Contents of Portfolio

by Tianrui Xiong

Max/MSP Patch & SuperCollder Synth: *The Silk Road Fantasia*

The Silk Road Fantasia is a computer music work consisting of a Max/MSP patch and a SuperCollider synth, presented in two parts: *The Silk Road Fantasia: Journey* and *The Silk Road Fantasia: Desert.*

The Silk Road Fantasia: Journey is an improvised interactive electronic music piece that combines vocals, Zhongruan (a traditional Chinese folk instrument), and real-time interaction with Max/MSP. It uses a unique blend of Western Chinese vocal techniques, conventional Chinese instrumentation, and modern music technology to narratively depict the journey of Zhang Qian, the ancient Chinese explorer who opened up the Silk Road.

The Silk Road Fantasia: Desert is a SuperCollider live coding work. It primarily uses atmospheric synthesizers defined by the "SynthDef" module to evoke the vastness and mystique of the Silk Road desert landscape.

These two pieces—created with Max/MSP and SuperCollider—offer a fresh interpretation of the exploration of the ancient Silk Road, merging traditional storytelling with the power of modern audio-programming environments.

Max/MSP Patch: *The Silk Road Fantasia: Journey*

-for Vocal, Zhongruan, and Max/MSP

YouTube Link: *https://youtu.be/9rqJU8SELoo?si=aK_6rgneej6O2GOi*

The Silk Road Fantasia: Journey is a piece of improvised interactive electronic music. It is inspired by the story of Zhang Qian's expedition to the Western Regions during the Han Dynasty, which led to the foundation of the Silk Road. It combines vocals and the traditional Chinese instrument, the Zhongruan, with real-time interaction through Max/MSP to create an improvised interactive electronic music performance. Within Max, I designed various interactive effects (Fig. 1) to reflect the challenges of founding the Silk Road. The Zhongruan, an instrument I used in the performance, has great historical importance as it is the predecessor of the Han Pipa and played a significant role in musical and cultural exchanges along the Silk Road.

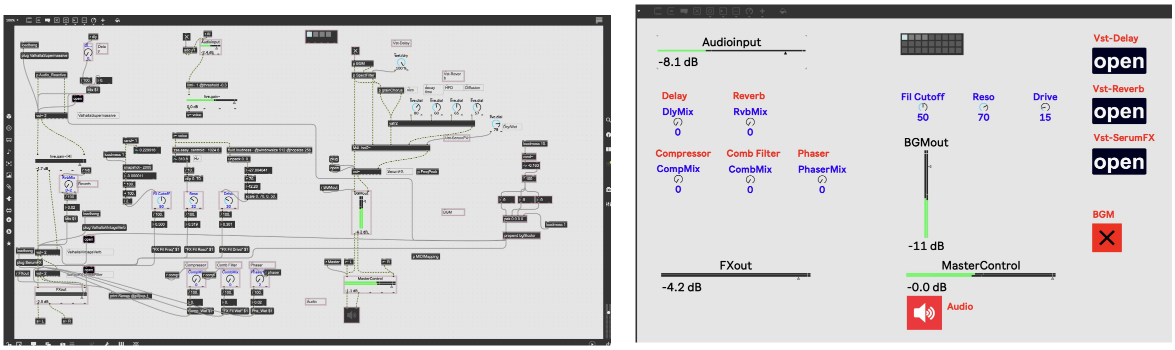


Fig. 1. Original patch & presentation mode

1. Technology Documentation:

I primarily designed two sounds in Max: a background sound and a jumping sound, both controlled by Zhongruan and vocals.

Background Sound: I connected a granular FM soundscape pad to a spectral filter. In this spectral filter, the spectral data comes from “zsa.easy_bark~” object (Fig. 2) that analyzes the signals from the microphone in real time, and “zsa.easy_bark~” object can analyze the bark coefficients in the input signal and feed them into the spectral filter to generate a drone sound controlled by the vocals and Zhongruan, symbolizing the strong winds of the Silk Road’s vast deserts.

Jumping Sound: Based on “FluCoMa’s Audio-Reactive” patch, I created a synthesizer that changes according to the microphone input in real time. I also designed a pair of thresholds for controlling “fluid.onsetslice~” object binaurally and used the “fluid.loudness~” object to analyze the true peak of my voice in real time. Through various mathematical operations (absolute values, addition, subtraction, multiplication, division), I made it so that the stronger my input signal, the lower the threshold would be, resulting in a denser Max-generated sound (Fig. 3). Within this effect, I also added objects to the right channel that randomly adjust the threshold within a specific range, aiming to bring more random and experimental effects to the performance. These sound effects also symbolize the grains of sand blowing across the deserts along the Silk Road.

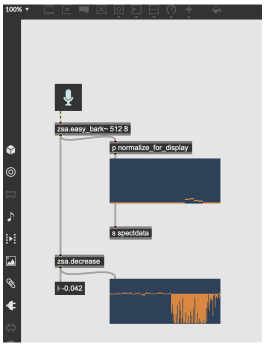


Fig. 2

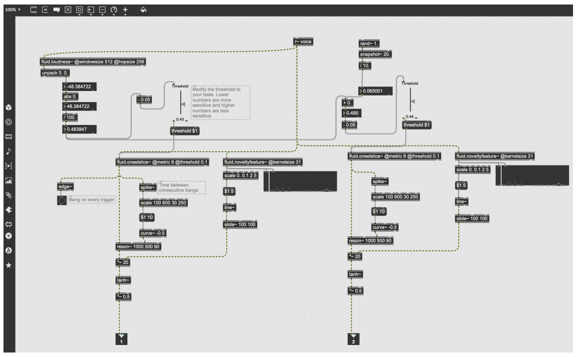


Fig. 3

Building upon the two primary sounds mentioned above, I also incorporated effects such as "Valhalla Supermassive", "Valhalla Vintage Verb", and the comb filter and Multiband Compressor from SerumFX to add more complexity and variety to the jumping sound effect in Max (Fig. 4). These effects were mapped to the AKAI Midimix controller (Fig. 5), allowing for real-time adjustments during the performance. Through these dynamic changes, I aimed to enhance the piece's complexity, tension, and emotional impact.

Fig. 4

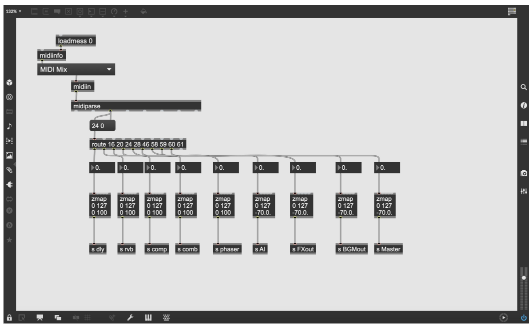


Fig. 5

I used a Neumann TLM170 large diaphragm condenser microphone to capture the sounds of the Zhongruan and vocals, which are fed into Max for interaction, and a pair of Neumann KM184 small diaphragm condenser microphones for live recording.

The piece is 5 minutes and 30 seconds long, and it consists of an introduction, two contrasting parts, and a coda.

2. Music Documentation:

Intro (00:00-00:48):

The piece starts with vocals singing three words from ancient Chinese poetry depicting the Silk Road: “Tian” (heaven), “cang” (vast), and “ye” (wilderness) from the phrase tian cang cang, ye mang mang (the sky is vast and boundless, the wilderness stretches endlessly). In real-time, the Midimix controller gradually introduces the background drone sound from Max. As the plucking of the Zhongruan intensifies, it brings out the jumping sound effect in Max, illustrating the early stages of the expedition into the desert along the Silk Road.

Part I (00:48-02:18):

After the introduction, a background sound of bells (Fig. 6), symbolizing camel bells along the Silk Road is played in Ableton Live and continues until the end of the work. The vocals shift to fragments of the Uyghur greeting ياخشىمۇسىز (yaheshimosezi), paired with the Zhongruan to propel the music forward. A higher dry/wet ratio of the reverb effect controlled by MidiMix enhances the sense of space, followed by tremolo and glissando techniques on the Zhongruan accompanying the increasing mix of compression effects from SerumFX to evoke the biting desert winds. This section builds towards the first climax, concluding with plucked notes from the Zhongruan, synchronized with the interactive delay sound from Max.

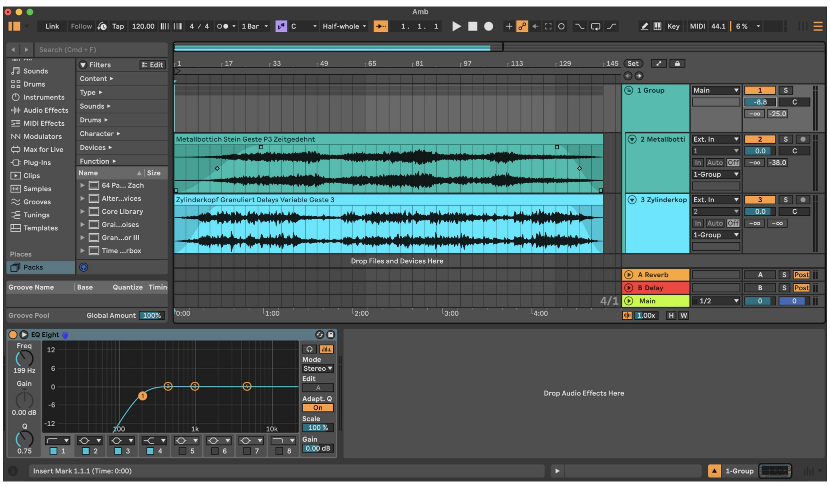


Fig. 6

Part II (02:18-04:46):

The second part builds upon the first, beginning with the harmonic tones of the Zhongruan. It continues with tremolo and glissando techniques, accompanied by vocals interacting with Max, and introduces a comb filter effect with resonance, still controlled by MidiMix, intensifying the sound and adding a metallic texture. These sound effects symbolize the deepening journey along the Silk Road, where the challenges grow more and more severe. After reaching a climax at 03:29, the comb filter and the compressor effect continue to build, amplifying the emotional intensity of the music. The section concludes with vocals mimicking intense wind sounds and granular vocal particles.

Coda (04:46-05:30):

The coda features vocals and increasingly fast strikes across the four strings of the Zhongruan up and down, paired with the gradual rise of the compressor effect's mix, pushing the performance to the last peak. The piece concludes with the plucking of the Zhongruan, enriched with Zhongruan's delay sounds, symbolizing the arduous journey along the Silk Road, finally reaching its destination.

3. Summary:

In summary, *The Silk Road Fantasia: Journey* is set against the backdrop of the Silk Road expedition, using vocals and the Silk Road's representative instrument, the Zhongruan, in real-time interaction with a designed Max coding system. This fusion of traditional performance with interactive electronic sound manipulation adds greater tension and dynamism. The intelligent combination of vocals, instruments, and electronic music highlights the hardships of the Silk Road journey, imbuing the piece with a unique style.

SuperCollider Synth: *The Silk Road Fantasia: Desert*

YouTube Link: https://youtu.be/4WqupbVmJCo?si=bxbe10B1WK5EDo6Q

The Silk Road Fantasia: Desert is a piece of live coding music. It is a derivative work developed from *The Silk Road Fantasia: Journey,* using SuperCollider for live coding. It primarily uses atmospheric synthesizers defined by the “SynthDef” module to evoke the vastness and mystique of the Silk Road desert landscape (Fig. 1).

```


s.boot;
s.plotTree;
s.meter;


~filePath = Platform.userHomeDir +/+ "Desktop/recorded_audio.wav";

    ~filePath.postln;

    s.prepareForRecord(~filePath, 2);

    s.record;

    (5).seconds.wait;
        s.stopRecording;
        "Recording stopped. File saved to Desktop.".postln;


(
SynthDef(\dpad,{
 var asd;
 asd = BPF.ar(Hasher.ar(Sweep.ar), 300, 0.1) * 10.dbamp;
 asd = asd * Env.perc(4, 7).ar(Done.freeSelf);
 asd = asd ! 2;
 asd = asd *\amp.kr(-30.dbamp);
 Out.ar(\out.kr(2), asd);
}).add;
)


Pdef(\dpad, Pbind(\instrument, \dpad,\degree, Pbrown(100, 1, 0.3,inf), \dur, 6 ));

x = Pdef(\dpad);

Pdef(\dpad).gui;

x. play;


(
SynthDef(\Pad, {
 var pad;
 pad = Saw.ar(\freq.kr(440) * ({ LFNoise2.kr(8) } ! 16 * 0.1).midiratio * [1, 2, 1, 2, 1, 2, 1, 2, 2, 1, 2, 1, 2, 1, 2, 1]);
 pad = DelayC.ar(pad, 0.01, { Rand(0, 0.01) } ! pad.size);
 pad = Splay.ar(pad);
 pad = LPF.ar(pad, \freq.kr(440) * 4);
 pad = BPeakEQ.ar(pad, \freq.kr(440) * 8, 0.1, 8);
 pad = pad * Env.asr(1, 1, 8).ar(Done.freeSelf, \gate.kr(1));
 pad = pad * \amp.kr(-100.dbamp);
 pad = pad ! 2;
 Out.ar(\out.kr(0), pad);
}).add;
)

(
~wrr = Scale.new([5, 7, 11, 13, 2, 3, 5, 7], 24, tuning:\et24, name:\wrr);
~www = Scale.new([4, 6, 8, 9, 10, 12, 10, 9, 8], 24, tuning:\et24, name:\www);
)

Ndef(\Pad).play;
Ndef(\Pad).free;
Ndef(\Pad).set

Ndef(\Pad, Pbind(\instrument, \Pad,\scale, ~www, \degree, [0,4,6],\dur, 0.4, \att, 0,\octave, Pseq([1, 2, 1, 2],inf)))

Ndef(\Pad, Pbind(\instrument, \Pad,\scale, ~www, \degree, [0,4,6],\dur, 0.4, \att, 0,\octave, Pseq([1, 2, 4],inf)))

Ndef(\Pad, Pbind(\instrument, \Pad,\scale, ~wrr, \degree, [0,4,6],\dur, 0.4, \att, 0,\octave, Pseq([8,5,3,4,7,6,2],inf)))

Ndef(\Pad, Pbind(\instrument, \Pad,\scale, ~wr, \degree, [0,4,6],\dur, 0.4, \att, 0,\octave, Pseq([8,5,3,4,7,6,2,8],inf)))

```

Fig. 1

1. Technology & Music Documentation:

In *The Silk Road Fantasia: Desert*, I used SuperCollider live coding techniques to create a soundscape that evokes the desolation and profound atmosphere of the desert. The entire piece lasts for 3 minutes and 56 seconds and utilizes two distinct sound elements—desert winds and an organ-like background—to illustrate a desert filled with life and mystery through their continuous changes and progression.

1.1 Desert Winds:

The desert winds are produced through the "dpad" synthesizer, which employs components such as "Sweep", "Hasher", and a bandpass filter, "BPF" (Fig. 2). The use of "Env.perc(4, 7)" plays a key role in ensuring the wind sound transitions naturally during live coding. With an attack of 4 seconds and a release of 7 seconds, the gradual fade-in and fade-out of the sound avoids abrupt changes, creating a natural decay and flow. This results in a wind effect that shifts smoothly, evoking a sense of movement and space, and effectively capturing the essence of desert winds.

```
(
SynthDef(\dpad,{
 var asd;
 asd = BPF.ar(Hasher.ar(Sweep.ar), 500, 0.1) * 10.dbamp;
 asd = asd * Env.perc(4, 7).ar(Done.freeSelf);
 asd = asd ! 2;
 asd = asd *\amp.kr(-30.dbamp);
 Out.ar(\out.kr(2), asd);
}).add;
)
```

Fig. 2

```
Pdef(\dpad, Pbind(\instrument, \dpad,\degree, Pbrown(100, 1, 0.3,inf), \dur, 6 ));
```

Fig. 3

To achieve sustained wind sounds, I used a combination of "\dur" and "inf" in "Pbind" and "Pbrown" respectively, allowing the wind sound to continuously hover and loop throughout the piece, adding to the immersive experience and a sense of continuity (Fig. 3). During the live coding, I adjusted the frequency parameter of the "BPF" filter in real time, the parameter was alternated between high and low values to mimic the effect of the wind gently across the desert. This creates a natural wind sound that seems to change direction and intensity smoothly, giving a strong sense of space and movement—capturing the shape of desert winds.

1.2. Organ-Like Background:

The organ-like background was generated using the “Pad” synthesizer. This synthesizer starts with a “Saw” waveform, modulated with “LFNoise2” (low-frequency noise) to add a slightly wavering and unpredictable quality, much like the shimmering mirages in the desert. I also used “Env.asr(1, 1, 8)” to ensure the wind sound transitions naturally during live coding. Additionally, effects like “DelayC”, “Splay” (stereo spread), “LPF” (low-pass filtering), and peaking “BPeakEQ” (EQ) were applied to create a wide and rich atmosphere (Fig. 4).

```
(
SynthDef(\Pad, {
 var pad;
 pad = Saw.ar(\freq.kr(440) * ({ LFNoise2.kr(8) } ! 16 * 0.1).midiratio * [1, 2, 1, 2, 1, 2, 1, 2, 2, 1, 2, 1, 2, 1, 2, 1]);
 pad = DelayC.ar(pad, 0.01, { Rand(0, 0.01) } ! pad.size);
 pad = Splay.ar(pad);
 pad = LPF.ar(pad, \freq.kr(440) * 4);
 pad = BPeakEQ.ar(pad, \freq.kr(440) * 8, 0.1, 8);
 pad = pad * Env.asr(1, 1, 8).ar(Done.freeSelf, \gate.kr(1));
 pad = pad * \amp.kr(-100.dbamp);
 pad = pad ! 2;
 Out.ar(\out.kr(0), pad);
}).add;
)
```

Fig. 4

During the live coding, I used “Ndef” to dynamically adjust parameters and switched between different “Ndef” configurations in real time, giving the organ sound a flowing and organic quality (Fig. 5). These dynamic adjustments allowed the background to shift continuously, adding changes to the chords and painting a sonic picture of the vast desert, filled with untold stories and legends.

```

(
~wrr = Scale.new([5, 7, 11, 13, 2, 3, 5, 7], 24, tuning:\et24, name:\wrr);
~www = Scale.new([4, 6, 8, 9, 10, 12, 10, 9, 8], 24, tuning:\et24, name:\www);
)


Ndef(\Pad).play;
Ndef(\Pad).free;
Ndef(\Pad).set

Ndef(\Pad, Pbind(\instrument, \Pad,\scale, ~www, \degree, [0,4,6],\dur, 0.4, \attt, 0,\octave, Pseq([1, 2, 1, 1],inf)))

Ndef(\Pad, Pbind(\instrument, \Pad,\scale, ~www, \degree, [0,4,6],\dur, 0.4, \attt, 0,\octave, Pseq([1, 2, 5],inf)))

Ndef(\Pad, Pbind(\instrument, \Pad,\scale, ~wrr, \degree, [0,4,6],\dur, 0.4, \attt, 0,\octave, Pseq([8,5,3,4,7,6,2],inf)))

Ndef(\Pad, Pbind(\instrument, \Pad,\scale, ~wr, \degree, [0,4,6],\dur, 0.4, \attt, 0,\octave, Pseq([8,5,3,4,7,6,2,8],inf)))

```

Fig. 5

Together, these sounds achieved a constantly evolving atmosphere through real-time parameter adjustments. Particularly with the organ background, the adaptive changes during live performance allowed it to seamlessly fit the musical flow, creating an ethereal and profound musical experience.

1.3. Natural Ending Transition:

At the midpoint of the performance (01: 48), I began altering the value of 0.1 in “pad = Saw.ar(\freq.kr(440) * ({ LFNoise2.kr(8) } ! 16 * 0.1))”. I gradually increased it to modify the influence of “LFNoise2” on the base frequency modulation. As the performance progressed, the sound became increasingly unstable, conveying a sense of growing unpredictability. Towards the end, I slowly decreased the value again, allowing the pad's sound to transition back to a stable state. This dynamic progression—from stability to intense instability and then returning to tranquillity—served to symbolize the mystery and danger of the desert along the Silk Road.

In the final part of the piece, both the wind and organ sounds faded out naturally. As shown in the area highlighted by the red box in Fig. 6, the organ background sound faded out by running a line of code with an undefined global variable, “~wr”, which could not make sound, allowing the sound to decay naturally. The wind sound gradually disappeared by using the “paus” (pause) function in “Pdef (\dpad).gui”, much like the wind slowly dying down, returning the desert to silence, like the last sigh of the desert, leaving the listeners with a lingering sense of wonder.

```
(
~wrr = Scale.new([5, 7, 11, 13, 2, 3, 5, 7], 24, tuning:\et24, name:\wrr);
~www = Scale.new([4, 6, 8, 9, 10, 12, 10, 9, 8], 24, tuning:\et24, name:\www);
)


Ndef(\Pad).play;
Ndef(\Pad).free;
Ndef(\Pad).set

Ndef(\Pad, Pbind(\instrument, \Pad,\scale, ~www, \degree, [0,4,6],\dur, 0.4, \attt, 0,\octave, Pseq([1, 2, 1, 1],inf)))

Ndef(\Pad, Pbind(\instrument, \Pad,\scale, ~www, \degree, [0,4,6],\dur, 0.4, \attt, 0,\octave, Pseq([1, 2, 5],inf)))

Ndef(\Pad, Pbind(\instrument, \Pad,\scale, ~wrr, \degree, [0,4,6],\dur, 0.4, \attt, 0,\octave, Pseq([8,5,3,4,7,6,2],inf)))

Ndef(\Pad, Pbind(\instrument, \Pad,\scale, ~wr, \degree, [0,4,6],\dur, 0.4, \attt, 0,\octave, Pseq([8,5,3,4,7,6,2,8],inf)))
```

Fig. 6

2. Summary:

The Silk Road Fantasia: Desert is a derivative piece from *The Silk Road Fantasia: Journey*. It is crafted with SuperCollider live coding and features two distinct sounds: desert winds and an organ-like background sound. The winds evolve smoothly, capturing the feeling of the breeze across dunes, while the organ sound brings a layered, dreamy quality to the atmosphere. Both sounds are adjusted in real time, giving the music a fluid, organic character that mirrors the expansive and mysterious nature of the desert. The piece gently fades away, like the wind dying down and the desert returning to stillness.

Max Patch: *Hello World!*

-for Facial Movements and Max/MSP

YouTube Link: https://youtu.be/be65zQkX1t4?si=zVCl6eHdJckuG-1d

Hello World! is a Max/MSP piece of improvised interactive electronic music that responds to facial movements. Named after the classic “Hello, World!” program in computer programming, this Max/MSP piece explores interactive music using facial movements. By leveraging FaceOSC (Fig. 1), my facial movements, including positions of features and jaw, are converted into digital signals, which are then transmitted to Max for real-time interaction (Fig. 2).

The interactive music piece uses a performance filled with tension and gradually building emotion to mimic and express a newborn’s first arrival to the world.

Fig. 1

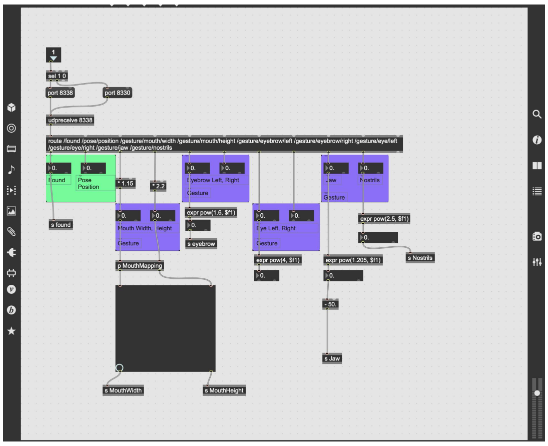


Fig. 2

1. Technology Documentation:

I designed two types of sounds in Max: a looped sound and a background drone sound (Fig. 3). These sounds are processed mainly using the “groove~” sample playback object, which manipulates loops of the audio samples stored in “buffer~” objects. The sounds are controlled by signals from FaceOSC, which adjust the start and end times of the audio loops processed by the “groove~” object.

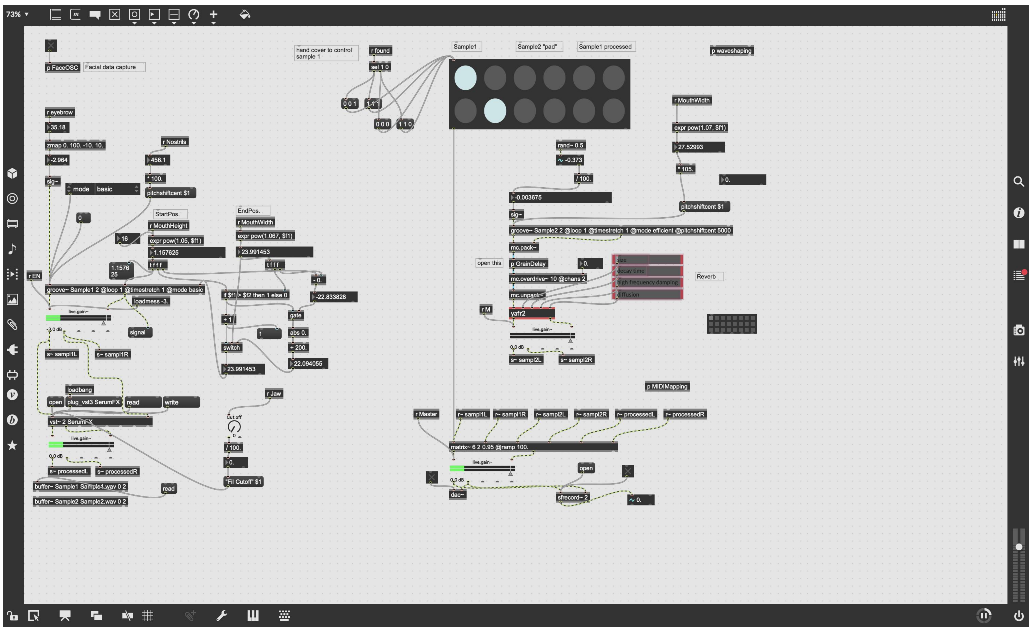


Fig. 3

1.1 Looped Sound:

As shown in the left part of Fig. 3, I designed an algorithmic system to create tension within the looped sound. First, FaceOSC generates signals based on my mouth height and width, which are projected onto a “pictslider” object to make sure that the numerical signals are not too large. The signals are limited to 0-127 (Fig. 2).

Next, I connected the mouth height and width signals to an “expr” object to perform an exponential function calculation. The base values of the functions are both greater than 1, while the exponents are dynamic values generated by the mouth’s height and width. Smaller values change less, whereas larger values change more, thus enhancing the tension effect in the real-time interactive music.

Then, I connected the mouth height and width signals to the Loop Min and Loop Max inputs of the “groove~” object respectively to control the looped sound's start and end times in real time. Since the mouth width is easier to maintain, it is connected to Loop Max, with the base for the exponential function set to 1.067. This results in a maximum output of approximately 3774, meaning that the 4-second audio stored in the “buffer” can have a loop endpoint up to around 3.7 seconds. For the mouth height (connected to Loop Min), I needed to reduce the time value's dynamic range, so I set the base of its exponential function to 1.05, making the maximum output approximately 490, which corresponds to starting the loop at most 0.49 seconds.

To prevent audio discontinuities when the Loop Min value (controlled by mouth height) exceeds Loop Max (controlled by mouth width), I designed an algorithmic system centered around the "if", "gate", and "switch" objects to analyze and process the values produced by the mouth width, thereby ensuring the end time not to exceed the start time in the looped audio:

"if" object: Compares values. When the height value exceeds the width value, it outputs a signal of 1; Otherwise, it outputs 0.

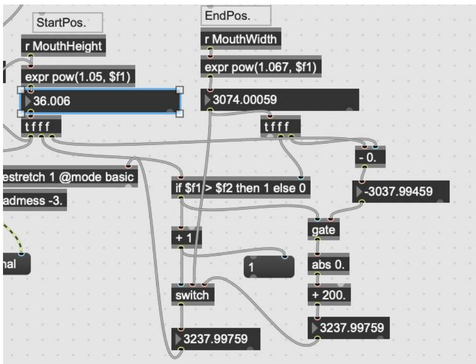
"gate" object: Receives the signal from the "if" object. The gate only outputs a value when the "if" object outputs a signal of 1.

"switch" object: Receives the signal from the if object after adding 1 to it, resulting in a value of 1 or 2. When the received value is 1, it outputs the mouth width value after exponential processing. When the received value is 2, it outputs the absolute difference between width and height.

As shown in Fig. 4, the value of the mouth height after exponential calculation always determines the start time of the looped audio.

Fig. 4

In Fig. 5, when the mouth is smiling (Width > Height), the "if" object outputs a signal of 0. The "gate" object does not output, and the "switch" object receives a signal of 1, directly outputting the mouth width value after exponential processing. This is sent to Loop Max of the "groove" object, defining the end time of the looped audio.



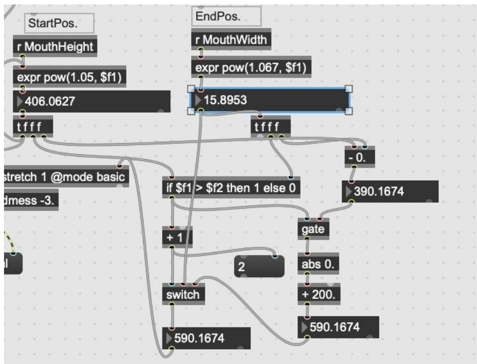
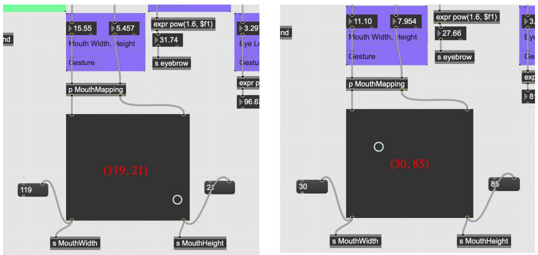


Fig. 5 **Fig. 6**

When the width value is less than the height value (Fig. 6), the sound effect is designed so that the audio stops when the mouth just starts to form an oval shape (i.e., when Height slightly exceeds Width, causing Loop Min to exceed Loop Max). As the mouth continues to tighten (Height increasingly exceeds Width), the looped audio speed becomes faster and more intense. Here is how I designed this behavior:

As shown in Fig. 6, when the mouth continues to tighten after forming an oval shape (Width < Height), to ensure Loop Max > Loop Min, the "if" object outputs 1, allowing the "switch" object to receive a signal from the "gate". The signal output from the "gate" is processed as |Width - Height| and then added to a constant *a*, ensuring that Loop Max is always greater than Loop Min. Therefore, I need to determine the value of this constant *a* to ensure continuous looping of the audio. Below is the method to determine constant *a*:

The numerical values of mouth height and width in the "pictslider" object (Fig. 2) generally approximate a decreasing linear function. By capturing the values of height and width at two different moments, assume Width = y and Height = x. As shown in Fig. 7, the relationship between y and x is approximately y = -89/64 * x + 9485/64. Then, after exponential function processing, assuming Height output is x' and Width output is y', we have x' = 1.067^x and y' = 1.05^y, which translates to 1.05^(-89/64 * x + 9485/64).



$$\begin{cases} 119 = 21\,a + b \\ 30 = 85\,a + b \end{cases}$$

Fig. 7

To ensure Loop Max is greater than Loop Min, we set the equation as: $|x' - y'| + a > x'$, which expands to:

$$1.067^x - 1.05^{-89/64*x+9485/64} + a > 1.067^x$$

Thus:

$$a > 1.05^{-89/64*x+9485/64}$$

Given that $x' > y'$, we can infer that:

$$x \in \left(-\infty;\ \frac{64\ln\left(\frac{1}{2^{\frac{9485}{32}}5^{\frac{9485}{64}}}\right) + 9485\ln(21)}{89\ln(21) - 89\ln(20) - 64\ln(10)}\right)$$

Since x is greater than 0, the range of x is approximately between 0-59.49, as Width output $y = 1.05^{-89/64 * x + 9485/64}$ decreases with increasing x, a must always be greater than the maximum value of y. The maximum value occurs when $x = 0$, with a's minimum value approximately equal to 1381, where Loop Max could always be greater than Loop Min. However, to design the condition where audio stops when the mouth starts to form an oval shape (Fig. 8), we also need to determine the minimum value of a that allows Loop Min to exceed Loop Max. In this scenario, x reaches its maximum in the range, yielding $a \approx 24.40$. Therefore, for Width < Height, the constant a must be between 24.40 and 1381 to ensure that the audio stops when the mouth just starts to form an oval shape and the looped audio speed becomes faster and more intense as the mouth continues to tighten. To make it easier for the mouth to reach a position where the audio stops, I set $a = 200$.

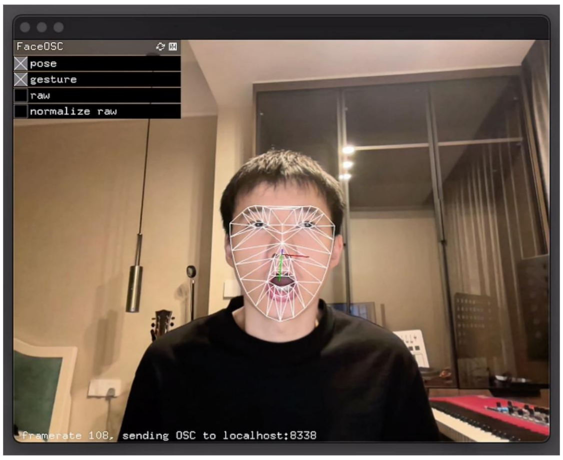


Fig. 8

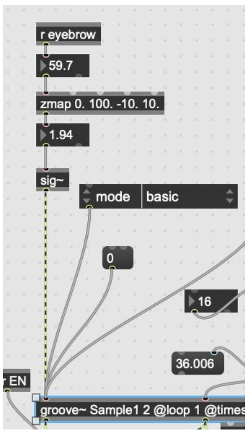


Fig. 9 **Fig. 10**

Next, I connected the groove~ object to a sig~ object controlled by the eyebrow (Fig. 9), allowing slight adjustments to playback speed. Finally, I connected the entire groove~ object to Serum FX (Fig. 10) to modify the timbre. With this setup, a smiling mouth results in slow loop playback, and as the mouth tightens, the speed increases until it forms an oval shape, at which point the audio stops. If the mouth continues to tighten, the audio speed increases further, and the tension of the interaction grows, creating a dynamic and interactive looped sound system.

1.2 Background Drone Sound:

As shown in the right part of Fig. 3, the background drone sound is also generated using the "groove~" object, which loops audio stored in the "buffer~" object. To create noticeable variations in the background drone sound, I connected the "groove~" object to a "pitchshiftcent" object controlled by my mouth width. When I smile, the mouth width increases, causing an increase in pitch shift and raising the pitch of the sound. Conversely, as the mouth tightens, the pitch drops, resulting in a lower-frequency sound. This allows the mouth width to effectively control the pitch of the sound. I then connected the "groove~" object to a grin delay sub-patch (Fig. 11) and a "yafr2" reverb object and adjusted the parameters of "yafr2", which could add granular texture and spatial depth to the sound.

This system results in a dynamic background drone sound, with the pitch rising as the mouth opens into a smile and lowering when the mouth tightens. The reverb adds atmosphere, making the drone sound affluent and immersive.

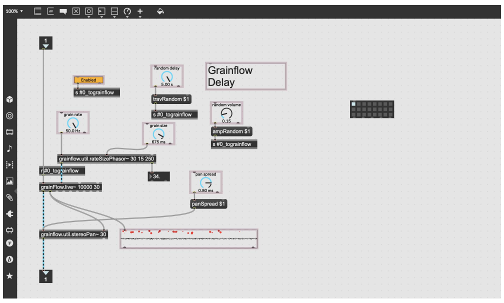


Fig. 11

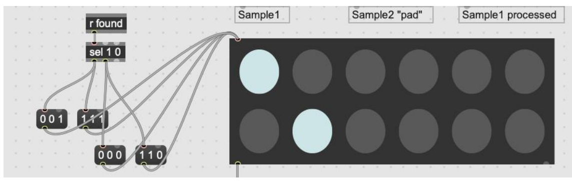


Fig. 12

Finally, I routed the background drone sound, the processed looped sound (after Serum FX), and the unprocessed looped sound to a "matrix~" object. The "matrixctrl" was then connected to a "sel" object that could detect whether FaceOSC was sending a signal (Fig. 12). This allowed for further interactive control—when I covered my face with my hands, FaceOSC would stop transmitting signals, resulting in the “sel” object controlling the unprocessed looped sound output. This enabled additional interaction using hand movements, enhancing the musical expressivity by adding another layer of performance control.

This piece utilizes the “sfrecord” object for internal recording, and I connected an AKAI Midimix controller to manage the looped sound and the background drone sound.

2. Music Documentation:

This piece begins with a representation of a newborn’s first arrival to the world.

During the first 20 seconds, my head is lowered, preventing FaceOSC from detecting my face. Due to the modulation by the “sel” object, the unprocessed looped sound remains silent, while the looped sound (after Serum FX processing) is set to loop an extremely short clip (0.001 seconds), creating a high-frequency, tension-filled sound (Fig. 13). This sound gradually comes out in using the Midimix, symbolizing the newborn's imminent arrival into the world.

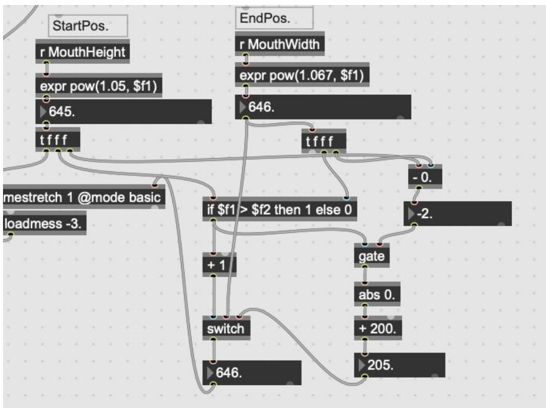


Fig. 13

After 20 seconds, I raise my head to let FaceOSC detect my face, initiating interactive control. My facial expressions and hand movements alternate, representing the newborn’s curiosity about the new world. The music swells and relaxes accordingly, embodying moments of curiosity interspersed with caution.

After a short transition, around 1 minute and 23 seconds, the background drone sound gradually enters, symbolizing a sense of danger. The newborn, represented by my facial gestures, begins to exhibit fearful expressions, and the movements become more rapid, increasing the tension in the looped sound. The music grows more intense, reaching a peak at 2 minutes and 14 seconds, where the music stops, transitioning into the final section.

After 2 minutes and 24 seconds, my head repeatedly lowers and raises, symbolizing a cautious release from fear. The tentative lowering of the head represents gradually letting go of caution, and eventually, continuous interaction between my face and Max signifies the danger has passed. The newborn becomes satisfied, and the music ends with several intermittent looped sounds, symbolizing the newborn's renewed curiosity and longing for the world.

3. Summary:

Overall, *Hello World!*, this piece of improvised interactive music embodies the experience of a newborn exploring the world for the first time. Through expressive facial movements and interactions with Max/MSP, it conveys the newborn's curiosity, fear of encountering danger, and the renewed longing for the world once the danger has passed. The performance strikes a balance between tension and release, designed to convey these three states in a flexible and dynamic way.

Electroacoustic Music: *Countdown*

YouTube link: https://youtu.be/WvVOjdn0SmU?si=8lPDOTMpOV4_UuVl

This piece is an electroacoustic music composition using the clock sound as a central motif. It centers on Alzheimer's disease, focusing on the countdown of a patient's life. Through electronic music techniques, the work portrays the different stages of illness, from the onset to the end of life. The work was mainly made by Logic Pro and is 3 minutes and 40 seconds in length, divided into three parts, following an ABA structure.

1. Technology & Music Documentation:

A1 (00:00-01:07)

The A1 section depicts the early stages of Alzheimer's, where patients often experience mild dementia marked by irritability and sadness. The ticking clock sound symbolizes the countdown of life. In the section's first half, the clock sound is continuously transformed using electronic music techniques, changing its speed, pitch, frequency, and timbre. For instance, using "Flex editing's Speed (FX)" feature, I altered both the speed and pitch of the clock sound (Fig. 1), while the "Flex Time - Polyphonic" technique allowed me to adjust the speed without affecting the pitch. These sound effects transformed a single clock sound into various auditory textures. To further develop the sound, I utilized numerous plug-ins and automation to modify its textures. For example, I applied "Valhalla Delay's Feedback" automation to create high-frequency transitional effects (Fig. 2), while "FabFilter's Micro" and SerumFX were used to morph the clock sound into a heart-beat-like sound through adjustments to frequency and peak (Fig. 3).

In the latter half of this section, I incorporated real voice recordings of Alzheimer's patient's conversations with their loved ones. These samples were processed through "Logic's Sampler" to generate different pitches and frequencies of breathing sounds (Fig. 4).

Additionally, I applied effects and transformations to the original dialogue using "Flex" techniques and overall tempo automation, bringing this section to a close (Fig. 5). This effectively captures the emotional instability and sadness experienced by patients when they find they have such a disease.

The ending of A1 includes a transition connecting Part A1 to Part B. This transition uses processed dialogue samples from Part A while introducing granular textures from Part B to create a seamless transition.

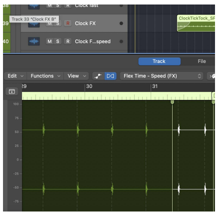


Fig. 1

Fig. 2

Fig. 3

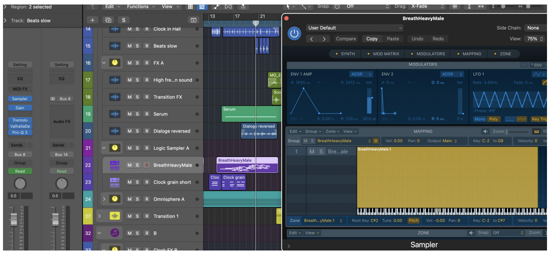


Fig. 4

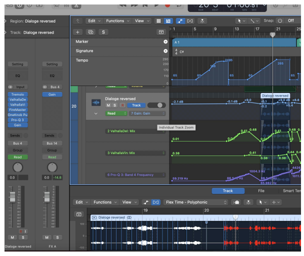


Fig. 5

B (01:07-02:22)

Part B is built around ever-changing clock sounds, with a lead synthesized by "Serum", reflecting the patient's progression into moderate dementia. As memory drastically fades and emotions become increasingly volatile, the sound design mirrors the patient's uncontrollable inner turmoil. The first half utilizes a wide range of plug-ins, samplers, and effects, with heavy automation, to create a dynamic, intense soundscape. For instance, in SerumFX, the lead combines a sine wave with a "BSOD_Square" preset oscillator, adding envelope and LFO modulation to the wavetable position and filter (Fig. 6), then processed with pitch bend to achieve a psychedelic sound, symbolizing the patient's mental decline into the moderate dementia stage. Simultaneously, the constantly shifting clock sound continues, accompanied by real patient dialogue samples from Part A1, processed with Ableton Live's Max for Live tools, including "Grain Scanner-Mercurial Buzz" and "Granulator III-Whispers" (Fig. 7). This layering represents the patient's growing instability.

In the second half, the volume and phase shifts become more pronounced. Using "Fab Filter's Pro-Q3" for high-cut automation (Fig. 8), I emphasized a harsh tearing sound to create a sense of intense conflict. Then, I introduced a second "Serum" sound featuring a combination of low-frequency sawtooth, two oscillators with rapid LFO modulation on their wavetable position, and a direct-out white noise layer, further enhanced with various effects and automation (Fig. 9). As these tearing sounds intensify, the frequency spectrum gradually brightens, representing the patient's intense inner struggle and sense of helplessness. The section culminates with a piercing 6k-8k frequency sound, leading to the climax of Part B.

The ending of B includes a transition to connect Part B to Part A2, continuing the dialogue samples from Alzheimer's patients processed through Max for Live. The transition is further enhanced by "Omnisphere" sound, blending seamlessly between B and A2.

Fig. 6

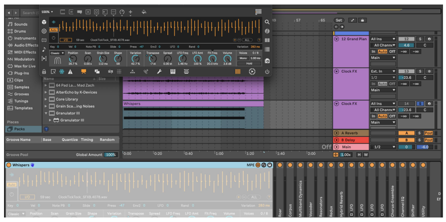


Fig. 7

Fig. 8

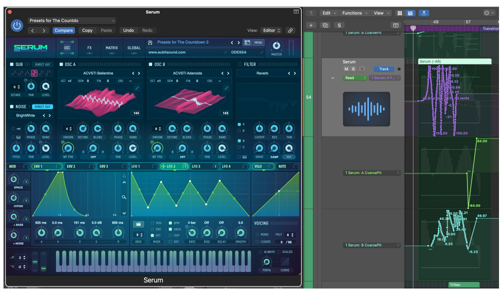


Fig. 9

A2 (02:22-03:13)

This section focuses on the late-stage progression of Alzheimer's, where the patients have almost lost all of their memories, can't take care of themselves, and will die soon. The clock sound continues as a recurring motif, symbolizing the final countdown of the patient's life. Technically, the clock sounds from A1 are revisited but reversed for added emotional depth (Fig. 10). I also recorded and manipulated the sound of flowing water to resemble blood flow sounds, applying automated "Bitcrusher" effects to create an evolving texture (Fig. 11), reflecting the subtle passage of life. In the first half of this section, the high-frequency transitions from A1 return, pushing the music toward another climax, symbolizing the patient's final moments of resistance.

In the second half, the clock motif reappears along with the occasional sound of flowing blood, while two tracks of a "Rhodes 1946 Piano" are introduced as the main melody. I detuned one track by -38 cents (Fig. 12), producing a distinctive sound that represents the patient's peaceful but inevitable decline into death, as they lose all self-sufficiency and fade into oblivion.

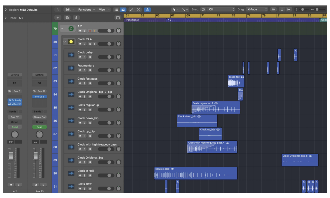


Fig.10

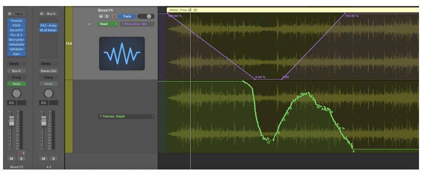


Fig.11

Fig.12

Coda (03:13-03:40)

The piano melody from A2 reappears with slight melody variations, continuing with the clock sound, which gradually fades away, symbolizing the end of the Alzheimer's patient's life. The piece concludes with an augmented triad, representing the tension and instability experienced by the patient throughout the disease's progression, which resembles the unresolved nature of this chord. Despite the tension, the coda ends with a calm sound, refining the central theme, encouraging the listeners to reflect on Alzheimer's patients' condition, and inspiring empathy and awareness towards patients.

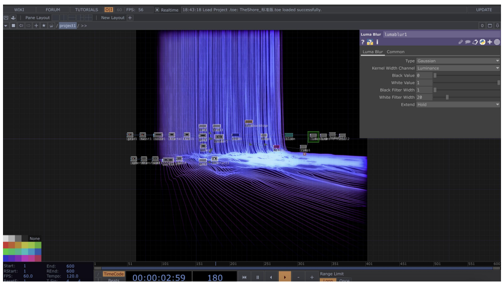
2. Summary:

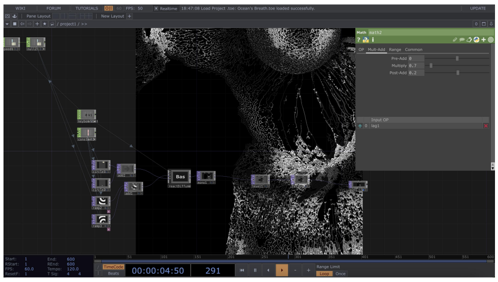
In summary, the electroacoustic music work *Countdown* is composed in an ABA structure, with the clock as the central motif, maintaining unity in material and clarity in structure. The piece employs various electronic acoustic techniques that align with its theme. Through my music, I aim to deepen listeners' understanding of the inner world of Alzheimer's patients, calling for more attention, care, and compassion for those affected by the disease.

Audio-Visual Interactive Suite: *Voice of Oceans*

Voice of Oceans is an audio-visual interactive suite inspired by the ocean, consisting of two movements: an ambient electronic music composition and an electroacoustic music piece. This suite is 7 minutes and 47 seconds long.

The first movement captures the ocean's mystical beauty with sweeping melodies and lush harmonies, while the second uses sampled ocean sounds, manipulated through electronic music techniques, to represent an ocean whose environment is deteriorating. The stark contrast between the two movements aims to raise awareness about protecting marine ecosystems. Additionally, I integrated both pieces of music with audio-visual interactive projects created by TouchDesigner, which represents the music through both visual and auditory experiences.





TouchDesigner projects

Ambient Electronic Music (Audio-Visual Interaction):

Voice of Oceans: 1st Movement

YouTube link: https://youtu.be/hr0HMvfDQMM?si=02Fh8Qrev-omMBox

Full Score link: https://drive.google.com/file/d/1VirOWoW_bptjUbLXjzwdYDpDvkGYwbyP/view?usp=sharing

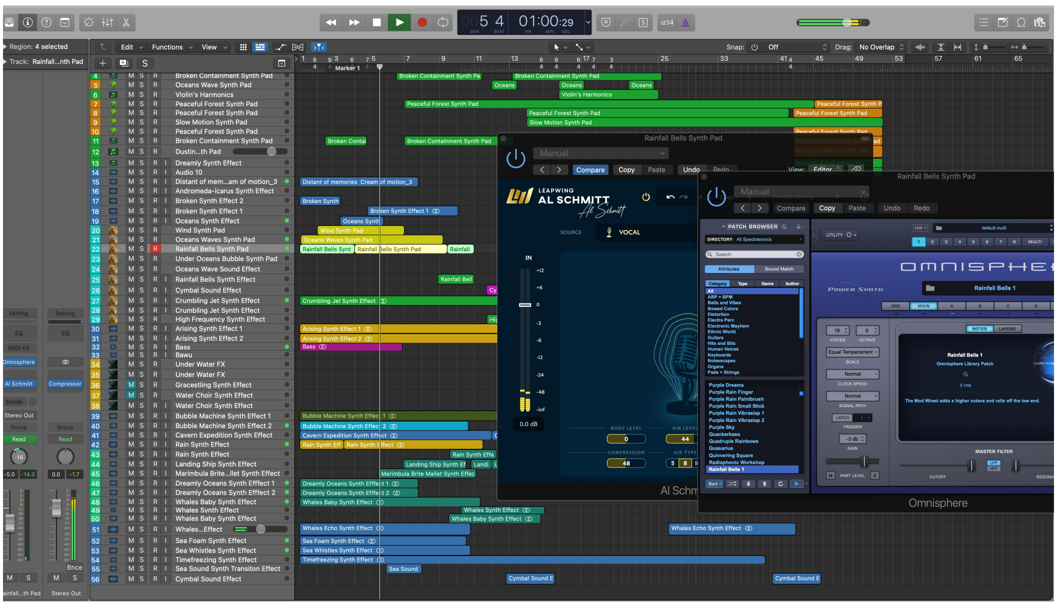
The first movement is an ambient electronic music piece. The sounds of waves and seagulls are paired with graceful electronic melodies, while shifts between major and minor keys and harmonic variations evoke the grandeur and mystery of the ocean. The music paints a stunning seascape, resembling a blue wonder, drawing in the listener.

This movement was created with Logic Pro and mainly features sounds from the "Omnisphere" synthesizer. The piece is structured in an ABC format and is 3 minutes and 45 seconds long.

1. Technology & Music Documentation:

A (00:00-00:57):

Utilizing "Omnisphere" patches like "Distant Memories" and "Cream of Emotion", section A begins with a rich, expansive soundscape in C# major. The sound is full-bodied and atmospheric, complemented by crashing waves, symbolizing the ocean's vastness and beauty. Additionally, I introduced variations in rhythm and tempo to break away from conventional time signatures, capturing the ocean wave's dynamic and unpredictable nature. In the latter half of this section, the music gradually modulates to A# minor, suggesting the sea's deeper, unknown parts.



The Project Screenshot

Parts of my plug-in

B (00:57-02:16):

The B section begins in C# minor, introduced by the Cavern Expedition sound from Omnisphere featuring striking sound effects, paired with dissonant metallic timbres from Kontakt's "Orchestral Essentials", symbolizing the dangers lurking in the depths of the ocean. Next, I recorded the sound of cymbal rolls performed by a drummer, using it as a transitional sound into the latter half. In this latter section, the music changes between the D#11 and Am chords, creating an unstable auditory experience reflecting the ocean's ever-changing environment. Finally, the section gradually transitions to an ending in E major with the Distant Memories timbre.

C (02:16-03:45)

Section C picks up the sonic material from the end of section B, naturally transitioning from E major to C# harmonic minor. This introduction leads into a more developed second half, incorporating deeper bass pads and bright arpeggiated sounds, accompanied by the sounds of rolling ocean waves. The section concludes in C# minor, musically depicting the beauty of the sea.

2. Summary:

The timbres in the first movement are cohesive. Using melodic electronic music composition techniques, the movement showcases this beautiful and mysterious blue wonder through ambient electronic music, evoking a sense of yearning among the audience.

Electroacoustic Music (Audio-Visual Interaction):

Voice of Oceans: 2nd Movement

YouTube link: *https://youtu.be/Mrwee9MxTIA?si=sAZSXctk4WmBo0un*

The second movement focuses on the destruction of marine ecosystems. I continued using ocean wave sounds from the first movement, transforming these sounds and incorporating dolphin call sounds as cues. Through electroacoustic music techniques, the material develops and evolves to depict the devastated ocean environment, making the audience feel as if the sea is lamenting.

I excel at and enjoy using the samplers in Ableton Live, as it allows for detailed and unique sound design. Therefore, the second movement is produced using Ableton Live, with nearly all sounds derived from manipulated real marine audio samples. The work has a total duration of 4 minutes and 2 seconds and is structured in ABA form.

1. Technology & Music Documentation:

A1 (00:00-00:59):

The A1 section begins with sampled recordings of natural ocean wave sound processed through the sampler serving as the foundational pad, enhanced with delay, resonator, reverb, and other effects, with some automation (Fig. 1), which is complemented by a low-frequency pad processed from this foundational pad with pitch shifting and phaser effects, alongside dolphin sounds that have undergone acoustic transformation (Fig. 2). The music gradually builds towards a climax in A1, concluding with a booming transition sound effect. The increasing distortion and intensity symbolize the destruction of marine ecosystems and the lament of ocean creatures.

Fig. 1

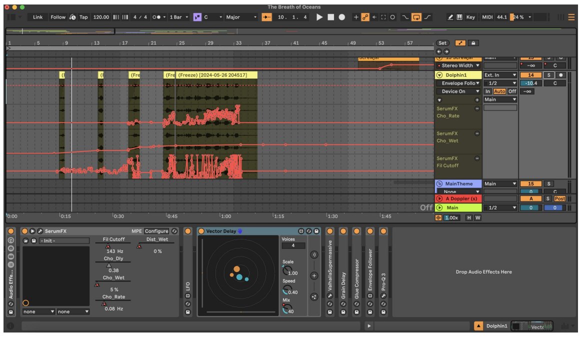


Fig. 2

B (00:59-02:31):

In the beginning part of this section, I introduce sounds of plastic bags that I recorded into the “Grain Scanner” sampler, using the “XY Pad” from Max Audio Effect to map the X and Y axes to control Pitch and Layers in the “Grain Scanner” for real-time manipulation of the recorded sounds (Fig. 3). The recorded sounds are then processed with an allpass filter in SerumFX that features a resonant “O” vocal sound (Fig. 4). This technique is also applied to dolphin sounds in the latter half, adding resonance from vocalizations like “ya” and “yi” to enrich the music. The dolphin sounds are also fed into another granular sampler connected to an “XY Pad” for real-time control of Pitch and Layer, followed by pitch shifting, vocoder processing, etc. This sound effect creates a low-pitched wail in dialogue with the high-frequency sounds, interweaving these experimental oceanic and waste sounds with elements of human voices, raising awareness that the destruction of marine environments ultimately threatens humanity itself.

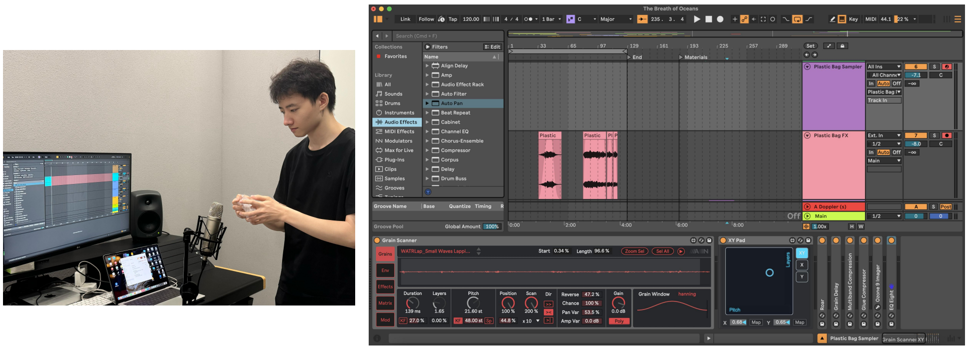


Fig. 3

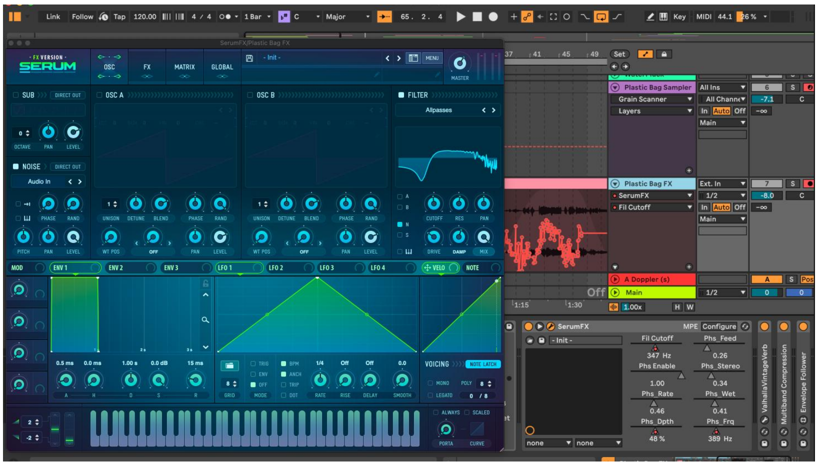


Fig. 4

A2 (02:31-04:02):

This section reintroduces the dolphin sounds from A1, with gradual pitch-lowering automation applied to create a descending wail (Fig. 5). This soundscape is complemented by the sound of plastic bags processed through an all-pass filter featuring changing vocal resonance, allowing the two elements to engage in a dialogue that reflects the decaying state of the marine environment and the devastating impact of pollution on ocean life. The piece concludes with the sound of waves processed through a spectral resonator, gradually reducing the wet/dry mix, which brings out the original pure ocean waves' sound for a moment, prompting the audience to reflect upon the theme.

Fig. 5

The materials used in the second movement are unified and straightforward, focusing on the increasingly dire state of the ocean ecosystem. Through electronic music techniques, this movement portrays a polluted sea, allowing the materials to engage in a dialogical processing that conveys oceans' wails, urges the audience to protect the marine environment, and encourages reflection.

Summary:

Overall, *Voice of Oceans* is a suite consisting of two strongly contrasting audio-visual interactive music pieces. It aims to deeply resonate with listeners about the damage inflicted on the ocean environment, using electronic music and audio-visual interactive technologies to highlight the importance of its protection.
