tchime tree
an interactive sound installation

by

Gabriel Samuel Greenberg
Class of 2014
Table of Contents

<table>
<thead>
<tr>
<th>TOPIC</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduction: The Sound Installation</td>
<td>3-6</td>
</tr>
<tr>
<td>tchime tree: The Site</td>
<td>7-11</td>
</tr>
<tr>
<td>The Politics of Space</td>
<td>12-14</td>
</tr>
<tr>
<td>Interfaces: Modes of Interactivity</td>
<td>15-28</td>
</tr>
<tr>
<td>Physical Layout</td>
<td>29-31</td>
</tr>
<tr>
<td>Conclusion</td>
<td>32-34</td>
</tr>
<tr>
<td>Acknowledgements</td>
<td>35-37</td>
</tr>
<tr>
<td>Bibliography</td>
<td>38</td>
</tr>
<tr>
<td>Appendix</td>
<td>39-52</td>
</tr>
<tr>
<td>Pictures of the Installation</td>
<td>39-41</td>
</tr>
<tr>
<td>The Opening</td>
<td>39-40</td>
</tr>
<tr>
<td>One Week Later</td>
<td>41</td>
</tr>
<tr>
<td>Code(^1)</td>
<td>42-51</td>
</tr>
<tr>
<td>Processing</td>
<td>42-43</td>
</tr>
<tr>
<td>Yahoo Weather Query</td>
<td></td>
</tr>
<tr>
<td>Interface.js</td>
<td>44-47</td>
</tr>
<tr>
<td>Evolver</td>
<td></td>
</tr>
<tr>
<td>SuperCollider</td>
<td>48-52</td>
</tr>
<tr>
<td>Interfaces</td>
<td>48-50</td>
</tr>
<tr>
<td>Weather</td>
<td>41-52</td>
</tr>
</tbody>
</table>

\(^1\) For the sake of space, I only included a fraction code used for the installation in the appendix of this paper. The code that is present should give the reader a picture of the scheme I set in place that allowed for the different level of interaction between Processing (YahooWeather.com for wind data), Interface.js and Interface Server.js (for the widgets that were hosted on the user’s devices), and SuperCollider (for interpreting interactions in the form of OSC messages and creating sound synthesis accordingly). For a complete version of the code, please refer to the accompanying CD-ROM.
**Introduction: The Sound Installation**

At its core, a sound installation’s position within the nexus of musical forms extends beyond the normal confines of a concert hall performance. Its sonic imprint on the space it inhabits is temporally stretched, and its existence should be considered permanent, regardless of whether or not logistical realities will allow this to be true. As such, a sound installation’s relationship to the space in which it sounds is of fundamental concern to the artist who creates it. Beyond practical concerns like power sources and viable speaker locations, questions about how the site functions normally (without an installation present) are deciding factors in the creation of a sound work that will fundamentally alter that functionality. What are the environmental sounds that already exist at the location? What is the reverberant nature of the space? What kind of people might find themselves there and how will the work’s presence affect the way in which those people might typically operate? All of these and more are relevant questions when deciding the form and content of a sound work that is, by its very nature, site specific. While it is true that other musical forms function on these levels—given that sound is inexorably tied to the particular space in which it is heard—for a sound installation, if only due to its prolonged nature, these issues are particularly relevant.

For one, the absence of a culturally normative focal point is highly consequential. At a typical concert, there is a stage, performers, possibly a stereo...
pair of speakers, and seating (often of fixed orientation and raked), all drawing the listener’s attention to a designated field of focus. The dichotomy of listener/performer that is engendered by this setup doesn’t necessarily exist at a sound installation where the artist may not even be present and the sound image is quite possibly diffused over a number of loudspeakers. Moreover, the audience members are usually free to move about the space and come and go as they please, annihilating the standard etiquette of “sit down, pay attention, and applaud when it’s over.”

Max Neuhaus, a sound artist known for being the first to use the term “sound installation,” began his career as a virtuoso percussionist. He ultimately stopped playing “music” altogether because he thought that, as a category, it was too limited, and that people were locked into the artificial “concert hall syndrome.” A noted founder of the genre later called “sound works,” he was of the opinion that installations of the sort should not even be considered “art.” In reference to his seminal work *Times Square*:

The work was unsigned. Anonymity is essential...when the Rockefeller Foundation wanted to advertise it’s funding with a “Times Square” plaque, Neuhaus said no. He was not making music, he required no applause. His sounds are “the opposite of music. They do not develop in time as music does.” He is “an artist working in a serious way” for a “public that is not necessarily looking for “culture” then and there. 

---

2 Ericson, NY Times, 1977
3 Zwerin, International Herald Tribune, 1999
Neuhaus was interested in the idea of "plausible" sound or sounds that might have been there anyway. Neuhaus calls it "a sound to carry you to a new place, to transport you to another frame of mind, just different enough to throw you. Perhaps you were not even aware of it."\(^4\) This particular installation had no visual component, that is to say it was built into the slits of a walkway with all of the mechanisms creating the sounds hidden underground. The result was that the sounds seemed to come from all around, the sides and above, even when the listener was standing right on top of the source.\(^5\) This was not merely a consequence of the logistics of the installation, but rather a conscious compositional choice made by the author. *Times Square* was “plausible sound” because, in Neuhaus’ own words, it was “designed to fit in with the sounds around it and yet be different from them.” It was “meant to provoke the feeling that it was discovered by the audience, who would then pass on the experience to others.”\(^6\) When reflecting upon a chance encounter in which a regular visitor to his installation witnessed him servicing his work, Neuhaus mused:

"Think of the thousands of people who crossed that island. That woman was startled to find an explanation all of a sudden. She’d come to think of the sound as her own. And it was."\(^7\)

It is this separation of the artist from their work’s result that is central to the notion that sound installations might be the strongest candidate for a more
egalitarian art form. The concert-hall syndrome is absent from a public, unmarked installation that features bleed over from the environment it inhabits. Rather than punishing inattentive listening, as an usher might do to a rowdy opera-goer, *Times Square* asks nothing of its audience, but rather rewards individual listeners who choose to pay attention. As Ericson put it in a contemporary NY Times review:

“It does not pollute the air nor force itself on one, the way Muzak does in restaurants and other public places, but rather steals into one’s consciousness, and by its uniqueness it makes one listen to the sounds—chiefly traffic noise—with a greater appreciation of their special textures.”

That is to say, sound installations should do more than just add atop the sonic environments in which they operate. They should also create the opportunity for the sounds that were already there to be given more attention and heard in a new way. Thus a relationship is created between composer, listener, and the environment at large in which each party takes an equally active role in the holistic perception of the work.

---

Ericson, NY Times, 1977
tchime tree: the site

My thesis installation, entitled tchime tree, was an attempt to create an environment that both reflected the naturally occurring sonic qualities of its location and presented the opportunity for individual and social interaction therein. The initial inspiration for tchime tree was The Singing Oak, a sound installation by artist Jim Hart that I stumbled upon in City Park, New Orleans. In the middle of the park, surrounded by green lawns and a glistening pond sits an ancient oak tree, adorned with an array of wind chimes hanging from its massive limbs, the largest of which is 14-feet long. The chimes were tuned to a pentatonic scale so that the chance melodic structures would be suggestive of the traditional songs of the area. According to the artist statement, “the music of The Singing Oak emerges from the interaction between natural forces and the human imagination. It demonstrates the harmony between nature and man in a way that is soothing and melodic to the human ear.”9 I was taken with the atmosphere that this installation created. People spending time under the tree were calm and relaxed, many sitting back with their eyes closed, as if they were bathing themselves in the rich textures of the chimes. In a city known for its boisterousness, people under The Singing Oak were noticeably quieter and seemed more meditative and introspective than even 200 yards outside of its sounding radius. While individuals and groups remained enclosed in their own

---

9 Hart, 2008
social spheres, the sense that a de-facto community existed between those of us under the oak was palpable.

For my installation, I wanted to create an environment that inspired a similar level of mindfulness, attention, and intention as did Hart’s *Singing Oak*, so I planned to place nine speakers in a cluster of trees at the corner of the Center for the Arts complex at Wesleyan University in Middletown, CT. In translating the sounding bodies from physical chimes to digital audio played through loudspeakers, it was necessary to develop a way to recreate the occurrent state of the physical space within the digital space of the computer program. In other words, I wanted the characteristics of the sound material playing out of the speakers at any given time to accurately portray the reality of the physical environment in which they were sounding. Because the speakers were in effect taking the place of wind chimes, which would normally only sound when the wind blew on them, it was necessary to update the rhythms of the digital chimes in real time. To accomplish this, I used Processing to query Yahoo Weather’s online website for the occurrent wind speeds in Middletown and live-updated the sound material accordingly. The effect was a relatively accurate representation of what physical chimes would sound like in that space, i.e. slow wind speeds led to more sparse tones and visa-versa.

The sound material in its most basic form consisted of a synthesized interpretation of chimes that I designed using the programming language
SuperCollider\textsuperscript{10} and, for the purposes of this paper, will refer to from now on as ‘tchimes’. Upon choosing the site, it became clear to me that the natural reverberance of the limestone walls that boxed in the grove on all sides rendered any need for digital reverb obsolete. The tchimes sounded at random rates constrained by established boundaries scaled by the speed of the wind. Noting Hart’s stated goal that listeners of \textit{The Singing Oak} would infer their own melodies from the randomly generated patterns if the tonal variances were relatively simple, I too wanted to create something where the listeners had the ability and agency to string together individual sonic events into cohesive structures as they imagined them. In an effort to accomplish this, each of the nine speakers was associated with a specific scale degree along two octaves of a pentatonic scale that started at a G#4 (415.3 Hz). The quality of the scale wavered back and forth from major to minor at rates that were also scaled by the wind speed data, with all the tchimes switching modes in unison. The rates at which the tchimes sounded were also scaled relative to their position within the scale, with the lower notes sounding slightly less often than the higher. This decision was based on the physical reality of wind chimes, that being larger, heavier bodies would take more energy to sound than smaller, lighter ones. To further harken back to the installation’s roots in physical chimes, each loudspeaker retained its respective place within the scale for the duration of the piece, despite any changes in timbre or rate it might undergo. This was meant to establish a sense of stability within a sound environment that had the potential

\textsuperscript{10} See appendix for complete code
to become overwhelming as much of the other sound material either panned or jumped around the nine channels.

While planning the installation, I looked for a site that was not entirely out of the way of normal foot traffic, but tucked away enough so as to demarcate the space apart from the normal college campus atmosphere. The site also needed to exist within a network of environmental sounds that would allow my installation a certain level of autonomy while retaining the preexisting characteristics of its natural soundscape. The site I chose, while being set away in an enclosed grove, was also set apart from the traffic of Middletown’s route 66 only by a short wall and a small stretch of lawn. One can hear the sounds of passing cars, the reoccurring beeping of a crossing signal, the occasional wail of a siren from a nearby fire department, as well as the birds and miscellaneous chatter typical of a New England university campus. In an effort to satisfy Neuhaus’ notion of “plausible sound,” the sound material contained in tchime tree—other than its distinguishing component (the tchimes themselves)—was designed to be harmonious with the sounds that already populated the space. Contrary to Times Square however, it was not necessary for me to have the installation completely out of sight; rather I wanted its presence to exist somewhere in the space between public and private. My intension was that, should a passerby be intrigued by the sounds they heard, they could easily go investigate, while uninterested parties would not be forced to unwillingly
participate. The intention of the potential listener was a deciding factor when determining both the location and content of the installation.
The Politics of Space

By its very nature, an outdoor installation is an intrusion upon public space, and the space onto which it imposes, as well as the people who typically or potentially inhabit that space, must be carefully considered. In his essay “The Nonaggressive Music Deterrent,” Jonathan Sterne analyses the implications of private businesses that play programmed music, or Muzak, in the public air space outside of their establishments in an effort to create “environmental feels” scientifically engineered to at once placate potential customers and stave off individuals deemed unwanted.\(^\text{11}\) He asks, “Who has the right to design the acoustic dimensions of the outdoor urban and suburban spaces? By what means?” While his essay does not give a clear answer to this question, he does show that the “nonaggressive music deterrent plays against a law of averages...teens, drug dealers, the homeless, sex workers and low-income nonwhite populations are all lumped together as targets of the new Muzak.”\(^\text{12}\) He concludes:

“the nonaggressive music deterrent is designed to discourage people from perceiving outdoor environment in terms of shared, multiple meanings and uses. Used outdoors...[it] is an attempt to code space, and specifically to code it in terms of social class, race and age.”\(^\text{13}\)

\(^{11}\) Sterne, The Nonaggressive Music Deterrent, pg. 9
\(^{12}\) Sterne, The Nonaggressive Music Deterrent, pg. 15
\(^{13}\) Sterne, The Nonaggressive Music Deterrent, pg. 15
Moreover, Sterne’s argument suggests that Muzak is particularly effective in coding spaces in this way because it rewards inattentive listening. That is to say, if you are “meant” to be in a certain store, the Muzak playing in there is likely to be inoffensive to you and you are less likely to pay it any attention.

With this in mind, I set out to center my installation as far away from the goals of the nonaggressive music deterrent as possible. I wanted it to be in a public space and therefore had to accept the possibility that some passersby might not like it, but I could try to make it as inoffensive as possible. When choosing the installation site, this was a major deciding factor. I didn’t want to intrude too much upon anyone’s space, but I wanted to do the opposite of the nonaggressive music deterrent in that I wanted the space to encourage any passersby, regardless of their background, to engage with the piece in a meaningful way. This notion reinforced my decision to use the pentatonic scale for the basic tchime material and to established the tonal center of all the material at the same pitch (415.3 Hz). This was my way of ensuring maximal sonic cohesiveness, even in moments of upmost density. My intention was that, no matter what ended up being played, if all the material was rooted at the same pitch and developed within the same pitch relations, moments of extreme dissonance would be avoided and thus the sonic material would remain minimally offensive.
Still, simply being minimally offensive is not sufficient if the goal is to use sound to create a space that fosters mindful listening and group interaction. In questioning what a truly egalitarian form of public music would look like, I came to the conclusion that individuals who had stake in the sonic character of a local environment should be rewarded for caring. I had to go further than just crafting the sounds to be easily digestible, interested parties needed to have the ability to assert their sonic identity within the sound environment.
Interfaces: Modes of Interactivity

By default, the basic tchime structure was designed to remain sparse and slowly move through timbral changes over the course of an hour, resetting itself at the top of every hour with the time signified by a tchime rendition of Westminster Quarters.\footnote{Note: I was not able to ever get this aspect of the installation fully operational and as such, opted to omit it from the piece this iteration.} It should be noted that each step along the progression through the structured set of changes was punctuated by a new generative music function. This meant that while the order and overall method of each section was the same every time, the way in which this new layer of processing manifested itself was not fixed. While this general pattern was the most composed component of the installation, I still wanted an interested user to have the ability to “fast-forward,” so to speak, should they become bored. This led me to develop an interface server that turned the computer running the program into the host of a local website. Users could then enter a URL into the browsers of their smart phones, mobile devices, or laptop computers to reach a website that contained a number of widgets that directly interfaced with the installation. Now, should they find themselves bored with the tchime pattern offered, they could press a button that moved things along (see figure 1). Thus it became possible to make interaction with the sound environment even more overt. Not only would listeners have an increased ability to subjectively perceive what they
heard due to the relative simplicity of the tonal vocabulary, they would also be able to directly interact with the occurring sound material being played.

Figure-1

The “evolver” widget allowed users to initiate the basic tchime pattern should it not already be playing (button, top left), and move to the next location within the pattern (arrow shaped button, middle)

Before going into all the modes of interaction offered to users of tchime tree, I must note that an important aspect of the way the entire installation functioned was that it was not necessary that anyone interact with the installation for it to sound. By default, the tchime pattern would run and continue to articulate itself relative to chance operations within the code and the environmental information. If however, someone cared enough about how it sounded, they could interact with the installation and alter the sonic space they shared.
The widgets I used were made possible by Charlie Robert’s free-source code Interface.js and the Interface.js Server application. Interface.js is a device agnostic GUI library, meaning the code for each widget has to be written only once and the developer can be “reasonably assured” that it will work on all kinds of smartphones, tablets, and laptops. This was particularly convenient given my desire to provide an equal opportunity for any given user to interact with the installation in the same way. For my purposes, being able to have a library of GUIs (sliders, buttons, etc.) made it possible to focus more on the desired modes of interaction and less on the logistics of hard coding interface components. Once a widget is coded and the file directory is loaded onto the Interface Server application, it facilitates the transfer of OSC or MIDI messages from the JavaScript code running on the user’s browser back to the Server, which then can be programmed to send incoming messages to any other applications running on the same computer. Interface Server also indexes users client numbers, which can be appended to outgoing OSC messages. This made it possible for me to read and send user specific data to and from SuperCollider. However, it is important to note that, given the code is still in its developing stages, I was not able to send user specific information back to the Interface Server at the time of the opening. This resulted in a number of interesting bugs and constraints that shaped the way in which piece ultimately functioned, both sonically and socially, but I will go into further depth about this in my conclusion. Regardless of the bugs in the

---

15 https://github.com/charlieroberts/interface.js
The first, and perhaps most important widget I designed controlled a simple sine tone with an envelope. The user had the ability to alter the pitch and rate of the tone, as well as the decay curve of the envelope, using a slider, buttons, and another slider respectively (see figure-2). This interface was particularly important because it was the only opportunity for a sound to exist that was outside of the tonal register of the rest of the sound material. What this meant was that users who selected this widget could easily and distinctively assert their own sonic identity within the larger sound environment, and thus determine their own degree of influence on the overall sound situation. Upon opening the widget, a user specific module was created on the host computer that was fully controllable by the user. That is to say, should another user open up the same widget at the same time, a new tone would sound that could only be controlled by that specific user. The intention here was that users would be given voice and the ability to communicate with each other, as well as the sound environment at large. To prevent the sound environment from being polluted by too many of these tones, I wrote a bit of code that checked the last time the widget was altered by the user against a clock. If two minutes ever went by

---

16 The limitations that exist as a result of using personal devices as the primary mode of interaction with the installation and the resulting social implications are not beyond me. They are, however, outside of the focus of this section and so I will touch on them in the conclusion.
without the user manipulating the widget, the user was assumed to be finished and the tone was terminated. I repeated this schema for many of my widgets to both protect against gratuitous density and to reinforce the overarching theme that pervaded the interactive element of the installation. That being, only when the user proves to the program that they are interested in their sonic imprints and have the desire to engage mindfully in the creation of a shared environment are their actions counted.

Figure-2

The “simple-tone” widget allowed users to initiate a sine tone (button, top right), control its pitch (slider, bottom left, or text box, middle middle), articulate rhythm (pre-set values using the button, top left, or user-defined values using text box, top middle), timbre (slider, bottom middle), and volume (slider, right)

Another widget of great sonic and theoretical importance to tchime tree allowed users to initiate pre-recorded audio files that were then filtered through what I will refer to as the “sample tchime” mechanism (see figure-3). Sample tchimes are audio samples that are constantly undergoing changes to their
location within the multi-channel distribution, the speed of playback, and the length of the sample itself. Take, for example, an audio recording of bird song that is 20 seconds long. Once initialized, the file might be played back starting from second 10, getting retriggered every half second, or it might start at the beginning and get retriggered after 10 seconds. It could be played back at the speed at which it was recorded, providing a clear acoustic image of a bird, or half speed, resulting in a stretched, low, resonant version, rendering it potentially unrecognizable. Moreover, once initiated, these defining characteristics, as well as the sound’s position amongst the speaker array, were constantly shifting at rates scaled by wind speed data. Also, the chunk of the sample being played and the speed at which it is played shift at different times, resulting in even more variety when it comes to the potential combinations of characteristics. The user is also given two options for additional after effects to be put on the tchime samples, both of which add delayed tonal resonances triggered by the attacks of the original tchime sample (see figure-4).

What results is a pre-recorded sound that has varying degrees of reference to its original source. How closely the eventual sound resembled its source is determined completely by chance, and this relationship is constantly in flux while the sample tchime is playing. Considering that any sample played in this way has an equal chance of lying anywhere on the ‘acousmatic’—“based on the abstraction of the recorded “sound-object” from any dependent relationship
to its origins”¹⁷—spectrum, it is fair to say that a user might spend their time focusing on the possible source of the sound rather than the acoustical properties inherent to it. However, I am of the opinion that the unpredictability of the character of a sample tchime at any given moment forces the listener to pay attention to the sonic qualities of the recording rather than to index the sound relative to its source, similarly to how Brian Kane believes Pierre Schaeffer achieves his desired mode of focus with Étude aux allures.

“By overloading the number of sources, Schaeffer denies the auditor any fixed point of orientation available via écouter. As a result, he compositionally forces the auditor to shift attention onto morphological characteristics...”¹⁸

What’s different about sample tchimes and the varied sound sources in Étude aux allures is that the user chooses for themself which recording to play in the former by picking though a list of names. One might assume that because the user knows which tchime sample should be playing after initializing it they would be less likely to attend to the task of deciphering its origins. However, the social aspect of tchime tree provides a further complication to the issue, namely that there exists the potential for multiple users to be playing multiple samples at once. This creates a level of confusion that results in each individual user attempting to trace their contribution within the convoluted sound mass. Thus the potential for reduced listening gives way to an egotistical desire on the part of the user to see the way in which their interactions have manifested. However,

¹⁷ Born, 12
¹⁸ Kane, 121
perhaps this can be considered a form of reduced listening in its self, for if we are to believe Kane’s claim that “acousmatic listening is a shared, intersubjective practice,” and thus is culturally performed, the act of distinguishing one’s sound from another necessarily requires the individual to listen closely and attentively. An inherent element of sound installations is that, because the standards of behavior are unclear, the social dimensions on which it operates are foregrounded. Framed within the goals of the project, the kind of close “inter-subjective” listening this produced was satisfactory for me.

Figure-3

The “tchime samples” widget allowed users to choose from a list of sound files to be processed (drop-down menu, top left), initiate its audibility (button, bottom left), and control its volume (slider, right)

---

19 Kane, 7
While the interactivity provided in the widgets explained above fulfills the need to bestow agency upon the user, they have all been up to now non-kinetic forms of interaction. That is to say, it is one thing to establish a tool by which a consumer of a sound installation can alter the sonic environment they're in, but it is an entirely different thing if that mode of interaction is tactile and kinetic rather than just moving sliders and pressing buttons on a screen. In an essay entitled “Human-Computer Interaction and Music” in the collection “Sound—Perception—Performance,” the authors present their research about the potential merits of such forms of interaction with computers and music. They find that:

The “tchime samples” widget also allowed users to add two after effects to the tchime sample sound file, pluck (button, left middle) and twink (button, right middle).
“Musical interaction allows for a learning-by-action exploration, lowering the barriers of the inherent abstract nature of many of these concepts and making the global experience much more accessible and enjoyable.”

However, when it comes to actually parsing out the technological mechanisms for creating meaningful modes of interaction, they run into problems when the virtual instruments the user might play exist in an “invisible visual field.”

“Several usability drawbacks can be found in this approach...the most glaring problem when the virtual drum is located at a fixed position is that the drum would be “invisible” in the real world... A second approach to the implementation of the air-drum would rely strictly on user’s motion to infer when the user “intends” to hit the drum... A feature that is clearly linked to a downwards drum-hitting gesture is that of velocity and acceleration. More specifically, a negative acceleration peak along the Y axis.”

Based on their research questions and methodology, the crucial aspect of a kinetic interaction with a computer that results in creating sound is the users intention to create such a sound. While the researchers in this article attempt to combat this issue using complex algorithms that try to simulate how a drum hit might physically be represented and use machine memory to try to establish user specific patterns, a simpler approach would be to allow the user to indicate their intention with a preparatory action, similar to a percussionist picking up their sticks before striking a sounding body. In an attempt to accomplish this, I used one of Charlie Robert’s pre-built acceleration GUIs that harnesses pre-

---

20 Bader, 369
21 Bader, 380
existing sensors in smartphones that gauge acceleration on the X, Y, and Z-axis to create a widget (see figure 5). That the user must navigate to the page, turn their widget on, and adjust the volume to an appropriate level as they see fit means that the intention of the user to emulate the striking of a drum is relatively assured. Moreover, the user has the ability to decide which virtual “gong” they want to strike by selecting from options in a drop down menu. The result was twofold. Firstly, by using this widget, users were able to get their faces away from their screens and look around as they milled about their space, striking virtual gongs by enacting an analogous motion with their devices in hand. Second, the fact that the options for gong strikes were 1-9 and that each number corresponded to a specific speaker and, by extension, that speakers specific tone, meant that the kinetic act of striking the gong revealed a hidden aspect of the installation, namely the tonal breakdown of speakers in space. While the sonic results of this widget were far from the most sonically interesting, they may have produced the most interesting behavioral results. Users were palpably delighted by the instant translation from motion to sound that their actions created. I even recall, on a subsequent date from the opening, an interaction between two young men engaging in a highly physical imaginary battle, using their iPhones as sonic light sabers. While such levels of immediacy were not logistically possible with all of the widgets, the fact that some of them provided a relationship with their technology that was at once typical and totally singular was gratifying. It also meant that, the more time a user spent with the installation, the more they
uncovered about the ways in which it functioned, glitches and all, and the better apt they were at navigating their possibilities.

Figure-5

The “strike” widget allowed users to choose from a list of possible speakers (drop-down menu, top middle), initiate its potential audibility (button, top right), and control its volume (slider, right).

The slider on the left was a visual representation of the y-axis acceleration of the device at any given moment. If a user caused the acceleration value to reach a certain threshold by physically moving their device, the tone would sound.

Writing in code that limited the speed at which any given user could manipulate the installation was conceptually discontinuous with certain aspects of the project as a whole. In other words, I was hesitant to insert my aesthetic opinions onto the very aspect of the work I intended to be determined by people other than myself. However, in certain circumstances, some boundaries were absolutely necessary. For example, the widget I spoke about earlier that allowed the user to speed up the basic tchime pattern had to have some level of built in
restraint. Whether prompted by a user or not, the transitions between each section of the tchime pattern took place over 17 seconds. This meant that, should a user press the button that caused the pattern to move to the next section, they would not immediately hear the result. I anticipated that in the absence of an immediate effect and, given the instant gratification we have come to expect from our intelligent devices, an impatient user would press the button again and again until the result manifested itself aurally. Logistically, this would cause an internal collapse of the composed progression onto itself, resulting in a myriad of computational hang-ups and the potential for unpleasant sonic artifacts. To combat this, I coded in a time restraint on this particular widget that only registered touches every 20 seconds. If a user pressed the “fast-forward” button without allowing sufficient time to go by, not only would their actions have no effect, but they would also see an image of a dinosaur displayed in ASCII symbols on their widget, essentially telling them to slow down and pay attention. This had the double effect of preventing the program from becoming overwhelmed and reminding the user to engage mindfully and that their actions have audible consequences. In an effort to ensure the desired effect of creating a meditative and cooperative space, certain restrictions had to be put in place. In this sense, a balance between providing the user the feeling of immediacy while at the same time guiding them towards good performance was navigated. However, by the very nature of the differing functionality of the various widgets, this balance could not be achieved completely within each individual widget, but rather over the entire widget library as a whole.
Still, it must be noted that the failure on my part to completely predict and account for the different ways in which people used the interfaces while the installation was running engendered a number of interesting social realities. Short of a soft opening which, in retrospect would have been a valuable experience, I had no way of knowing how the program would function once upwards of 30 people were interacting with it at once. As the sounds became overly dense, the program became overloaded and lost the ability to effectively handle all the incoming messages. While this was officially an undesired side-effect, the communication amongst the participants that resulted was exactly what I was going for in that a complex relationship between myself as composer, the technology itself, the environment, and the interacting individuals was created. Where before there was no sound and not necessarily any connection between the group of individuals present, the glitch space and the desire on the part of the users to understand that space proved to produce highly fertile grounds for interaction.
Physical Layout

Considerations concerning the modes of perception of the pieces necessarily include more than just the auditory content and schema, but also the visual and the physical. When deciding how to portray the loudspeakers, I had to consider what a potential participant would be experiencing. Both Max Neuhaus and Jim Hart\textsuperscript{22} found it an important aspect of their work that the sources of the sound (i.e. the loudspeakers and the wind chimes, respectively) be hidden from plain view. Both artists believed that the lack of visual cues aided in the found nature of their installations and provided the opportunity for the listener to suspend their belief about what might be the cause of the sounds they heard. Splitting with their aesthetic choices, I chose to leave the speaker cabinets in plain sight, while suspending them in branches high above the listener's heads.

In creating \textit{tchime tree}, I had no desire to leave the mechanism that was forming the sound behind the curtain. While the content and character of the sounds produced were central to the perception of the piece as a whole, its existence as a tangible object in a particular space was equally, if not more important. If one of the main goals in creating an interactive installation is to carve out a space where people are interacting with their sonic environment and their fellow users, any attempt to hide the fact that there does indeed exist a sound installation at that location would be counter-productive. In an effort to provide

\textsuperscript{22} artist who created \textit{The Singing Oak}, the piece by which \textit{tchime tree} was originally inspired
a relatively equal opportunity for all willing participants to interact, anything that grounds the users in the physical and temporally reality of their situation serves to solidify the bedrock upon which interaction can occur. Moreover, speakers hanging in the air above the heads of the potential users provides them something tangible to look at other than their phones or the ground, and if the goal is to foster interaction, anything that gets people looking around is probably a plus.

Still, the installation had to not be an eyesore, for to make the speakers overly visually stimulating would serve to distract from the auditory field in which interaction was to take place. In addition, in an effort to not intrude upon the aesthetic comfort of passersby who may not be interested in participating, the speakers had to blend in relatively well with their environment. It was for all these reasons that I decided to build the speaker boxes out of composite aspen board, which I believe fits nicely between looking natural and manmade. The rugged, slightly utilitarian design also served to further remove the installation from any kind of commercial experience, emphasizing the specificity of the intended aesthetic occurrence. The decision to not cover the faces of the speakers with any fabric was also intentional, for to do so would be to give the boxes too strong an identity as objects in and of themselves and counterproductively hide the fact that these were indeed speakers hanging from trees.
While there were nine speakers all together, there also existed within them three smaller subgroupings of three speakers each. The point of this breakdown was to provide the possibility for what Dennis Smalley calls “nested space,” or smaller spaces within a larger space. Nested spaces create the potential for sound material to interact with behavioral variance from the larger sound environment. For Example, while the tchimes were spread out amongst all nine speakers in an effort to create a coherent whole, the sample tchimes jumped around nested groups of three so as to evoke the sense that smaller more specific events were occurring within the larger space. Furthermore, in an effort to create a physical ground on which interaction could occur, the speakers roughly demarcated a center arena type space in which I envisioned sonic exchanges might transpire.

---

23 Smalley, pg. 55
24 See appendix for images of the installation
Conclusion

One of the major points of discomfort that the outcome of *tchime tree* brought me was the fact that I was only able to offer the possibility of interaction to people who owned or could borrow the appropriate devices. One of the main things I did not want to do in creating a space was to exclude a certain group of people for any reason. I fully recognize that by only creating digital interfaces I ended up doing just what I sought out to avoid. However, I will point out that the creation of digital interfaces is the most cost effective. Instead of making physical objects, Interface Server allowed me to convert devices that individuals already had into tools for interacting with the piece. Had I had more time and should I ever repeat this project, I will either create the possibility for users who want to interact but have no access to the technology needed to borrow said technology. Furthermore, I would be interested in creating some kind of simple analog interface that would function as a structural component of the installation itself, creating the possibility for more modes of interaction that offer themselves to individuals of all socio-economic status. Another unintended occlusion was the fact that to interact with the installation one needed to be able to connect to the campus Wi-Fi and to do so required an institutional password. While this was avoidable given the current state of the technology, I was not able to prevent it by the time of the opening. Given the circumstances, this was not detrimental to the realization of the piece, but it still served as a point of conceptual dissonance. In addition, at the time of the opening I was not able to send client specific
messages to Interface Server. What this meant was that many of the widgets which should have functioned with a unique relationship to the specific user operating them were in practice treating all the users as one. As a result, certain widgets did not display information that was accurate to their sonic effects. While this was simply a bug that has since been fixed, it did create an interesting atmosphere in which users were confused as to whether or not what they were doing was actually effecting the installation, or if it was the actions of another user. This confusion was not intentional, but it did inspire an environment in which some found it necessary to communicate vocally with other participants, thereby establishing a deeper level of social interaction, a stated goal of the piece at large. Furthermore, the logic of the code broke down after a lot of outside interaction with the interfaces, preventing the basic tchime pattern from functioning to its full composed potential. The reality of the glitch space this engendered was not however altogether negative. Throughout the weekend, people would come up to me with a wide variety of reports. Some would say they walked by the site and no sound was playing at all, while others, reporting on the same day, would share experiencing no sound at first, but said that once they started up a widget, the piece resumed in full force. Overall I was pleased with both the reception of the installation as well as the modes of interaction, both anticipated and not. Should I revisit the piece, I would first and foremost figure out how to update the code so as to accurately treat each user with their respective client IDs as indexed by Interface Server. This would allow users to retain their sonic identity without unnecessary muddling due to faults in the
code. I do believe, however, that given the nature of the sound material there would still be some level of ambiguity should multiple users try to interact at the same time. Still, if it were functioning to its full potential, individual users would have a cleaner ability to navigate that ambiguity and through it learn more about the piece as well as the aesthetic preferences of their co-experiencers.
Acknowledgments

I would first like to thank the authors of all the free source code that made this installation possible. For Processing, I would like to thank Marcel Scwittlick, author of the YahooWeather library, the code structure that allowed me to receive wind data live. I would also like to thank Andreas Schlegel, author of the Processing library oscP5, an open sound control implementation that allowed me to send the data to SuperCollider. Much thanks goes out to Charlie Roberts, author of the Interface.js and Interface Server.js code, which made interaction with the installation possible. Also, I would like to express my gratitude to developers everywhere who have made their code free-source. Being someone who is new to the world of computer programming, the fact that I didn't have to start from scratch with this project was huge. That there are people out there who want to share their work and allow others to build off it makes me optimistic for the future of the art world in general.

Thank you Mark Gawlak and the rest of the CFA staff for being supportive of my desire to use the space for this project. Your openness to my vision (shout out to Mark) is greatly appreciated.

Thanks to Bob Russou and Bobby Khul for your logistical recommendations and general all-purpose sound-guy knowledge. I've learned a lot from you over these past few years, especially you Bob. Thanks for that.
Thank you to Paula Matthusen and all my other professors and colleagues in the music department for expanding my notions of what music can be.

Special thanks goes out to my advisor, Ron Kuivila, for all his help throughout this whole process. You’ve taught me so much about things I never thought I could do and never even knew I was interested in. Your practical and theoretical advice has forever changed the way I interact with sound and its potential to pervade the worlds that surround it.

Shout out to Noah Rush and the rest of the SuperCollider users group for the inspiration, motivation, and help coding. Thanks to David Preddy for making me realize that installation art is even an option. You guys are the coolest.

Thank you to all my friends and family who supported me throughout this process. Coding can be an oppressively solitary act at times, and after a long day starting at a computer screen, I’m grateful to have people in my life to reassure me that I am a human. Special thanks goes to my housemates Jon Lubec, Scott Zimmer, Jessie Napier, Delphine Starr, and Jackie Soro for being there for me in my moments of distress and putting up with the strange noises that emanated from my bedroom for months on end. You all are the best.
Stash, you’ve been my rock throughout this whole process. You believed in me even when I didn’t and was ready to give up. I don’t know what I would have done without you. I love you.

Last but not least thank you to my family (including Sean C.) who helped me set up the installation on the day of the opening. Special thanks goes out to my Dad, Alan Greenberg, for helping me with the design of the speaker boxes and for lending me his studio for the building process. Thank you both Mom and Dad for being artists and supporting me these past four years. Your aesthetic preferences have rubbed off on me and I wouldn’t have it any other way. It all came together because of you guys, I love you all and my gratitude is eternal.

Thank you to everyone who came to the installation and interacted with the piece. The purpose of this project was to create something that could be shared within the community it created. Without interested parties, the piece would have only manifested a fraction of its true identity. It is because of all of you that my vision was realized.

Thank you!
Bibliography

Works Cited


https://github.com/charlieroberts/interface.js


Works Consulted

Appendix

Pictures of the Installation

The opening:

The box containing the amplifiers, interfaces, and the host computer running the program
One Week Later:
Processing Code:

Yahoo Weather Query:\(^{26}\):

```java
import com.onformative.yahooweather.*;
import oscP5.*;
import netP5.*;
OscP5 oscP5;
NetAddress myRemoteLocation;
YahooWeather weather;
int updateIntervallMillis = 60000;

void setup() {
    oscP5 = new OscP5(this,12000);

    myRemoteLocation = new NetAddress("127.0.0.1",57120);
    frameRate(0.5);
    size(700, 300);
    fill(0);
    textFont(createFont("Arial", 14));
    //638242= the WOEID of Berlin
    // use this site to find out about your WOEID : http://sigizmund.info/woeidinfo/
    weather = new YahooWeather(this, 2450408, "f", updateIntervallMillis);
}

void draw() {
    try {
        OscMessage myMessage = new OscMessage("/weather");
        myMessage.add("/wind_speed");
        myMessage.add(weather.getWindSpeed());
        myMessage.add("/location");
        myMessage.add(weather.getCityName());
        myMessage.add("/wind_temp");
        myMessage.add(weather.getWindTemperature());
        myMessage.add("/wind_dur");
        myMessage.add(weather.getWindDirection());
    }
```

\(^{25}\) For the sake of space, I only included a small fraction code used for the installation. For a complete version of the code, please refer to the accompanying CD-ROM

\(^{26}\) The function of this particular code is to gather wind data from yahooweather.com and send it in the form of OSC messages to SuperCollider.
oscP5.send(myMessage, myRemoteLocation);

weather.update();
background(255);
text("City: "+weather.getCityName()+"; Region: "+weather.getRegionName()+"; Country: "+weather.getCountryName()+"; Last updated: "+weather.getLastUpdated(), 20, 20);
text("Lon: "+weather.getLongitude()+"; Lat: "+weather.getLatitude(), 20, 40);
text("WindTemp: "+weather.getWindTemperature()+" WindSpeed: "+weather.getWindSpeed()+" WindDirection: "+weather.getWindDirection(), 20, 60);
text("Humidity: "+weather.getHumidity()+" Visibility: "+weather.getVisibleDistance()+" pressure: "+weather.getPressure()+" rising: "+weather.getRising(), 20, 80);
text("Sunrise: "+weather.getSunrise()+" Sunset: "+weather.getSunset(), 20, 100);

} catch (NullPointerException e){
  println("somethings wrong!");
  setup();
}

public void keyPressed() {

OsMessage myMessage = new OsMessage("/weather");

myMessage.add(weather.getWindSpeed());

// myMessage.add(123); /* add an int to the osc message */
// myMessage.add(12.34); /* add a float to the osc message */
// myMessage.add("some text"); /* add a string to the osc message */

oscP5.send(myMessage, myRemoteLocation);
}
Interface.js Code:

**Evolver**:27

```html
<html>
<head>
<script src='interface.js'></script>
<script src='interface.client.js'></script>
<script src='interface.server.js'></script>
</head>
<body>
<script>
/////////////////////////////////panel///////////
panel = new Interface.Panel({ useRelativeSizesAndPositions: true });
panel.background = "#008B8B";
pl = new Interface.Label({bounds:[.25,.01,0.5,.1], stroke: 'rgb(216,191,216)', size: 60, style:'bold', hAlign:'center',vAlign:'middle', value:'~evolve~'});

/////////////////////////////////button///////////////////////////////////
a = new Interface.ButtonV({ bounds:[0,0.05,1,1], points: [{x:0.1,y:0.35},{x:.55,y:0.35},{x:0.55,y:0.2},{x:0.9,y:0.5}, {x:0.55,y:0.8},{x:0.55,y:0.65},{x:0.1,y:0.65}, {x:0.1,y:0.35}], mode:'contact', stroke: "rgb(255,235,205)", fill: "rgb(240,248,255)", label:'push',textLocation : {x:.5, y:.5},textSize: '50', target:"OSC", key:'/evolve_forwards'
});
al1 = new Interface.Label({bounds:[0.01,.8,.6,.1], stroke: 'rgb(255,235,205)"; fill: "rgb(240,248,255)"; label:'push',textLocation : {x:.5, y:.5},textSize: '50', target:"OSC", key:'/evolve_forwards'
});
al2 = new Interface.Label({bounds:[.5,.9,0.7,.1], stroke: 'rgb(255,235,205)"; size: 30, style:'bold', hAlign:'left',vAlign:'middle', value:'^{press this to move things along}^'
});
</script>
```

27 This code represents the visual and functional components of one Interface widget, posting GUIs to a webpage and sending and receiving OSC messages.
value:"", stroke: "rgb(216,191,216)",
size: 15,style:'bold'});

d6 = new Interface.Label({
  x:0.56, y:0.96, width:.2, height:.2,
  value:"", stroke: "rgb(216,191,216)",
  size: 15,style:'bold'});

/////////////////////////////////////////////////////////////////////
s = new Interface.Label({
  bounds:[.05,.3,.5,.2],
  style:'bold',size: 30,
  fill: 'white', stroke: "rgb(216,191,216)",
  hAlign:'left',
  value:""
});
l = new Interface.Label({
  bounds:[0.05,0.05,.25,.1],
  style:'bold',size: 20,
  fill: 'white', stroke: "rgb(216,191,216)",
  hAlign:'center',
  value:""
});

/////////////////////////////////////////////////////////////////////
panel.add(pl);
panel.add(a, al1, al2);
panel.add(b, bl);
panel.add(g);
panel.add(d1,d2,d3,d4,d5,d6);
panel.add(s,l);

/////////////////////////////////////////////////////////////////////
Interface.OSC.receive = function(address, typetags, parameters) {
  if( address === '/location' ) {
    g.setValue('location: '+parameters[0]);
    //Interface.OSC.send('/test', 's', parameters);
  }
  if( address === '/slow_down_evolve' ) {
    s.setValue(parameters[0])
  }
  if( address === '/evolve_client' ) {
l.setValue("acting client is number: "+parameters[0]);
}
if( address === '/a' ) {
    d1.setValue(parameters[0]);
}
if( address === '/b' ) {
    d2.setValue(parameters[0]);
}
if( address === '/c' ) {
    d3.setValue(parameters[0]);
}
if( address === '/d' ) {
    d4.setValue(parameters[0]);
}
if( address === '/e' ) {
    d5.setValue(parameters[0]);
}
if( address === '/f' ) {
    d6.setValue(parameters[0]);
}

</script>
</body>
</html>

/////////////////////////////////////////////////////////////////////
</script>
</body>
</html>
SuperCollider Code:

**Interfaces**

```
~enable_interfaces.value;

~enable_interfaces = {
  ~interface = NetAddr.new("127.0.0.1", 8083);

  OSCdef(evolve_forwards).clear;
  OSCdef(evolve_forwards, { ...x| var message,clientID,time,name,value,loc;
    message = x[0]; time = x[1]; clientID = message[2];
    value = message[1];

  });
```

This code represents how SuperCollider receives OSC from specific Interface widgets, interpreting them and evaluating functions accordingly while sending messages back to the interface.
```
~interface.sendMsg("/location",~current_location);
if(~time_check.isPlaying==false, {~time_check.play;},{"already going"});
if(time - ~check_time < 10, {"slow down, too fast yo.";~dino.value;},
/*loc = ~current_location.asString;
~interface.sendMsg("/location",loc);
*/
{
if(~evolution.isPlaying,{
    ~next.prStop; ~next.reset; ~next = nil;
    ~next = (Routine{
        [("node: "++~current_node), ("location: "++~current_location)].postln;
        60.wait; "moving on...".postln;
        ~evolution.next; "reset".postln.yieldAndReset;
    });
    ~next.reset;
    ~evolution.next;
    
    });
["client: "++clientID, "forwards: "++value,"time: "++time].postln;
}, "/evolve_forwards");
); //OSCdef(\evolve_forwards).enable;
////////////////////////////////////////////////////////
//OSCdef(\evolve_play).clear;
OSCdef(\evolve_play, { | ...x|
    var message,clientID,time,name,value,location;
    message = x[0]; time = x[1]; clientID = message[2];
    value = message[1];
    if(value==1, {
        if(~evolution.isPlaying==false, {~make_evolve.value;},{"already playing"};)
    });
```
"client: "++clientID, "evolve playing: "++value, "time: "++time].println;
}

');

OSCdef(evolve_play).enable;

};//end evolution interface

///////////////////////////////////////////////////////////////////////////////////

-50-
Weather\[29\]:

```plaintext
// processing file must be running for this to work...

~weather_listen.value;
//~weather.disable;
//~weather.enable;

// evaluate below...

class~weather
        {"OSCdef(weather).clear;
                OSCdef(weather, { | ...x| var time, message, location, wind_speed, wind_direction, wind_temp;
                                var wind;
                                time = x[1];
                                message = x[0];
                                location = message[4];
                                wind_temp = message[6];
                                wind_speed = message[2];
                                wind_direction = message[8];

        });

        ~wind_mph = wind_speed;
        ~wind_direction = wind_direction;

        if(wind_speed < 1,{wind_speed = 1});

        wind = wind_speed.linlin(0,50,0,5);
```

This code represents how SuperCollider receives wind data from Processing in the form of OSC messages and updates the values of different synthesis functions running on the SC server accordingly.

\[29\] This code represents how SuperCollider receives wind data from Processing in the form of OSC messages and updates the values of different synthesis functions running on the SC server accordingly.
Pdefn(tchimes_rates, Pwhite(0.0025*wind,0.5*wind,inf));
Pdefn(tchimes_rates_dur, Pwhite(1/wind,25/wind,inf));
Pdefn(tchimes_freqs_dur, Pwhite(50*wind,100*wind,inf));
Pdefn(tchimes_pan, Pwhite(0.0025*wind,0.05*wind,inf));

Pdefn(buffer_tchimes_trig, Pwhite(0.0025/wind,0.5/wind,inf));
Pdefn(buffer_tchimes_trig_dur, Pwhite(0.5/wind,2.5/wind,inf));
Pdefn(frame_jump_dur, Pwhite(5/wind,10/wind,inf));
Pdefn(buffer_tchimes_bufrate, Prand([0.25,0.5,0.75,1,1.25,1.5],inf,0));
Pdefn(buffer_tchimes_pan, Pwhite(0.0025*wind,0.05*wind,inf));

["wind_speed: "+wind_speed,"wind_direction: "+wind_direction].postln;
}

"/weather";
});

*/
};

/**See accompanying CD-ROM for a more complete code library*/

-52-