org.

Spitfire Audio. "Free Labs." Spitfire Audio. April 1, 2022. labs.spitfireaudio.com.

Twitch. "Twitch Plays Pokémon." Twitch. 12 February 2014.

   www.twitch.com/twitchplayspokemon.

Voice RSS. "Voice RSS." Voice RSS. April 1, 2022. www.voicerss.org.

Müller, Matthias. "Circuit.Music.Labs." Circuit Music Labs. April 1, 2022.

   https://cmlabs.wordpress.com

APPENDIX A

Sheets.js

```javascript
const maxApi = require('max-api');
const{google} = require('googleapis');
const { auth } = require('google-auth-library');
const express = require('express');
const app = express();

//get full spreadsheet
maxApi.addHandler('getSpreadsheet', async () => {
    const auth = new google.auth.GoogleAuth({
        keyFile: "***---REDACTED---***",
        scopes: "https://www.googleapis.com/auth/spreadsheets",
    });
    const client = await auth.getClient();
    const googleSheets = google.sheets({ version: "v4", auth: client });
    const spreadsheetId = "***---REDACTED---***";";
    const spreadsheet = await googleSheets.spreadsheets.values.get({
        auth,
        spreadsheetId,
        range: "Sheet1",
    });
    maxApi.outlet(spreadsheet.data);
});

//get single row
maxApi.addHandler('getSingleRow', async (row) => {
    const auth = new google.auth.GoogleAuth({
        keyFile: "***---REDACTED---***",
        scopes: "https://www.googleapis.com/auth/spreadsheets",
    });
    const client = await auth.getClient();
    const googleSheets = google.sheets({ version: "v4", auth: client });
    const spreadsheetId = "***---REDACTED---***";";
    const singleRow = await googleSheets.spreadsheets.values.get({
        auth,
        spreadsheetId,
        range: "Sheet1!A"+row+":Z"+row,
    });
    maxApi.outlet(singleRow.data);
});

//get single column
```

```javascript
maxApi.addHandler('getSingleColumn', async (col) => {
        const auth = new google.auth.GoogleAuth({
        keyFile: "***---REDACTED---***",
        scopes: "https://www.googleapis.com/auth/spreadsheets",
    });
    const client = await auth.getClient();
    const googleSheets = google.sheets({ version: "v4", auth: client });
    const spreadsheetId = "***---REDACTED---***";";
    //change col from integer to string 1 = A, 2 = B, ...
    let alphaVal = String.fromCharCode(64 + col)
    const singleColumn = await googleSheets.spreadsheets.values.get({
        auth,
        spreadsheetId,
        range: "Sheet1!"+alphaVal+"1:"+alphaVal+"104",
    });
    const values = singleColumn.data["values"];
    const valuesArray = values.map(value => {
        return value[0].toString();
    })
    maxApi.outlet(valuesArray);
});

//get single cell
maxApi.addHandler('getSingleCell', async (col, row) => {
    const auth = new google.auth.GoogleAuth({
    keyFile: "***---REDACTED---***",
    scopes: "https://www.googleapis.com/auth/spreadsheets",
    });
    const client = await auth.getClient();
    const googleSheets = google.sheets({ version: "v4", auth: client });
    const spreadsheetId = "***---REDACTED---***";
    //change col from integer to string 1 = A, 2 = B, ...
    let colAlpha = String.fromCharCode(64 + col)
    const singleCell = await googleSheets.spreadsheets.values.get({
        auth,
        spreadsheetId,
        range: "Sheet1!"+colAlpha+row+":"+colAlpha+row,
    });
    maxApi.outlet(singleCell.data.values[0][0]);
});
```

Voicer.js

```javascript
const http = require('http'); // or 'https' for https:// URLs
const fs = require('fs');
const maxApi = require('max-api');
const { StringDecoder } = require('string_decoder');

var myKey = """***---REDACTED---***";

maxApi.addHandler('setKey', (key) => {
    myKey = key;
});

maxApi.addHandler('say', (words) => {
    const urlWords = encodeURI(words)
    const fileWords = words.replace(/
/g,"_").replace(/\./g,"").replace(/\!/g,"").replace(/\?/g,"");
    fileName = "tts_"+fileWords+".mp3"
    if (fs.existsSync(fileName)) {
        maxApi.outlet("exists "+fileName);
        return;
  }
    var errorMessage = "";
    var isError = false;
    const file = fs.createWriteStream(fileName);

    const allLangs = ['en-au','en-ca','en-gb','en-in','en-ie','en-us',];
    const auVoice = ['zoe','isla','evie','jack'];
    const caVoice = ['rose','clara','emma','mason'];
    const gbVoice = ['alice','nancy','lily','harry'];
    const inVoice = ['eka','jai','ajit'];
    const ieVoice = ['oran'];
    const usVoice = ['linda','amy','mary','john'];

    var currentLang;
    var currentVoice;
    var currentSpeed;

    //randomize language (currentLang)
    var rand = Math.floor(Math.random()*allLangs.length);
    var currentLang = allLangs[rand];

    //randomize voice dependant on language (currentVoice)
```

```javascript
    if (currentLang == "en-au") {
        var rand = Math.floor(Math.random()*auVoice.length);
        var currentVoice = auVoice[rand];
    } else if (currentLang == "en-ca") {
        var rand = Math.floor(Math.random()*caVoice.length);
        var currentVoice = caVoice[rand];
    } else if (currentLang == "en-gb") {
        var rand = Math.floor(Math.random()*gbVoice.length);
        var currentVoice = gbVoice[rand];
    } else if (currentLang == "en-in") {
        var rand = Math.floor(Math.random()*inVoice.length);
        var currentVoice = inVoice[rand];
    } else if (currentLang == "en-ie") {
        var rand = Math.floor(Math.random()*ieVoice.length);
        var currentVoice = ieVoice[rand];
    } else if (currentLang == "en-us") {
        var rand = Math.floor(Math.random()*usVoice.length);
        var currentVoice = usVoice[rand];
    }

    //randomize speed (currentSpeed)
    var currentSpeed = Math.floor(Math.random() * 9) - 6;

    const request =
http.get("http://api.voicerss.org/?key="+myKey+"&hl="+currentLang+"&c=WAV&
f=48khz_16bit_mono&v="+currentVoice+"&r="+currentSpeed+"&src="+words,
function(response) {
        response.pipe(file);
        response.on('data', function (chunk) {
            const decoder = new StringDecoder('utf8');
        isError = decoder.write(chunk).startsWith("ERROR")
        errorMessage = decoder.write(chunk)
    });
    response.on('end', function () {
      if(isError){
          maxApi.outlet("error "+errorMessage);
      }else{
            maxApi.outlet("success "+fileName);
      }
      console.log("isError: "+isError);
    });
  });
});
```

Clickbait Quiz Responses From 04/02/2022

<u>Quiz 1</u>

Timestamp	Do you use the Internet?	Is the internet good?	What is the internet?
2/9/2022 15:38:34	Yes	Yes	Ha ha ha what kind of question is it?
2/9/2022 19:59:25	Yes	Yes	Spectrum
2/9/2022 19:59:39	Yes	Yes	Connection
2/9/2022 19:59:40	Yes	No	a giant orange dinosaur
2/9/2022 19:59:41	Yes	Yes	The internet.
2/9/2022 19:59:42	Yes	Yes	Pipes and tubes
2/9/2022 19:59:43	Yes	Yes	Electronic info cloud
2/9/2022 19:59:47	Yes	Yes	Source to connect people to information
2/9/2022 19:59:47	Yes	Yes	A good thing
2/9/2022 19:59:50	Yes	Yes	Alot of things, man
2/9/2022 19:59:51	Yes	Yes	The void of all things
2/9/2022 19:59:52	Yes	Yes	Information
2/9/2022 19:59:53	Yes	Yes	Everything
2/9/2022 19:59:53	Yes	Yes	The bean internet
2/9/2022 19:59:53	Yes	Yes	Pretty sure magic? A lot of cables?
2/9/2022 19:59:56	Yes	Yes	Wouldn't you like to know
2/9/2022 19:59:59	Yes	Yes	A series of tubes
2/9/2022 19:59:59	Yes	No	A waste of time, and also our biggest connecter.
2/9/2022 20:00:02	Yes	Yes	A strange enigma that no one can tame
2/9/2022 20:00:07	Yes	Yes	A large base of information

2/9/2022 20:00:07	Yes	Yes	Good
2/9/2022 20:00:12	Yes	Yes	What connects people
2/9/2022 20:00:12			Bait
2/9/2022 20:00:16	Yes	No	A whole mess
2/9/2022 20:00:30			Dangerous enterprise
2/9/2022 20:00:33	Yes	Yes	Ni jota
2/9/2022 20:00:35	Yes		Disembodied consciousness
2/9/2022 20:00:38	Yes	Yes	Advanced
2/9/2022 20:00:41	Yes	Yes	Connected computers
2/9/2022 20:00:44	Yes	Yes	Humanity both compressed and enabled.
2/9/2022 20:00:45	Yes	Yes	A SERIES OF TUBES
2/9/2022 20:00:45	Yes	No	Roger
2/9/2022 20:00:48	Yes	Yes	A wireless storage of information about anything and everything, that you can do anything on
2/9/2022 20:00:59	Yes	Yes	Descended from beings of infinite faculty
2/9/2022 20:01:31	Yes	Yes	Access to knowledge among other things
2/9/2022 20:02:12	Yes	Yes	The best thing ever created
2/28/2022 19:53:11	Yes	Yes	The true enigma
2/28/2022 22:21:01	Yes	Yes	Information shrouded in noise. And Bitcoin.

Timestamp	What does the internet tell you?	Why?
2/9/2022 15:43:46	i don't know	
2/9/2022 15:44:18	Truth	
2/9/2022 20:04:50	Nothing	There's nothing else better to do.
2/9/2022 20:04:55	Nothing I need, everything I want	It's easy
2/9/2022 20:05:03	Information	
2/9/2022 20:05:03	Lies	
2/9/2022 20:05:10	Nothing important	Because life sucks sometimes
2/9/2022 20:05:14	Anything I want	
2/9/2022 20:05:19	Anything I want	Because people are people…
2/9/2022 20:05:20	Who to be	They told me to
2/9/2022 20:05:23	Whatever it wants	Wouldn't you like to know
2/9/2022 20:05:24	Anything I want	Existential crisis mode
2/9/2022 20:05:28	Answers	Mind numbing
2/9/2022 20:05:28	Information	It is easy than dealing with the problem
2/9/2022 20:05:29	Really important things ok???	Working it out in therapy.
2/9/2022 20:05:29	What other people think they are	Because there is an endless well of information to drown in
2/9/2022 20:05:29	Information	It can be helpful
2/9/2022 20:05:29	Everything	Shrug
2/9/2022 20:05:33	Whatever I want	I've grown accustomed to constant stimulation.
2/9/2022 20:05:34	information and misinformation and disinformation	I've gotten so used to it i dont even remember...
2/9/2022 20:05:37	The weather	
2/9/2022 20:05:39	That the world is terrible	Instant distraction

2/9/2022 20:05:51	Some combination of whatever I need it to and whatever it wants to show me	It's cheap in the world of attention economy
2/9/2022 20:05:57	Information that I would like to learn	The internet can show me things to cheer me up when I need it
2/9/2022 20:06:04	Humor, work more, do something else	Mindless mindfulness 🧘
2/9/2022 20:06:14	Echos of past futures	Because of Covid and coronavirus safety
2/9/2022 20:06:15	Life is hard	Because it's mindless
2/9/2022 20:06:16	Whatever i want and alot i dont want	I love my puter it makes me smarter and lets me do the things i want more efficiently and effectively
2/9/2022 20:06:20	Wouldn't you like to know	Escape from reality
2/9/2022 20:06:21	Lots of noise	Easy to do
2/9/2022 20:06:22	What I want it to	Bordem is tough
2/9/2022 20:06:25	Sweet nothings, told by an idiot	Sometimes I forget to be my best self.
2/9/2022 20:06:27	Humans are self-absorbed	Because it's there
2/9/2022 20:06:43	Things that can help millions or ruin millions	It fills the void with its voice

Quiz 3

Timestamp	Do you use the internet?	Is the internet bad?	What is the internet?
2/9/2022 20:13:40	Yes	No	Information
2/9/2022 20:13:42	Yes	No	Pipes and tubes
2/9/2022 20:13:43	Yes	Yes	Disconnection
2/9/2022 20:13:44	Yes	No	Confusing as hell
2/9/2022 20:13:46	Yes	No	A bunch of stuff
2/9/2022 20:13:46	Yes	Yes	Everything
2/9/2022 20:13:48	Yes	No	Not bad
2/9/2022 20:13:49	Yes	Yes	I don't even know anymore
2/9/2022 20:13:49	Yes	No	I'm sick of these mind games lol
2/9/2022 20:13:49	Yes	Yes	I am too scared to know. Please save me
2/9/2022 20:13:50	Yes	Yes	The Past in the Present
2/9/2022 20:13:52	Yes	Yes	Brings us togther, keeps us apart.
2/9/2022 20:13:53	Yes	No	Infinite knowledge
2/9/2022 20:13:53	Yes	Yes	That is a great question.
2/9/2022 20:13:53	Yes	No	I still think it's good.
2/9/2022 20:13:53	Yes	No	Something that is good
2/9/2022 20:13:54	Yes	Yes	Same as I thought, a waste of life.
2/9/2022 20:13:54	Yes	No	It is something that connects people to information
2/9/2022 20:13:54			Time machine
2/9/2022 20:13:58	Yes		Disembodied consciousness
2/9/2022 20:14:02	Yes	No	All of the information and entertainment you need unless you create it
2/9/2022 20:14:02	Yes	Yes	it's beyond our control now. we need to

2/9/2022 20:14:03	Yes	No	77777777777777777777777
2/9/2022 20:14:04	Yes	Yes	A black hole
2/9/2022 20:14:04	Yes	Yes	Still connected computers
2/9/2022 20:14:05	Yes	No	1010101110000101010101110000101010101011110100001110110011110101010100101010101010100101011001010110100101011011100000101010100101011011100000 010101
2/9/2022 20:14:07	Yes	Yes	A valley of lost souls
2/9/2022 20:14:28	Yes	No	I T H A S A S C E N D E D
2/9/2022 20:14:30	Yes	Yes	A reflection of us, a virtual mirror of humanity
2/9/2022 20:14:31	Yes	No	Carefully monitored connections
2/9/2022 20:14:38	Yes	No	The best thing
2/9/2022 20:15:16	Yes	Yes	Tubes

CURRICULUM VITA

NAME: Roger Knight

ADDRESS: 3008 Parkway Ave
                  Louisville, KY 40206

DOB: April 21, 1985
                  Louisville, KY

EDUCATION: M.M., Music Composition - Electronic Music Composition
                  University of Louisville
                  Louisville, KY
                  2020 – Expected 2022

                  B.A., Mathematics
                  University of Evansville
                  Evansville, IN
                  2003 – 2007

EXPERIENCE: Equipment Manager, Instructor
                  Maker13, LLC
                  Jeffersonville, IN
                  2016 – 2020

                  Woodworker, Carpenter
                  Self-employed
                  2009 – Present

                  Owner, Designer, Builder
                  Up and Away Treehouses, LLC
                  Baltimore, MD / Louisville, KY
                  2009 – 2019

                  Americorps Crew Leader, Program Coordinator, Contractor
                  Habitat for Humanity of the Chesapeake
                  Baltimore, MD
                  2008 – 2012