View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by DigitalCommons@WPI

Worcester Polytechnic Institute
Digital WPI

Interactive Qualifying Projects (All Years) | Interactive Qualifying Projects

September 2015

Multichannel Sound Installation IQP

Brede Emil Doerner
Worcester Polytechnic Institute

Leif Erik Skramstad
Worcester Polytechnic Institute

Trevor Anthony Rocks
Worcester Polytechnic Institute

Yutong Li
Worcester Polytechnic Institute

Follow this and additional works at: https://digitalcommons.wpi.edu/iqp-all

Repository Citation

Doerner, B. E., Skramstad, L. E., Rocks, T. A., & Li, Y. (2015). *Multichannel Sound Installation IQP*. Retrieved from https://digitalcommons.wpi.edu/iqp-all/1968

This Unrestricted is brought to you for free and open access by the Interactive Qualifying Projects at Digital WPI. It has been accepted for inclusion in Interactive Qualifying Projects (All Years) by an authorized administrator of Digital WPI. For more information, please contact digitalwpi@wpi.edu.

MULTICHANNEL SOUND INSTALLATION:

An Interactive Qualifying Project Report
submitted to the Faculty
of the
WORCESTER POLYTECHNIC INSTITUTE
in partial fulfillment of the requirements for the
Degree of Bachelor of Science
by:

Leif Skramstad
Trevor Rocks
Brede Doerner
Yutong Li

Project Advisor:
Professor Frederick Bianchi

Date: 1 September 2015

Abstract:

The purpose of this project was to advance the development of a multi-channel audio system that plays various wind-chime sounds based on external wind speed and direction data. The project draws upon the works of prior groups under the direction of Professor Frederick Bianchi at Worchester Polytechnic Institute, contemporary works of artists and other academics, and individual research and experimentation. This project should asses 1) the feasibility of constructing and installing a multi-channel weather driven sound-art exhibit, and 2) the immediate and long term implications of developing and deploying such an exhibit.

Table of Contents:

Authorship:

Every section of the proposal was written and edited by each member of the group collaboratively. The primary authors of each section are listed below.

Editing & Formatting: Leif Skramstad

Abstract: Leif Skramstad

Introduction: Leif Skramstad

Background:
- Existing Sound Installations: Brede Doerner
- Philosophy and Design: Yutong Li, Leif Skramstad
- Environmental Sound and Medicine: Trevor Rocks

Methodology:
- Sensor and Basic Programming: Trevor Rocks
- Recording: Leif Skramstad
- Sonic Mapping: Brede Doerner

Future Implementation:
- Programming: Trevor Rocks, Leif Skramstad, Brede Doerner
- Sensors: Trevor Rocks, Leif Skramstad
- Audio Devices: Leif Skramstad
- Locations, Uses, and Other Considerations: Leif Skramstad

Conclusion: Leif Skramstad

Introduction:

In the public consciousness, nature plays an important role in the mental, physical, and overall health of the community at large. We see time and again that people place enormous value on the ability to access the environment around them or, at least, access spaces that replicate the feelings of the natural environment. One needs to look no further than Central Park in New York or any other national park or protected space in order to see this. The common sense belief that spending time with nature is important for one's health- particularly one's mental health- is strongly held. This notion is largely supported by historical analysis and, more recently, by scientific research. Stephen Kaplan, of the University of Michigan, conducted a study about the relationship between "directed attention", stress, and the environment. He summarized his findings by saying: "Experience in natural environments can not only help mitigate stress; it can also prevent it through aiding in the recovery of [directed attention.]" (Kaplan 8).

The need for such natural stimuli in urban environments spurred us to take on the problem of connecting people inside a building to the weather occurring outside. Specifically, we have advanced the creation of a multi-channel audio system that plays various wind-chime sounds based on external wind speed and direction. The audio system itself was intended to have forty separate channels in order to provide a truly unique surround sound experience for any user. While the number of channels was relatively arbitrary, we wanted it to be large enough to push the boundaries of what is possible for one's immersion in an audio experience. This audio system, when coupled with live input from a weather station located outside, will allow occupants to feel more connected to their environment and hopefully reap the benefits of such an experience. In addition to its obvious uses as a greenspace or art installation, the technology and methodology incorporated within our project can also be used to inspire or directly advance the fields of entertainment, communication, or scientific research. The benefits of such a project can appear somewhat abstract or insubstantial, but as we will show; the viability, adaptability, and promise of this project makes such an effort not only well justified, but timely and necessary.

Background:

Existing Sound Installations:

While examples of environmental art installations are plentiful, there are not yet very many examples of environmental sound art installations and few that use the sonification of environmental data which this project seeks to study. A good example which seeks to accomplish similar goals to this project is the *Rainbow Cove* [1] installation at Logan airport. Its purpose is to calm travelers by using sound to put them in a more calming environment than a typical airport. The sounds used are of natural origin such as frogs and crickets as well as man-made objects such as a distant fog horn, which together could allow the traveler to feel as if they are outdoors in a natural environment near a body of water. Although the ultimate goal of this installation is soothing travelers, it accomplishes this by using sound to bring parts of a more natural environment inside an otherwise very unnatural environment.

An example of a different method to place the observer or listener in a different environment using sound is the exhibit *Silent Echos* [2] by Bill Fontana, in which he places microphones directly inside old bells which are outdoors in abandoned temples. The only sounds present are the sounds of the environment which in this case is forest. However, these sounds are modified by being recorded inside the bells which serves to completely change the environment that the listener is reminded of when they listen. This illustrates and takes advantage of the fact that it isn't only the types of sounds that are present that influence what we think of when we hear them, but also the environment or simulated environment that they are recorded or

---

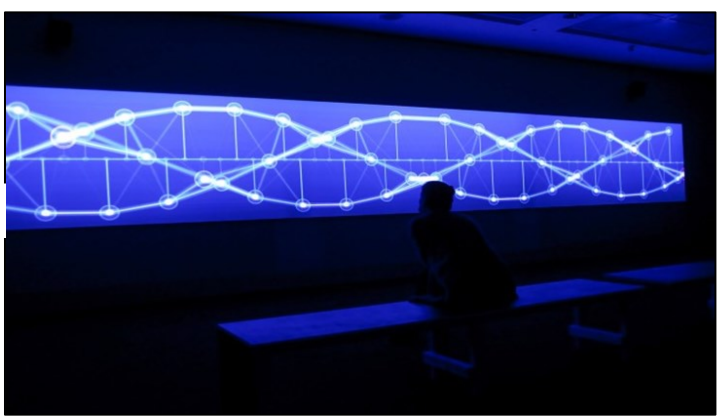
[1] See Ngowi for more on the exhibit.

[2] Fontana, Bill. "Silent Echoes by Bill Fontana." *Silent Echoes by Bill Fontana*. N.p., n.d. Web. 01 Sept. 2015

synthesized in. This will be important to consider as this technique could be used to increase the effectiveness of the sounds we use or generate to make the subject feel as though parts of the outside environment are present indoors, such as wind in this project.

Simple Harmonic Motion #5 [3] by Memo Akten is a sound art installation that is different from most in that it sonifies data rather than simply manipulating and recording sound and then playing it back. Even more interesting is that it sonifies data that already could be listened to directly as sound, but by sonifying it in a creative way, rather than simply playing it back, the listener is able to hear and interpret properties of the data that would not have been noticed if the data was simply played as sound literally. This piece of art uses a number of different sine waves oscillating at slightly different frequencies. Due to the number of different waves it would likely sound like a big mess of sound if it was simply played as sound, however this data is instead sonified by making a plucking sound whenever a wave crosses the middle. This, combined with the accompanying visual, allows the observer to hear and see how many different waves or different frequencies will eventually come together again in phase with each other, a property which wouldn't be audible if this data was simply played back as sound. As this is a more visual exhibit, a picture of it in action has been included below in figure 1.

Figure 1



[3] Akten, Memo. "Simple Harmonic Motion #5." *The Mega Super Awesome Visuals Company Simple Harmonic Motion 5 Comments*. N.p., 26 Aug. 2011. Web. 01 Sept. 2015.

Philosophy and Design:

The philosophy behind our research has its roots in the Arts and Crafts Movement of the 19th Century, the works of Frank Lloyd Wright, and other proponents of "organic architecture." From 1880 to 1910 in Europe and North America a new movement in design called the Arts & Crafts Movement [4] took root. It was driven in part by John Ruskin and Augustus Pugin, who sought to counteract the cultural changes induced by industrialization through art. The movement's principal drivers felt that the quality of commercial products had decreased markedly after the introduction of machine driven production. Artists too, in their view, tended to cater to the efficiency and accuracy of mass production and had begun to ignore the important role of traditional craftsmanship in their work. In spite of its populist message and spiritual fervor, craft makers and artists who sought to turn the clock encountered little success, partly because their dreams of perpetuating the designs of the Gothic and Middle Ages were unrealistic and utopian. However, the Art and Crafts Movement did provide future artists with a reference for new design styles. The movement would now evolve from one focused on craftsmanship to one which involved the conflict and the relationship between nature and industry.

The idea of the Art and Crafts Movement spread to the United States and was closely linked to its European counterpart throughout the period surrounding the turn of the century. The Chicago School of Architecture was profoundly influenced by the movement and a new sort of philosophy and style in architecture began to develop. The representative figures of this new philosophy, "organic architecture", were Frank Lloyd Wright and Louis Sullivan. The prototype of the philosophy was first mentioned by an American sculptor, Horatio Greenough, who

[4] See Obniski for a more in depth summary of the Arts and Crafts movement.

advocated that there should be synergy between nature and architecture. He also argued that the beauty of nature should be inherent in the design of architecture. These ideas combined with Louis Sullivan's philosophy of "form follows function," became the most important principles of the organic architecture movement.

Louis Sullivan was influenced by the thoughts of American poet Walt Whitman and the liberal idea of free verse and democracy in his poems, as well as the ideas of Friedrich Nietzsche, to form an ideology that architecture should be used to uplift the human spirit. While credited with coining the famous phrase "form follows function," he incorporated more aesthetically pleasing elements into his designs than one might expect. Elements that his pupil, Frank Lloyd Wright, would later adopt.

Wright had a philosophy that was similar, but still distinct from the ideas of Sullivan. He mainly focused on the harmony between the buildings and the environment and the importance of the connection of nature and human spirit. The principle Wright insisted on in his work, was that architecture, or art design, cannot be complete without the surrounding environment. Frank Lloyd Wright introduced the word 'organic' into his philosophy of architecture as early as 1908. It was an extension of the teachings of his mentor Louis Sullivan whose slogan "form follows function" became the mantra of modern architecture. Wright changed this phrase to "form and function are one," using nature as the best example of this integration.

Although the word 'organic' now commonly refers to something which has the same characteristics as animals or plants, Frank Lloyd Wright's organic architecture uses a different meaning. His is not a style of imitation, as he never claimed to be building forms which were representative of nature. Instead, organic architecture is a reinterpretation of nature's principles as they are filtered through the intelligent minds of men and women, who could then use them to build forms which are more natural than nature itself.

Organic architecture involves a fundamental respect for the properties of the materials, (one should not twist steel to form a flower, for example) and a respect for the harmonious relationship between the form, design, and function of a building. Organic architecture is also an attempt to integrate the spaces into a coherent whole: a marriage between the site and the structure and a union between the structure and its context.

Environmental Sound and Medicine:

Environmental sound has been shown to have effects on the healing process of patients in hospitals. Noise can be defined as “a loud or unpleasant sound,[5]” and high levels of disruptive noise are shown to increase the heart rate and blood pressure of patients in a study on the “Effect of Environmental Sound and Communication on CCU Patients’ Heart Rate and Blood Pressure.”[6] The study measured the sound levels in a hospital room and cross-referenced the decibel levels with the patients’ heart rate and blood pressure at the time. Louder and more disruptive sounds, such as conversation in the room or conversation held nearby in the hallways were shown to cause greater levels of stress in the patients, while quieter environmental or background sounds, were shown to cause less of an increase in stress. This shows that more natural sounds from the environment can cause less stress than man-made or conversational sound. Another study, “Urban green spaces' effectiveness as a psychological buffer for the

---

[5] "Noise." *Merriam-Webster.com*. Merriam-Webster, 2011. Web. 12 August 2015.

[6] See Baker.

negative health impact of noise pollution: A systematic review"[7] showed that people were less disturbed by the sounds generated by green space then by the sounds created in an urban environment. By measuring the stress levels between people living near green space and living in more urban areas the correlation became clear that green spaces generate less disruptive noise. Both of these studies have led to the conclusion that subtle environmental sounds lead to a less stressed person. A study done on the effects of stress on human health showed a "significant relationship between daily stress and the occurrence of both concurrent and subsequent health problems such as flu, sore throat, headaches, and backaches." (DeLongis 1). This implies that lowering the stress level an individual feels can cause noticeable health benefits to the average person. By using environmental sound in the place of urban sound, stress in an urban environment can be lowered significantly enough to create health benefits in the urban population.

Using the rhythms of nature to create a sound installation allows us to recreate the calming effects of natural background noise outside of nature. While nature brings us a calming rhythm, that is not the only piece of the equation. As shown above, the sound being made is also important, mapping a natural rhythm to the sound of a jack hammer would do nothing to make it a more calming experience. By mapping these natural rhythms to a sound already known to be calming, wind chimes, we were able to create a relaxing sound installation, combining natural and artificial relaxants to create a new calming experience.

---

[7] See Dzhambov.

Methodology:

Sensor and Basic Programming:

The first step was to set up the Ambient Weather WS-2095 [8] wind sensor. The system has 2 components, the weather sensor in Figure 2...

Figure 2

[8] A basic amateur weather station with the wind measuring capabilities required for our project.

…and the receiver in Figure 3:

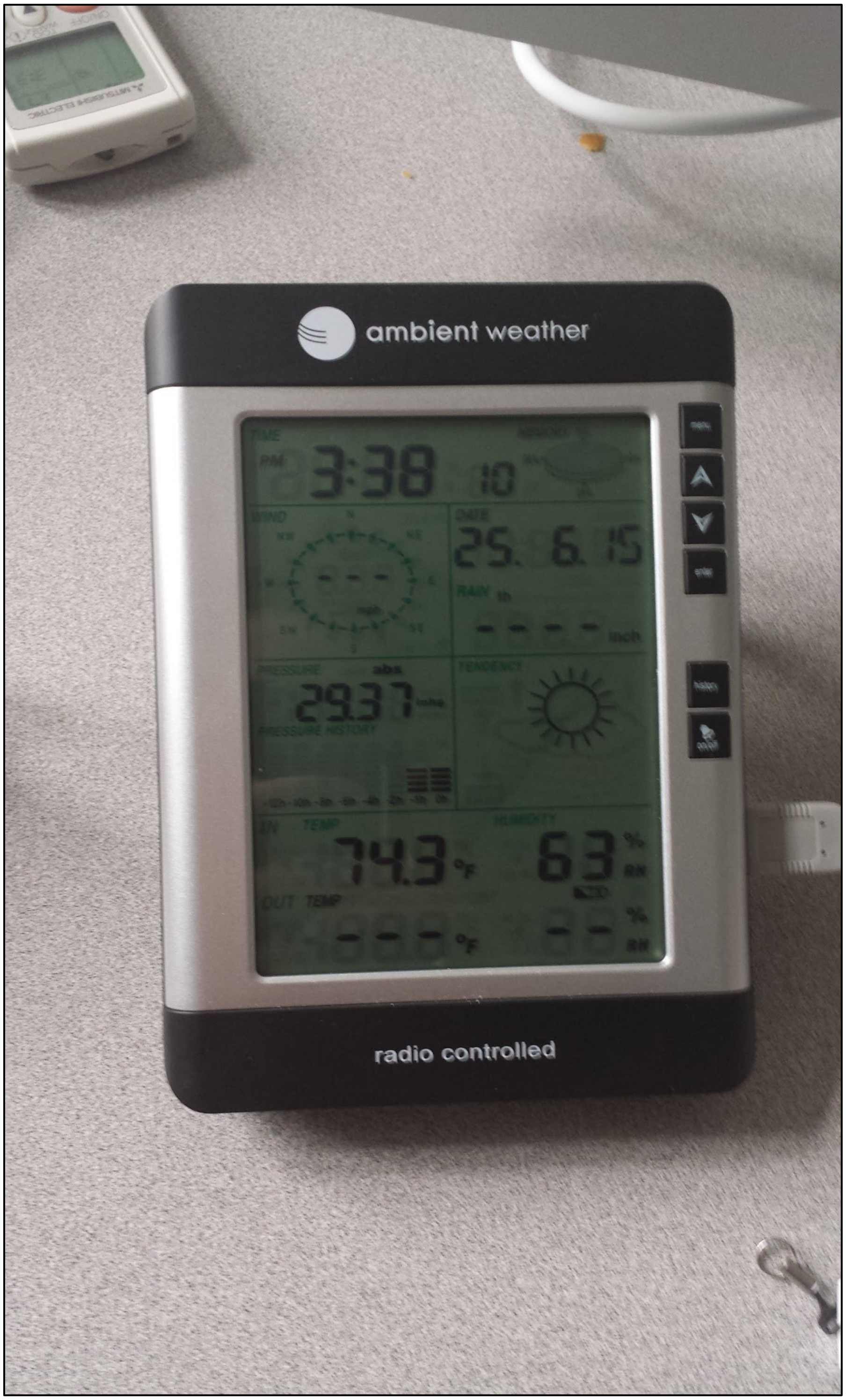


Figure 3

The weather sensor was assembled and fixed to a mobile mount for ease of movement for indoor and outdoor testing. The WS-2095 manual states that the optimum range for communication between the sensor and the receiver is a range greater than 10 feet and less than 300 feet (Ambient 10). Therefore, for testing, the sensor was placed outdoors and close to the lab room, just outside the window and within this range. This provided natural wind for testing in real conditions, and reliable communications between the sensors and the receiver.

For other testing scenarios, the sensor was placed indoors in order to be within reach of the tester at the computer. The receiver was connected to the computer using a USB cable. We used the computer program WeatherSnoop 3 to interpret data from the WS-2095 system. WeatherSnoop 3 is a general weather interpreting program for Macintosh computers capable of interacting with a number of different weather sensors. WeatherSnoop 3 is capable of processing a large number of different weather properties such as barometric pressure, indoor and outdoor temperature, measurements of precipitation as well as wind speed, direction, average speed and wind chill. Wind direction is given in degrees, but is limited to the 8 main points on the compass, north, north-east, east, south-east, ect. The system is capable of running an ample amateur weather station.

The highest polling rate the WeatherSnoop 3 supports is once per minute, meaning that we could only receive updated weather data once every 60 seconds. Initially we looked to receive data at a higher polling rate, but further research showed that the WS-2095 had a maximum polling rate of once every 48 seconds.

WeatherSnoop 3 came with built in compatibility to export data both to a web server and to AppleScript, a basic coding language used to create applications and processes on Macintosh computers. Using AppleScript commands to pull data from the WeatherSnoop 3 program fit our process best. After initial testing with AppleScript we began writing a program that could utilize the wind direction data gathered from the WS-2095 which was exported from the WeatherSnoop program. By identifying each of the possible 8 wind directions, we created a program that could open a one minute audio file based on the direction of the wind. By looping the program we were able to create a program that could continuously trigger audio files based on the direction and speed of the wind. After choosing the corresponding audio file it was then played by the QuickTime Player. See Figure 4:

```
AppleScript    <No selected element>

tell application "WeatherSnoop 3" --calls WeatherSnoop 3 program
	set file_path to (null) --ensures filepath is empty at start
	tell weather property "Wind Direction" of agent of first document --pulls value
			of windspeed
		if value = "0" or value = 360 then --tests for value equalling north
			set file_path to ("/Users/lab/Desktop/RocksIQP/Music/North.mp3") --
					sets filepath to the file labeled north
		else if value = "45" then --north east
			set file_path to ("/Users/lab/Desktop/RocksIQP/Music/Northeast.mp3")
		else if value = "90" then --east
			set file_path to ("/Users/lab/Desktop/RocksIQP/Music/east.mp3")
		else if value = "135" then --southeast
			set file_path to ("/Users/lab/Desktop/RocksIQP/Music/southeast.mp3")
		else if value = "180" then --south
			set file_path to ("/Users/lab/Desktop/RocksIQP/Music/south.mp3")
		else if value = "225" then --southwest
			set file_path to ("/Users/lab/Desktop/RocksIQP/Music/southwest.mp3")
		else if value = "270" then --West
			set file_path to ("/Users/lab/Desktop/RocksIQP/Music/west.mp3")
		else if value = "315" then --northwest
			set file_path to ("/Users/lab/Desktop/RocksIQP/Music/Northwest.mp3")
		end if

		tell application "QuickTime Player" to play (open file_path) --tells
				QuickTime Player to open selected file
		delay 60 --one minute delay
		tell application "QuickTime Player" to quit --closes QuickTimePlayer ending
				the file

	end tell
end tell
```

Figure 4

In addition to audio files of recorded wind chimes, a program/ algorithm was also developed to access a software simulated wind chime.

The program shown below in Figure 5.

```
AppleScript    <No selected element>
tell application "WeatherSnoop 3" to open "/Users/lab/Desktop/RocksIQP/IQP.wssite" --opens weathersnoop file for our site
tell application "System Events" to tell process "WeatherSnoop 3" to set visible to false --hides weathersnoop program for
    cleaner look
tell application "WeatherSnoop 3" --calls the WeatherSnoop 3 application
    launch file "/Users/lab/Desktop/RocksIQP/starter.scpt"
    run script "/Users/lab/Desktop/RocksIQP/starter.scpt" --ensures that weathersnoop is recording data
    tell weather property "Wind speed" of agent of first document --pulls value of windspeed
        get value
        log value --records and logs value for testing purposes
        repeat 5 times --contents within will repeat x times
            tell application "AppleScript Editor"
                launch file "/Users/lab/Desktop/RocksIQP/windAccessFile.scpt" --launches access file
                run script "/Users/lab/Desktop/RocksIQP/windAccessFile.scpt" --runs access group
            end tell
        end repeat
        if running then --stops recording data
            stop
        end if
    end tell
    tell application "WeatherSnoop 3" to close --closes weathersnoop at the end of run
end tell
```

This program was used to initialize and run the application using two separate helper functions. The first function, shown below in Figure 6, ensures that WeatherSnoop 3 is running and is pulling data from the correct weather station:

```
AppleScript    <No selected element>
tell application "WeatherSnoop 3" --starts recording data if not already recording
    tell agent of first document
        if running then
        else
            start
        end if
    end tell
end tell
```

Figure 6

The second helper function was shown above in Figure 4, which chooses the correct audio file to play based on the wind direction. After ensuring that the necessary elements are open and running on the computer this program loops the program shown in Figure 4 in order to repeatedly select and play sound every 60 seconds based on the wind direction.

Recording:

Early on in the planning process, we saw the need for a two pronged approach to creating the sonic portion of the project: one based on an algorithm that would simulate wind chimes being stuck to produce a unique stream of sounds, and one based on a simpler program that would play pre-recorded sounds based on wind data. In doing so we would establish a backup method of playing sounds in case we encountered trouble with creating or implementing our algorithmic program. To obtain the sounds needed for both approaches, we established the following objectives:

- To examine the waveform of the sound of a chime and apply observations to our programs to improve authenticity.
- To determine the feasibility of recording our own sounds with adequate quality and variety
- To determine the feasibility of downloading sounds that meet the same criteria
- To determine the feasibility of editing acquired samples for compatibility
- To obtain single, isolated tones for use as individual notes for our algorithmic program
- To obtain one-minute samples of sufficient quality and variability to be played by our simpler program

The first step in obtaining wind chime sounds was to compare the feasibility of recording our own samples to that of acquiring samples from a third party. Initial research produced limited results for digital copies of wind chime sounds online. Conversely, it was quickly established that microphones of sufficient quality and compatibility were readily available in the WPI Computer Music Labs. Coupled with the availability of a relatively well insulated sound lab, capable computers, and software, it quickly became apparent that recording our own tones was a much more appealing option.

Next, we had to establish a working setup for recording our sound samples. This setup consisted of a differential microphone and an amplifier connected via USB to a Mac desktop computer. In terms of software, we used QuickTime (for recording sounds) and Audacity (for editing sound files). Our microphone was placed on a stand, in front of which we suspended a wind chime that was struck by hand to create the desired sounds. While this setup was at times ungainly, it was simple and intuitive.

Our first recordings were short clips of random sounds to ensure that all of our devices were connected and working properly. In addition to ensuring basic functionality, we wanted to ensure that the microphone's gain was set at a level that was high enough so that sound could be picked up at an adequate volume- but low enough to ensure that we were neither picking up too much background noise, nor too many reverberations throughout the recording process. We quickly found, however, that our designated lab was not as insulated from environmental sound or sounds made by user error as we had hoped. To compensate, we adopted a strategy of bulk collection, by which we would record a large number of samples and select ones that had little to no unwanted noise. These processes constituted the majority of our feasibility study- by the end of which we were confident that we could proceed.

In recording our isolated tones, we encountered some difficulty in positioning the set of chimes so that only the desired chime was struck and produced sound. By its nature, the wind chime was designed to have multiple chimes struck with relative ease. This factor helped later with recording our 1-minute samples, but complicated matters when recording isolated tones. We also found that if one strikes a chime too hard, it can produce a sound that is somewhat distorted due to having too many overtones. This effect can be compounded when the created note is too loud for the microphone to register without distortion due to instrument limitations. By increasing the gain further, we could strike the chimes more softly and avoid these distortions at the cost of picking up slightly more environmental noise. We viewed this as a necessary tradeoff.

After these initial adjustments, we were able to record samples of a high enough quality for editing. Using QuickTime, we were able to record simple .aac files and then edit them in Audacity. Once recorded, we were then able to splice and rearrange our samples as needed and ensure that samples were of good quality. Using Audacity, we also were able to explore the possibilities of changing the pitch of a sample and mixing single tones to create artificial arrangements. After some time, we were able to conclude that the process of altering the pitch of certain samples and the process of arranging individual samples to create artificial arrangements were inferior solutions to our given problems. Instead, we found that it would be much simpler and more effective to record our own samples using an actual chime. In the future, we would recommend that future groups should try to obtain a variety of chimes and record sounds themselves rather than try to acquire an adequate sample by other methods.

Once we had obtained adequate samples, it was relatively simple to export them in a usable format for our program. Again, using Audacity, we were able to reformat our samples from a stereo .aac file format (with a single audio channel instead of the requisite two) to a mono .wav file format. Unlike our .aac files, these .wav files would be compatible with both our Applescript and Python programs, as well as easily edited and played on many different devices. Once these files were created, it was a simple matter to transfer them from one device to another for playback.

Later on in the project, we discovered that there would be a need for one-minute samples of wind chime sounds for our program to play due to a change in our program design. Instead of using a program that could synthesize incoming data and predict how each individual chime would be struck, we would need to settle for a simpler program, as our original program proved to be difficult to transfer from Applescript to Python. We also discovered that our selected weather instrument and others like it could only provide usable data once a minute. This required us to record samples that could be played throughout the downtime between data inputs.

Recording these one-minute samples proved to be relatively simple, as much of the groundwork had already been laid through our previous recording attempts. Recording remained a time consuming endeavor though, as it was likely that some form of random environmental noise would occur at some point during the sixty second period. Once eight adequate samples were recorded, they were each assigned a compass direction and labeled appropriately. The samples were then reformatted from .aac files to .wav files and exported. This concluded our recording process.

In adapting and performing this procedure, we were able to obtain the sound samples we needed with limited difficulty. We found that while these samples exceeded our expectations for quality and clarity, they did not possess the variety of tones we would expect in a real world installation similar to our project. In the future, we would recommend that one should record any desired sounds themselves, instead of attempting to acquire them from a third party. We believe that it can be much more painstaking to edit mediocre samples from a third party into something desirable than it is to record the finished product oneself, provided that one has the proper instruments and tools available. We would also recommend that one practices bulk collection when recording, to ensure that at least one good sample is obtained. Finally, we would recommend that one obtain one's recordings in a lab that is well insulated from unwanted environmental noise to make collection less tedious.

Sonic Mapping:

While using pre-recorded sound samples was a relatively simple solution for this installation, a more flexible solution is to synthesize these samples. This would allow for many more variables to influence the sound in more advanced installations as well as allowing the variables to be more continuous rather than just a handful of several discrete values. For an installation in which all sounds are completely synthetic, or in other words not recorded at any stage, the audio quality does not depend on the recording equipment and environment which can get costly if very high sound quality is desired. Instead, the sound quality of a fully synthetic system would only depend upon the effectiveness of the algorithm used to create the sound.

There were two general components in the virtual model of the wind chimes: 1) being the synthesis of the sound the individual chimes make when they were struck, and 2) the control model. There were, however, options in how the sounds of the individual chimes would be created. In a fully synthetic model, the sounds of the chimes would be computer generated.

As wind chimes are effectively random a stochastic model was used to model them. This model was made using an example from the paper "Modal Synthesis of Wind Chime Sounds with Stochastic Event Triggering" by a team from the Helsinki University of Technology. Their model used only one variable, the energy of the system, which was easy to determine given a function for wind speed. The system energy was used to determine the probability, p, that an event should be triggered. Then, at a randomly varying rate (every 0.03 seconds to 0.05 seconds as used in the example outlined in the model), p was used to determine what happened in the system, which was essentially a finite state machine (FSM). In the example there were six states as five chimes were used. State zero had the pendulum in the middle not hitting anything. States one to five corresponded to the pendulum hitting one of the five respective chimes. From state zero there was a p/5 chance it would move to one of the other states, and a 1-p chance that the system would stay in state zero. From states one through five there was a p/2 chance the next state would be one of the adjacent states (one to five or two for example) and a 1-p chance it would stay in state zero.

Synthesizing the wind chimes themselves proved to be very difficult with the methods described in the paper mentioned above. It created a transfer function for each chime which can be determined from real chimes using recordings. However implementing this with arbitrary chimes was very time consuming as it was very difficult to predict how tuning the parameters of the equation would affect the sound. The paper did however specify that the impulse response of the system should be a series of harmonic decaying sinusoids (Lukkari 3). These proved to be fairly easy to synthesize in a much more simple way that allowed for easier tuning to achieve a realistic sound.

The implementation of this model was done in MATLAB, as it has many mathematical tools built in which gave a lot of flexibility in implementation. The implementation was also made modular, with the sound synthesis being separate from the control part of the model. Both of these decisions made it easy to explore many different approaches, and the latter made it easy to try the different configurations of using pre-recorded chime sounds vs. synthesizing them.

While implementing the sound synthesis component was easy in MATLAB as it is very good at generating waveforms from a variety of types of functions, the stochastic model of the wind chimes proved to be more challenging. MATLAB isn't well suited for making state machines, especially when each state needs to perform an action after the state has changed such as a chime continuing to create sound after the pendulum has moved away from it as had to be done here. This was tackled by only using traditional FSM architecture for the simple state machine outlined above, which in the example had six states, and then effectively making separate state machines for each chime which then interact with the first state machine by doing nothing after they have reached the end of their chime sound, playing the next sample of sound when nothing happens, and then starting playback from the beginning whenever they are triggered by the simpler state machine. This effectively creates an extremely large and complicated state machine, especially for implementations with a lot of chimes, but without having to manually encode each state.

Future Implementation:

Our current setup, in spite of the progress we have made, is not a finalized version of the solution we are advancing. Currently our installation is using wind chime sounds that are either prerecorded or pre-simulated and are not a live stream simulation of a wind chime. Because of this, the sounds generated by our system are not quite as random as we would like them to be in a real world installation. Additionally, our current system is not capable of playing its output audio across forty different audio streams. Our focus was instead to develop a feasible method for triggering wind chime sounds based on weather data, as ensuring that an audio stream could be broadcast on forty separate channels would be pointless without ensuring that an audio stream existed in the first place.

Programming:

Due to our simulation in MATLAB being incompatible with a Macintosh computer we took the approach of using audio files simulated then played on a computer instead of a real time streaming simulation. With further time and research it would be possible to bridge this gap, and have the installation running a continuous stream of data into the simulation in real time. Our best recommendation to achieve this would be to use either a higher end wind sensor, or a custom built sensor in order to have a constant stream of wind data which could then be implemented in a simulation algorithm. Doing so could allow one to simulate a higher polling rate than can be done locally with WeatherSnoop 3. By doing this, or through the use of more advanced programming that could build off of a single data point to create sounds for an extended period of time, it would allow one to increase the variability of sounds generated without needing to replace any other hardware components.

Some improvements could be made to the synthetic wind chime model as well. With more time it could have been written in a high level object oriented programming language such as python or C++ which would allow it to run as an application. By using one of these languages it would be possible to model each chime as an object which would allow the removal of the "popping" sound whenever a chime is struck that is still ringing from the previous strike. Also, by allowing it to run as an application, receiving the constant stream of wind data mentioned above, it would be possible to implement the model as a real time model that would adjust its output constantly based on the current and past wind speeds rather than generating a number of sound files for fixed wind speeds. This would require much more time to be allowed for programming. In the case of a real time model, a wind speed sensor that can be polled at a higher rate and can be read directly from the software rather than going through a third party application such as WeatherSnoop would also be necessary for such an improvement to be worthwhile and practical.

Sensor:

If one did wish to use different hardware components, or if small-scale home-use weather stations prove to be inadequate, several other options exist. In terms of limitations regarding polling rate, one would simply have to acquire or modify a weather station with a higher polling rate than the Ambient Weather WS-2095 and WeatherSnoop 3 allow. Such a station could be developed by building a sensor from scratch designed with the capability of providing real time data, or by designing a hardware piece that could directly interpret the analog data coming from a given sensor. A simpler, but possibly more expensive method would be to buy a more sophisticated weather instrument, such as those found on sailboats or weather sensors used by commercial media outlets and government agencies for weather prediction. Some of which provide accurate wind speed and direction data in real time. These sensors are more expensive than the type of sensor used in the project, as they tend to be more accurate and are used in only a niche market. Using one of these higher end sensors the installation could have a real time stream of data for use in any program. Should acquiring or constructing an adequate sensor be an issue, it could be possible for a system to obtain its weather data from a sensor owned by a third party, whose data could be streamed over the internet.

Audio Devices:

In a real world installation similar to our project, it will be necessary to improve upon the audio system we are already using. Specifically, it will be necessary to develop a plan for how sounds will be distributed across the forty channels and to develop a system capable of processing and utilizing forty different audio streams. Our current setup currently outputs a single channel that changes based on wind direction. In a more advanced installation, one could utilize the multi-channel system by directing a single audio signal along a single stream. Which stream of the forty that would be utilized could be determined by the wind direction. To fully utilize such a system and have all streams being utilized at once would require a much more complicated programming setup. This could consist of a virtual wind chime with a speaker assigned to each chime, or could consist of forty or more unique recordings for each speaker to play simultaneously. The downsides of either plan being: the need to develop a complicated program that can simulate the random nature of a wind chime, and the need to record a vast number of sounds to be played, respectively. In addition, both potential setups must overcome a common challenge. Currently, few if any amplifiers exist that can process forty different audio channels at once- most home theater setups can only process six to eight channels. A complicated, but feasible, solution would be to have five or more instances of the program running in parallel on five separate computer-amplifier installations. By differentiating each of the programs slightly, one could have one installation be prompted to play a tone while another remains silent without there being any communication between the two. To ensure that each installation is properly synchronized, it will likely be necessary for each installation to receive weather data from the same source and to build regular pauses into each program during which each installation would wait for an "OK" signal from each of its four counterparts before continuing. Doing so could help prevent programs from diverging throughout the course of their operation.

Locations, Uses, and Other Considerations:

Finally, and perhaps most importantly, any real world installation similar to our setup will require careful thought and planning to ensure that it is useful and sustainable. Key considerations would likely follow in this order: Firstly, will the installation have the desired effect of improving a subject's state of mind, or will it be obtrusive or annoying? Careful consideration must be given to the volume, pitch, and composition of the sounds being played to ensure that they are pleasant to listen to and do not interfere with other activities. Second, will the installation be installed in a location where it will see a significant amount of use? Ideally, an installation such as this is best suited to a large indoor public space with high traffic, such as a transit station or office building lobby. Places such as these would make for a unique and easily accessible public attraction and can instill a greater degree of calm in a stressful and busy environment. Third and finally, will the installation be feasible to fund, construct, and maintain throughout its lifetime? Developing a very advanced system with a high degree of variability and precision can produce greater variability and precision, but cost and maintenance constraints must be kept in mind. A large, expensive, and temperamental system will be less likely to be funded by public or private contributors and may tax resources such as space and manpower more heavily than those that host the installation can afford, even if the project is fully funded. If one encounters difficulties, it could be possible to modify the system to serve other functions normally associated with audio installations. It could be relatively simple to make the systems speakers and amplifiers available as an extension of a PA system or as a permanent surround system available for use in special events hosted in the immediate area. If the area designated for this installation requires the use of a large audio system, these added capabilities may help convince entities to adopt it as both a public attraction and useful tool.

Conclusion:

The future of this project is a promising one. With our contributions to technology, methodology, and planning of this project, it is closer than ever to being realized. While a real world installation is not yet in place, we do not anticipate that this will remain the case for much longer. The importance of greenspace in the world today is ever growing, as urban areas themselves grow. We must be prepared to live in a world where opportunities to connect with the environment are less frequent, and harder to realize. This project, and the projects that evolve from it will help to make that future more tolerable and will help advance the cause of organic architecture well into the future. A future in which it will be so desperately needed.

Bibliography:

Ambient LLC. "Ambient Weather WS-2095 Wireless Home Weather Station User Manual." Vers. 1.1. 2015. *ambientweatherstore.com.* PDF. <http://site.ambientweatherstore.com/Manuals/ws2095.pdf>.

Baker, Carol F., Bonnie J. Garvin, Carol W. Kennedy, and Barbara J. Polivka. "The Effect of Environmental Sound and Communication on CCU Patients' Heart Rate and Blood Pressure." Res. Nurs. Health Research in Nursing & Health: 415-21. Web. 10 June 2015.

Bianchi, Frederick. "Sound Design: Green-linking Exterior and Interior Themes." *The International Journal of Environmental, Cultural, Economic and Social Sustainability* n.d.: 41-44. PDF. <http://frederickbianchi.cgpublisher.com/product/pub.41/prod.448>.

Browning, William, Joseph Clancy and Catherine Ryan. *14 Patterns of Biophilic Design*. Terrapin Bright Green. New York, 2014. Web. <http://www.terrapinbrightgreen.com/reports/14-patterns/#content>.

DeLongis, Anita, Susan Folkman and Richard S. Lazarus. "The Impact of Daily Stress on Health and Mood: Psychological and Social Resources as Mediators." *Journal of Personality and Social Psychology* 54.3 (1988): 486-495. Web.

Dzhambov, Angelmario, and Donkadimitrova Dimitrova. "Urban Green Spaces′ Effectiveness as a Psychological Buffer for the Negative Health Impact of Noise Pollution: A Systematic Review. "Noise Health Noise and Health: 157. Web. 10 June 2015.

Kaplan, Stephen. "The Restorative Benefits of Nature: Toward an Integrative Framework." *Journal of Environmental Psychology* 15 (1995): 169-182. Web.

Lukkari, Teemu and Vesa Välimäki. "Modal Synthesis of Wind Chime Sounds with Stochastic Event Triggering." *Proceedings of the 6th Nordic Signal Processing Synopsium -*

NORSIG 2004. Espoo, Finland: Helsinki University of Technology, 2004. 212-215. Web. <http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=1344561>.

Ngowi, Rodrique. "Artist Displays Vision, Sounds of New England at Boston Airport." *The Boston Globe*. The Associated Press. 21 April 2007. Web. <http://www.boston.com/news/education/higher/articles/2007/04/21/artist_displays_vision_sounds_of_new_england_at_bostons_airport/>.

Obniski, Monica. "The Arts and Crafts Movement in America." *Heilbrunn Timeline of Art History*. The Metropolitan Museum of Art. New York, 2000. Web. <http://www.metmuseum.org/toah/hd/acam/hd_acam.htm>.

Appendix:

Figure 1: Simple Harmonic Motion #5

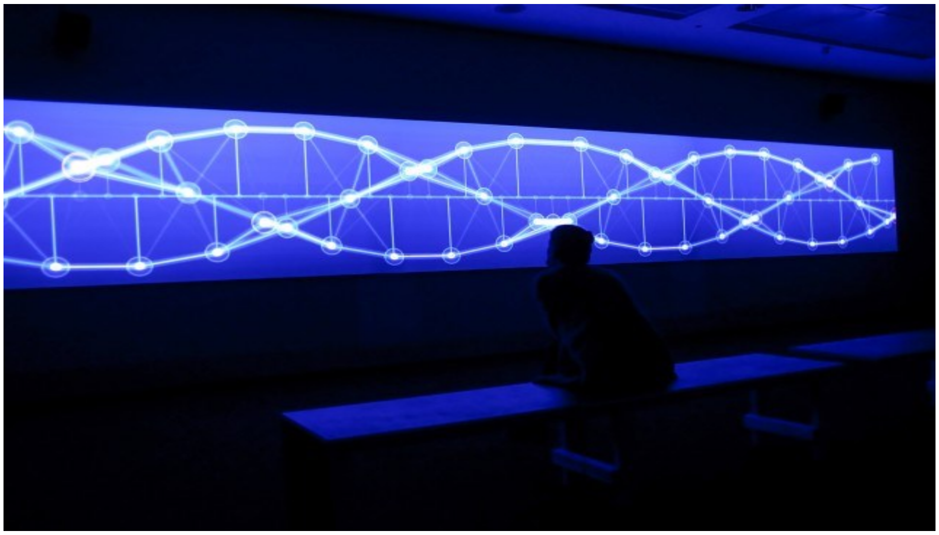


Figure 2: Ambient Weather Sensor

Figure 3: Ambient Weather Receiver

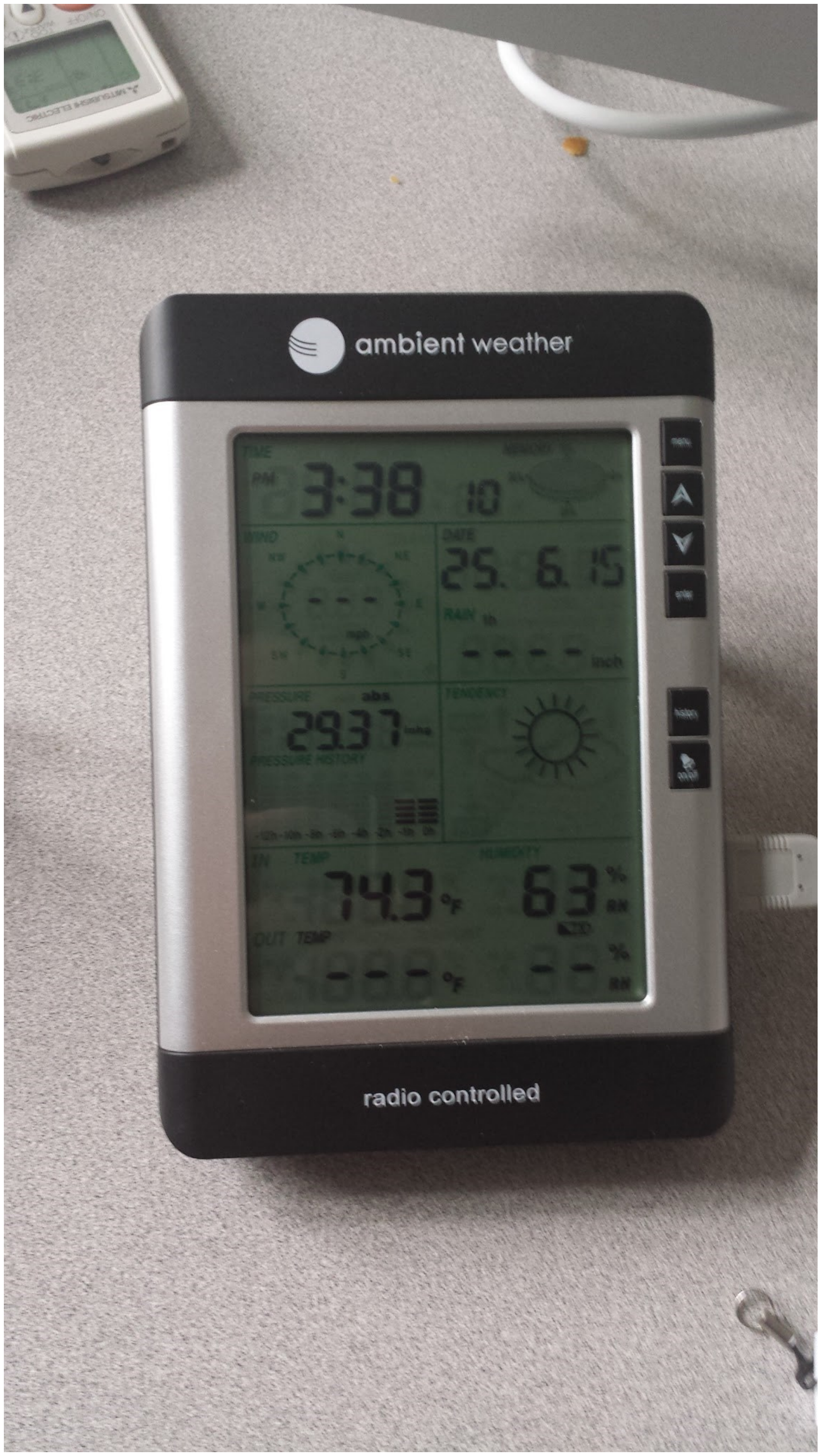


Figure 4: One Minute Program Code

```
tell application "WeatherSnoop 3" --calls WeatherSnoop 3 program
	set file_path to (null) --ensures filepath is empty at start
	tell weather property "Wind Speed" of agent of first document --pulls value of windspeed
		if value < "2" then --tests for value less then 2
			set file_path to ("/Users/lab/Desktop/RocksIQP/Music/North.mp3") --sets filepath to the file labeled north
		else if value < "3" then --less then 3
			set file_path to ("/Users/lab/Desktop/RocksIQP/Music/Northeast.mp3")
		else if value < "4" then --less then 4
			set file_path to ("/Users/lab/Desktop/RocksIQP/Music/east.mp3")
		else if value < "5" then --less then 5
			set file_path to ("/Users/lab/Desktop/RocksIQP/Music/southeast.mp3")
		else if value < "6" then --Less then 6
			set file_path to ("/Users/lab/Desktop/RocksIQP/Music/south.mp3")
		else if value < "7" then --less then 7
			set file_path to ("/Users/lab/Desktop/RocksIQP/Music/southwest.mp3")
		else if value < "8" then --less then 8
			set file_path to ("/Users/lab/Desktop/RocksIQP/Music/west.mp3")
		else if value > "8" then --greater then 8
			set file_path to ("/Users/lab/Desktop/RocksIQP/Music/Northwest.mp3")
		end if
		tell application "QuickTime Player" to play (open file_path) --tells QuickTime Player to open selected file
		delay 60 --one minute delay
		tell application "QuickTime Player" to quit --closes QuickTimePlayer ending the file
	end tell
end tell
```

Figure 5: File Selector Program Code

AppleScript <No selected element>

```
tell application "WeatherSnoop 3" --calls WeatherSnoop 3 program
	set file_path to (null) --ensures filepath is empty at start
	tell weather property "Wind Direction" of agent of first document --pulls value of windspeed
		if value = "0" or value = 360 then --tests for value equalling north
			set file_path to ("/Users/lab/Desktop/RocksIQP/Music/North.mp3") --sets filepath to the file labeled north
		else if value = "45" then --north east
			set file_path to ("/Users/lab/Desktop/RocksIQP/Music/Northeast.mp3")
		else if value = "90" then --east
			set file_path to ("/Users/lab/Desktop/RocksIQP/Music/east.mp3")
		else if value = "135" then --southeast
			set file_path to ("/Users/lab/Desktop/RocksIQP/Music/southeast.mp3")
		else if value = "180" then --south
			set file_path to ("/Users/lab/Desktop/RocksIQP/Music/south.mp3")
		else if value = "225" then --southwest
			set file_path to ("/Users/lab/Desktop/RocksIQP/Music/southwest.mp3")
		else if value = "270" then --West
			set file_path to ("/Users/lab/Desktop/RocksIQP/Music/west.mp3")
		else if value = "315" then --northwest
			set file_path to ("/Users/lab/Desktop/RocksIQP/Music/Northwest.mp3")
		end if
		
		tell application "QuickTime Player" to play (open file_path) --tells QuickTime Player to open selected file
		delay 60 --one minute delay
		tell application "QuickTime Player" to quit --closes QuickTimePlayer ending the file
		
	end tell
end tell
```

Figure 6: 5 Minute Program Code

```
tell application "WeatherSnoop 3" to open "/Users/lab/Desktop/RocksIQP/IQP.wssite" --opens weathersnoop file for our site
tell application "System Events" to tell process "WeatherSnoop 3" to set visible to false --hides weathersnoop program for
    cleaner look
tell application "WeatherSnoop 3" --calls the WeatherSnoop 3 application
    launch file "/Users/lab/Desktop/RocksIQP/starter.scpt"
    run script "/Users/lab/Desktop/RocksIQP/starter.scpt" --ensures that weathersnoop is recording data
    tell weather property "Wind speed" of agent of first document --pulls value of windspeed
        get value
        log value --records and logs value for testing purposes
        repeat 5 times --contents within will repeat x times
            tell application "AppleScript Editor"
                launch file "/Users/lab/Desktop/RocksIQP/windAccessFile.scpt" --launches access file
                run script "/Users/lab/Desktop/RocksIQP/windAccessFile.scpt" --runs access group
            end tell
        end repeat
        if running then --stops recording data
            stop
        end if
    end tell
    tell application "WeatherSnoop 3" to close --closes weathersnoop at the end of run
end tell
```

Figure 7: WeatherSnoop Check Program Code

```
tell application "WeatherSnoop 3" --starts recording data if not already recording
    tell agent of first document
        if running then
        else
            start
        end if
    end tell
end tell
```

Figure 8: Code for Advanced Synthesis Model

```
clear all;
fs = 44100;
L = 60;   %length of the audio files to be generated in seconds

samples = fs*L;


W = 5;   %wind speed
c = 0.1;   %constant to adjust wind speed to corresponding energy
E = W*c;
p = 1/(1+(99*exp(-2*E)));

state1 = 0;
nexthit = 1;

[chime1, fs] = audioread('chime1.wav'); %chimex.wav should point to the corresponding chime file
[chime2, fs] = audioread('chime2.wav');
[chime3, fs] = audioread('chime3.wav');
[chime4, fs] = audioread('chime4.wav');
[chime5, fs] = audioread('chime5.wav');

output = zeros(samples,1);

n1 = 0;
n2 = 0;
n3 = 0;
n4 = 0;
n5 = 0;

hit = 0;

        for n = 1:samples
        if n == nexthit
        nexthit = n + 1323 + randi(882);
        roll = random('unif', 0, 1);
        if state1 == 0
        if roll < p
                if (roll/p) < 1/5
                state1 = 1;

                elseif (roll/p) < 2/5
                state1 = 2;

                elseif (roll/p) < 3/5
                state1 = 3;
```

```
        elseif (roll/p) < 4/5
        state1 = 4;

        else
        state1 = 5;
        end
end
end

if state1 == 1
n1 = 1;
elseif state1 == 2
n2 = 1;

elseif state1 == 3
n3 = 1;
elseif state1 == 4
n4 = 1;
elseif state1 == 5
n5 = 1;
end
end
if n1 > 0
output(n) = output(n) + chime1(n1);
if n1 == length(chime1)
n1 = 0;
else
n1 = n1 + 1;
end

end
if n2 > 0
output(n) = output(n) + chime2(n2);
if n2 == length(chime2)
n2 = 0;
else
n2 = n2 + 1;
end

end
if n3 > 0
output(n) = output(n) + chime3(n3);
if n3 == length(chime3)
n3 = 0;
else
n3 = n3 + 1;
end
```

```
end
if n4 > 0
output(n) = output(n) + chime4(n4);
if n4 == length(chime4)
n4 = 0;
else
n4 = n4 + 1;
end

end
if n5 > 0
output(n) = output(n) + chime5(n5);
if n5 == length(chime5)
n5 = 0;
else
n5 = n5 + 1;
end

end

if state1 == 1
if roll < p
        if (roll/p) < 1/2
        state1 = 5;
        else
        state1 = 2;
        end
else
        state1 = 0;
end
end
if state1 == 2
if roll < p
        if (roll/p) < 1/2
        state1 = 1;
        else
        state1 = 3;
        end
else
        state1 = 0;
end
end
if state1 == 3
if roll < p
        if (roll/p) < 1/2
        state1 = 4;
        else
        state1 = 2;
        end
```

```
	else
		state1 = 0;
	end
	end
	if state1 == 4
	if roll < p
		if (roll/p) < 1/2
		state1 = 5;
		else
		state1 = 3;
		end
	else
		state1 = 0;
	end
	end
	if state1 == 5
	if roll < p
		if (roll/p) < 1/2
		state1 = 1;
		else
		state1 = 4;
		end
	else
		state1 = 0;
	end
	end

	end


audiowrite('output.wav', output, fs);
```