

CART 263- Final Project Proposal

Submitted By: Alana DeVito

Date: Tuesday, March 30, 2020

Title of Project:

The Any Day – A Weather-Based Generative Music Machine

Artistic Vision

The Any Day Music Machine is a web-based application where users can log in and play a generative sound piece that is algorithmically customized by their location and their real-time weather forecast. The user may also choose whether they would like to play the scape connected to their location or if they would like to try a location with a different weather forecast and therefore a differently composed soundscape. For example, if a user was in a location with rainy weather, but their preferred a location with sunny weather, they could pick a different location to listen to.

A relaxing visualization of ripples that reflect the different frequencies across the composed music accompanies each composition – which they can project onto a wall to give the experience a fully immersive feel. Since the soundscape will always be in the ambient instrumental genre and leaning towards the more minimalist side, the music machine may be the perfect accompaniment for a casual dinner party, or just as your own relaxing soundtrack to a quiet night at home, with the projection acting as a mood lighting fixture. The music and concept are inspired by both Brian Eno's music and his *Bloom* application and Grimes & Endel's *AI Lullaby*. New additions to this type of project include:

- A multidimensional visual aspect intended to be used as an accompanying projection
- The generative scape will not only be a synthesized music piece, but users will have the option to incorporate field recordings of weather and nature into the composition, introducing a new dimension to the audiovisual experience.
- The seven weather elements (temperature, air pressure, humidity, precipitation, cloud formations, wind speed, and sun) from that user's location will choose the natural field recording layers (nature or weather sounds), key, rhythm, scale mode and FM carrier wave and FM modulating wave (creating the synthesiser sounds and texture quality) that will construct the sound piece for the listener.

The soundscape will be composed from the 8 different 'church' or Gregorian modes (the combination of 4 authentic modes and 4 plagal modes) due to their expressive and emotive nature, which will help in embodying the different weather characteristics through sound. The following chart lists the different modes (from brightest to darkest), the harmonic characteristics (semi-tone distribution) the emotions or feelings that they inspire and possible weather partnerships:

(citation: <https://blog.flat.io/choosing-the-key/>)

Mode Name	Semi-Tone Distribution	Characteristics	Weather Connections
Lydian	2-2-2-1-2-2-1	Brightest, uplifting and quirky.	Extremely sunny and clear sky, slight breeze.

Ionian	2-2-1-2-2-2-1	Happy, positive.	Perfect weather, balmy and sunny.
Mixolydian	2-2-1-2-2-1-2	Happy but serious.	Very hot, or hot and humid.
Dorian	2-2-2-2-2-1-2	Sad but hopeful. "folky"	Cloudy and grey, but some sun
Aeolian	2-1-2-2-1-2-2	Sad	Raining.
Phrygian	1-2-2-2-1-2-2	Dark	Heavy rain, very grey, windy.
Locrian	1-2-2-1-2-2-2	Evil	Storm, lightening, high winds, extreme weather, nighttime

Once the modal scale is identified, the key will also create further depth and customization to the composition. The following chart lists the different keys and their 'emotional personalities' to add even more sonic colour to the atmosphere of the experience:

(Citation: <https://blog.flat.io/choosing-the-key/>)

Key	Emotional Personality
C	Innocence, happiness with spiritual feeling.
Cm	Innocence, sadness, heartbroken, evokes yearning.
C#m	Passionate and deep. It evokes sorrow, grief, despair, wailing, self-pity.
Db	Depressive masked by an air of happiness. Evokes grief and despair.
D	Triumphant and victorious. War marches and holiday songs.
Dm	Serious and melancholic. Concern, contemplation.
D#m	Deep and anxious. Distress, terror, darkness, hesitation.
Eb	Evokes cruelty but also devoted love, openness and intimacy.
E	Ready-to-fight, dissatisfaction, but also joy and delight.
F	Optimism, exploding energy.
Fm	Dark, funeral, depression, death, loss, misery.
F#	Conquest, relief, triumph, victory and clarity.
F#m	Resentment, discontentment, lamentation, but with a little hope.
G	Happy, serious, idyllic, poetic, calm, satisfaction, tenderness, gratitude and peace.
Gm	Discontent, uneasiness, failure, concern and struggling.

Ab	Death, eternity, judgement, darkness.
Abm	Related to wailing, suffocation, lamentation, struggle and negativity.
A	Joy, reciprocated love, satisfaction, optimism, trust, spirituality.
Am	Sad, but tender.
Bb	Joyful and cheerful. Love, consciousness, hope, optimism and peace.
Bbm	Nighttime, darkness, blasphemy, death and destiny.
B	Strength, wildness, passion, jealousy, fury, negativity and the will to fight.
Bm	Solitude, melancholy, patience, calm, submission and acceptance.

The library that will assist in fetching the notes and keys of each of these modes for composition is Tonal.js. For more weather nuances to deepen the uniqueness of each composition and add texture, Tone.js will be used to generate the synthesized sounds in the composition – the built-in effects and modularity (referring to the patching/connecting logic being similar to classic modular synthesis technique) will provide many opportunities to specially design the sound and composition for the user's customized experience. JQUERY UI will be used for the instruction modals and further customization menu within the application. An addition that is not currently in the prototype is the implementation of the ClimaCell API for the weather API needed to fetch the weather data. After some research, this particular API seems like the most appropriate for this project as it was listed as the best for hourly, real-time and short-term forecasts. The API, located at <https://rapidapi.com/ClimaCell/api/climacell>, requires the latitude and longitude of the user's location, so the same p5 geolocation library (p5.geolocation.js) I utilized for my Labyrinth Go Game for project one will be integrated into this program to make that possible.

From the current 2-part prototype, there are a number of technical goals and challenges that I need to engage with to accomplish the project. The first, overarching challenge is integrating the two parts of the prototype, as they use different types of code organization and libraries. Part 1 (in the folder titled, "mainUO") began from a tutorial called, "The Generative Musicbox Project" (https://www.youtube.com/watch?v=PTzUXTE_GtA&ab_channel=WickedlySmart and <http://bit.ly/wickedlysmart> - although many edits and changes were made along the way) and Part 2 (in the folder titled, "synth") was built off and expanded from the tutorials, "Generative Processes: Week 6, Part 1 - TONE.JS, DRONES" and "Generative Processes: Week 6, Part 2" (https://www.youtube.com/watch?v=ddVrGY1dveY&ab_channel=DavidBouchard and https://www.youtube.com/watch?v=CkjM8et49II&ab_channel=DavidBouchard). I changed many elements of the visualization from the original "Generative Musicbox" tutorial so that the tones would be accompanied by color changing ripples that built up on the canvas over time rather than just bubbles that disappear – the ripples also move very slightly at each iteration and with different opacities, giving each ripple a more glowing ring-like look. Other important code that will be repurposed from this part of the prototype is the way in which the tones are timed to re-trigger, with my current code improvements where the tones that are re-triggered move through the scale linearly in a more melodic and generative fashion rather than just retiggering the same note (which was the initial design from the tutorial). I would like to keep the general organization and

modularity of classes from the current state of prototype Part 1 as well as the JQUERY UI elements that are being tested in said prototype, however, I would like to update the code to a more modern OOP style, rather than the currently employed ‘prototype’ syntax. I will also replace the samples that are being used in prototype Part 1 (folder: ‘mainUI’) with the code employing the Tone.js and Tonal.js libraries in the prototype Part 2 (folder: ‘synth’) for authentic computer based digital synthesis techniques. I will also take the drawing of the waveform from prototype Part 2 and adjust the code so that it creates frequency modulated ripples in dotted lines instead of just the replicated waveform like an oscilloscope (which is its current prototype state), and layer it over the glowing-ring-ripples.

Another item that I may either expand on or remove is the CSS constructed ripples (loosely inspired by this tutorial:

https://www.youtube.com/watch?v=FCPNN730c98&ab_channel=DarkCode

that are draggable visualizations in the prototype Part 1 example. The ripples are currently movable for visual aesthetics and customization but may be used as delay effect triggers that the user can ‘attach’ to tone ripples. The delay effect would be added to that track of notes that are generated by that JS scripted ringed ripple object (see annotated images below).

Lastly, I will need to plan out - through the use of pseudo code before the actual coding – the logic needed to organize the weather element data and then assign the different ranges of data sets to specific parameters for the sound scape. I will start from the music characteristics charts above and then assign different weather data sets to each mode, key and Fm Synth ‘web structure’ (the different arrangement of waveforms effecting a string of other waveforms to create different sounding synth instruments and textures – the carrier signal and the modulating signal) – see image 1 below. The different waveforms – sine, triangle, square and sawtooth – will be assigned to weather parameters and how they combine in FM synthesis will determine the synth tone’s sound. Another bit of code that needs special attention is the conditionals handling the playing order of the notes from the different modes. This is of particular importance because modal music is very dependent on the direction of the progression of notes through the scale. If the home note of the mode is not played or resolved to regularly then the piece will just sound like a regular scale. Fairly detailed conditional statements will need to handle this important note playback aspect.

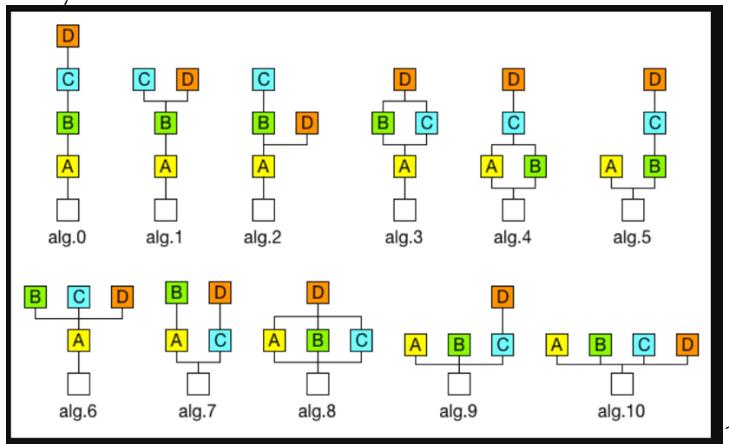
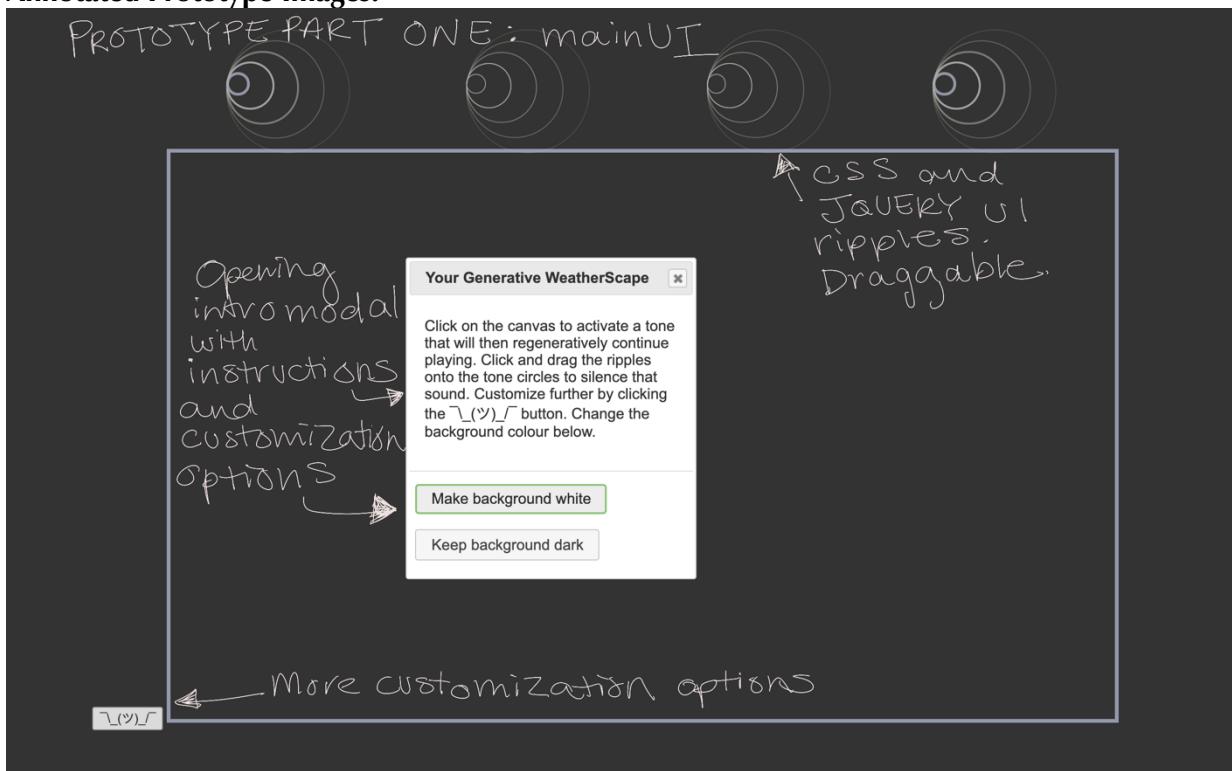
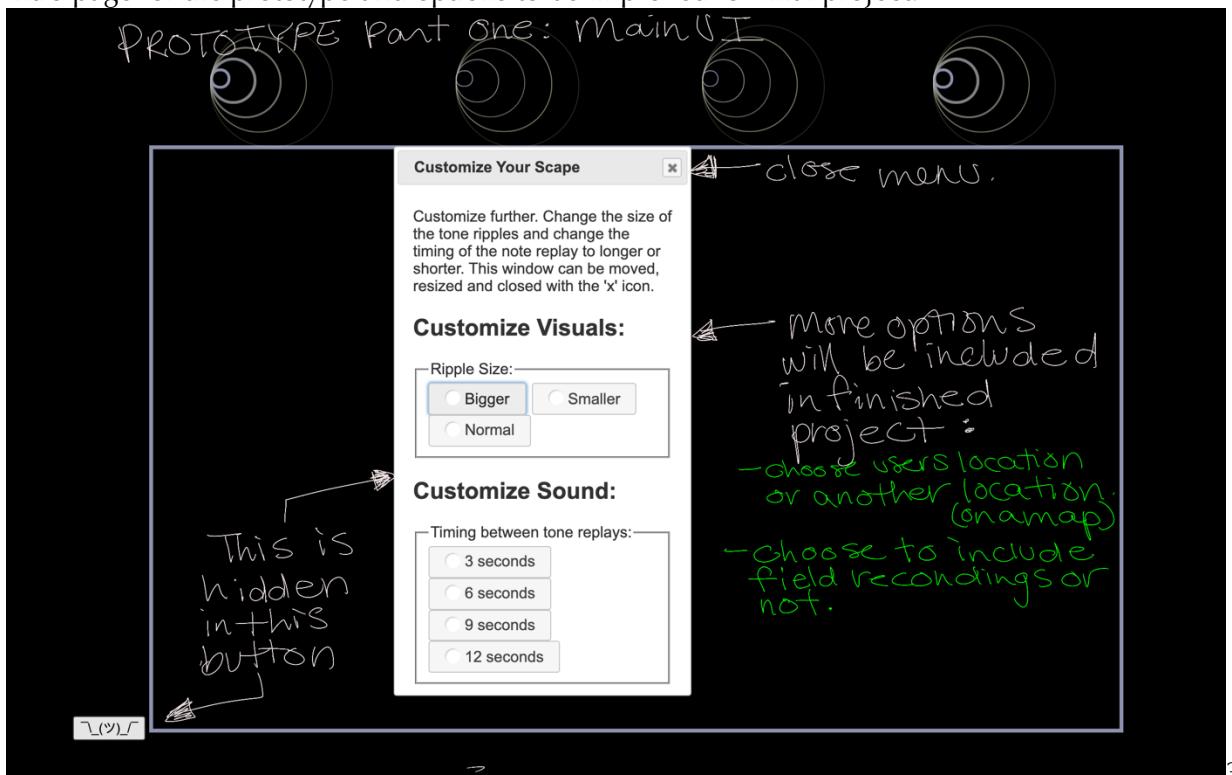


Diagram of different arrangements of carrier and modulating wave structures in FM synthesis.

Annotated Prototype Images:

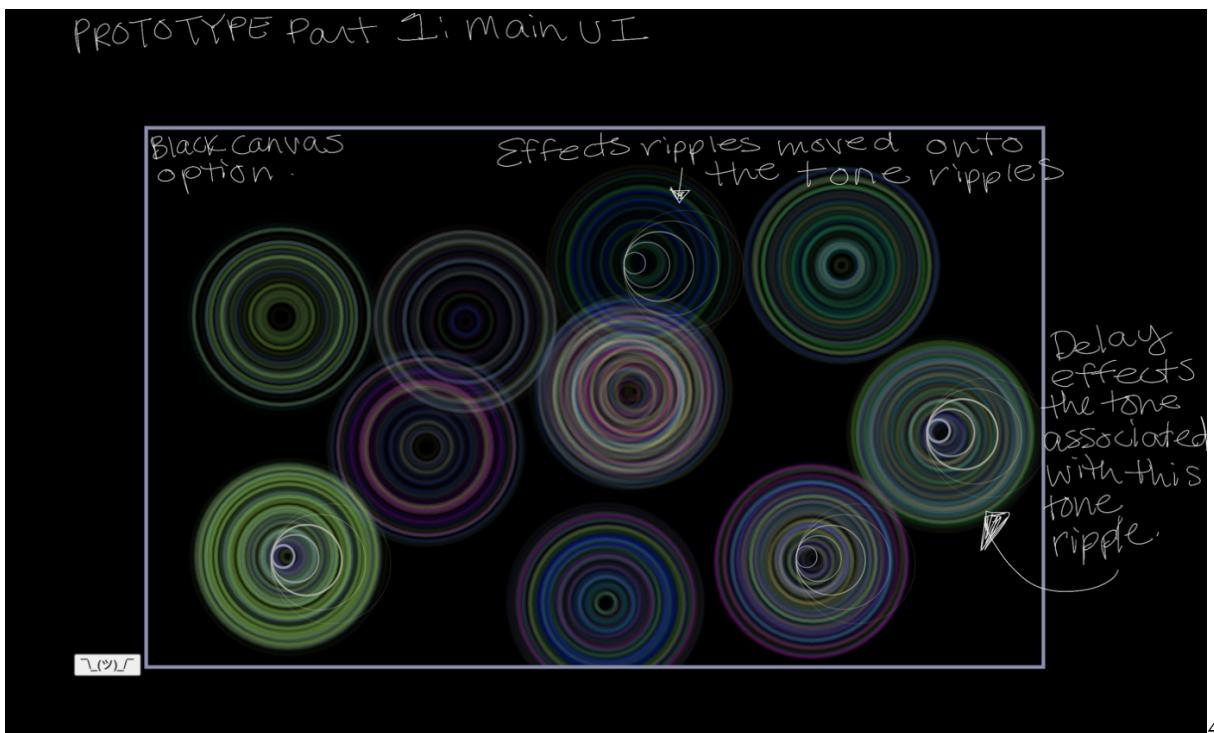


Intro page for the prototype and options to be improved for final project.



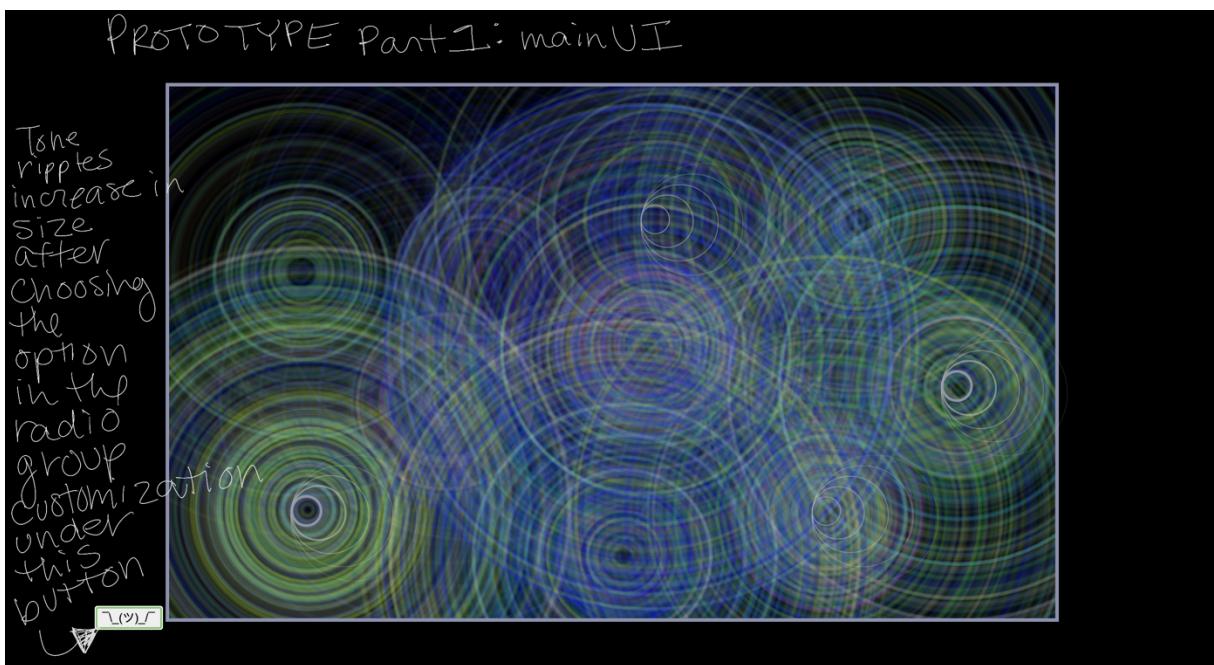
The hidden customization modal under the button and additions to be made for final.

PROTOTYPE Part 1: Main UI



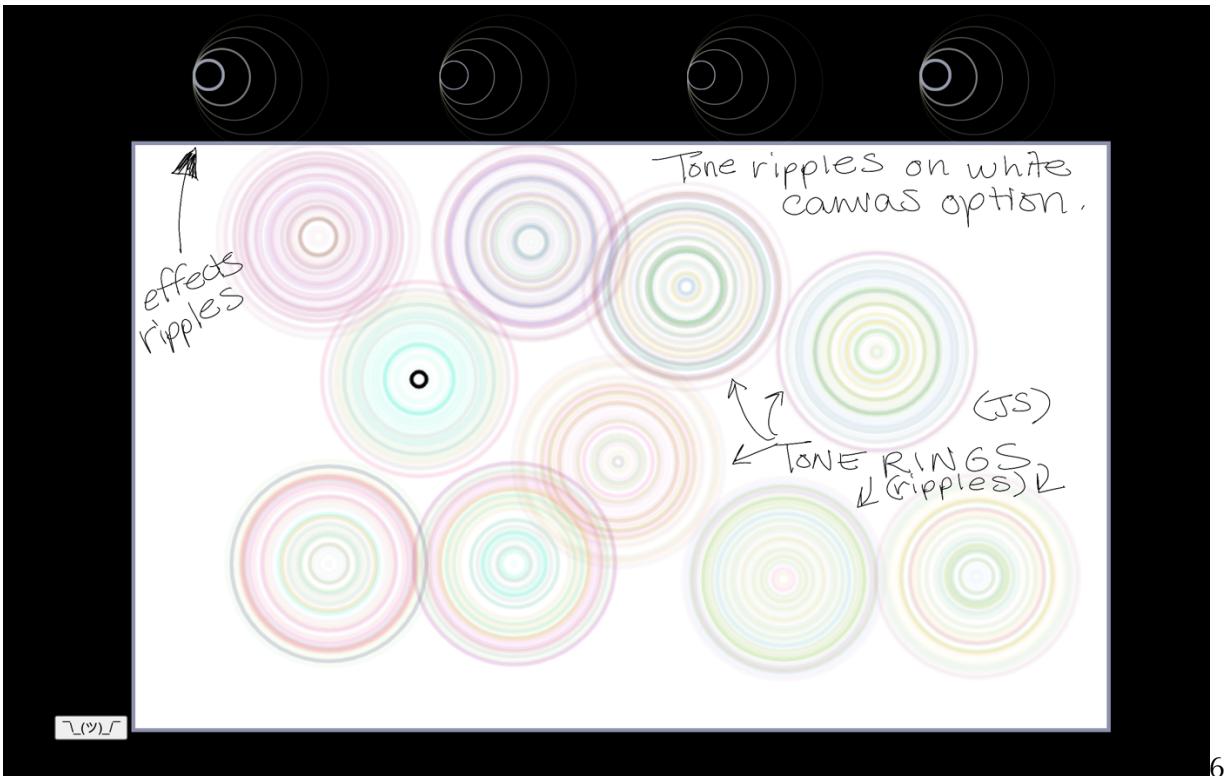
4

Showing how the CSS ripples will be moved onto the JS tone ripples to trigger a delay effect on that tone (not an active element in prototype, will be added to final project).



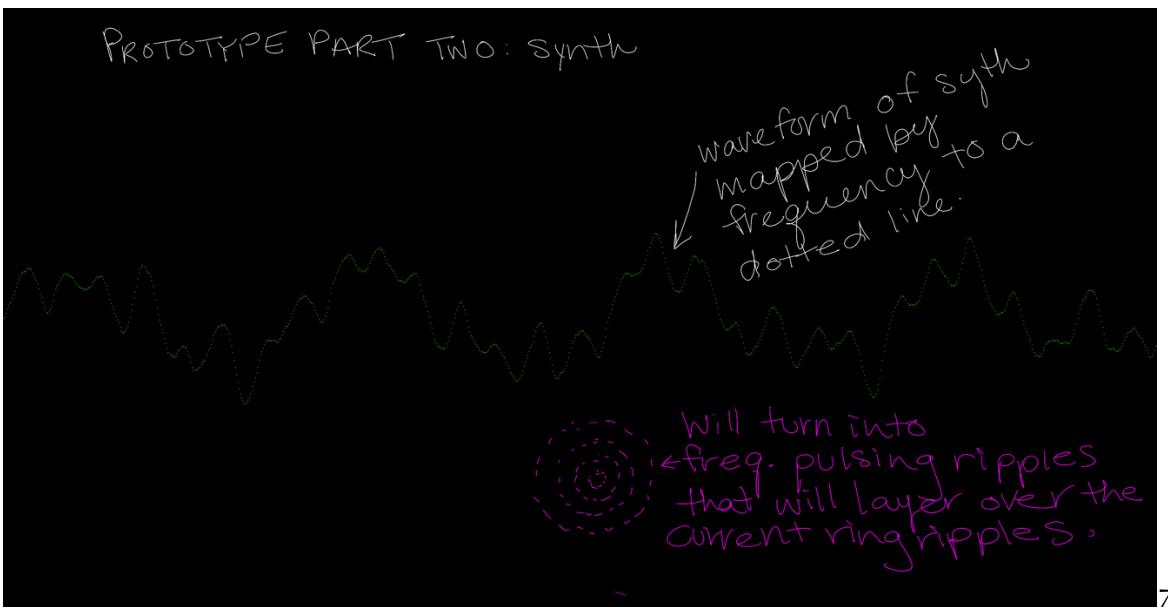
5

The customization of the ripple sizes after activation.



6

Example of the white canvas option in action. May not continue to the final project.



The current state of prototype part 2, with the waveform mapped from the frequency values. For the final, this will be layered as modulating ripples over the glowing ring ripples, or as a vertical curtain of multiple waves.

Citations:

- A generative music box project. (2016, July 12). Retrieved March 29, 2021, from
https://www.youtube.com/watch?v=PTzUXTE_GtA&ab_channel=WickedlySmart+and
- Bouchard, David. (2020, October 20). Week 6 - Part 1 - TONE.JS, DRONES. Retrieved March 29, 2021, from https://www.youtube.com/watch?v=ddVrGY1dveY&ab_channel=DavidBouchard
- Bouchard, David. (2020, October 20). Week 6 - Part 2 - TONE.JS, Synths. Retrieved March 29, 2021, from https://www.youtube.com/watch?v=CkjM8et49ll&ab_channel=DavidBouchard
- Ripple animation using only html & css. (2018, December 29). Retrieved March 29, 2021, from
https://www.youtube.com/watch?v=FCPNN730c98&ab_channel=DarkCode
- Valverde, R. (2020, October 27). How to project emotions through music: Choosing the key. Retrieved March 29, 2021, from <https://blog.flat.io/choosing-the-key/>