

Tileyard London

Development Module

Element 2b: Instrument Essay

MA Commercial Music Production

Philip Emerson
Student ID 2009419

Word count 4398

Introduction

In this essay I will discuss my creative process in designing my own instrument. My instrument is three fold: using a software program I have written to manipulate data collected from an external hardware sensor to trigger and manipulate samples of spoken vocal recordings. I will discuss the choices made along the way and how I would improve it going forward. “The human voice is the first and most natural musical instrument, also the most emotional” (Schulze, 1995)

Reasoning behind my software/hardware choices

Max/MSP Software

As this was my first time trying to create a digital instrument I researched software programs that I could use to build it. Lacey (2009) compares some of the top platforms to create music software. In his top 5 he mentions Open Frameworks and Juice which both run on C++ coding, but as someone with no previous coding experience I was interesting in finding software that would appeal to the way I learn visually, so I looked further into the other three programs he mentioned. Processing, Pure Data (both free platforms) and Max/MSP.

According to Smith (2008) Max/MSP has the capability of being used by artists, composers, performers and technologists to create, control, and interact with almost anything that is able to be represented as a digital signal. For example Smith (2008) used the program for various live performances including one where he triggered multiple tracks as clarinettist and percussionist in “Points in the Sky” to give the effect of conducting the piece with only a foot pedal. Reveillac (2018) claims that with Max/MSP you can do almost anything you could imagine for audio and even video, with an open structure that supports connections with numerous external connections such as electronic instruments and even robots. The more research I completed into which platforms to use the more I was drawn to Max/MSP, although the cost (\$399) was off-putting, I realised you could get 30 day free trial, so I would need to create this “instrument” in that time frame.

Leap Motion Controller (L.M.C.) Hardware

I wanted to be able to manipulate my sounds using gestures and movements, instead of building an instrument that recorded sounds from its surroundings. It was important to me that this new system would allow the collection of positional data to control these manipulations. Therefore, I quickly realised that building my own circuitry for example would never be able to collect the type of positional data effectively in the way I desired. Instead I spent time reviewing other ways to collect situational data, during this review I read a few different case studies where people had used their mobile device as the sensor. For example, Aroca et al (2012) looked into utilising smartphones by using its camera and reading its sensors to build and

control a robot. In their research they state that most mechatronic systems and sensor networks need sensors to interact with their surrounding environment and their own state to be able to decide what to do next. Even though they used their phone in the system, they still had to use some microchips and external hardware as well, so I decided this was not the correct route for me. I required a more sophisticated type of sensor than what a phones camera could provide. I decided to look into premade sensors (see fig.1) rather than building one myself.

Results of the reviewed articles.

Name	Mapping	Sampling Rate	Connectivity	Cost
LMC	Algorithmic	50–200 Hz ¹	USB 2.0 or 3.0	US\$80 [96]
Optotrak marker	Strobe	120 Hz	Wired/Wireless	Not available.
Myo Armband	Electrodes	200 Hz	Bluetooth	US\$200 [97]
Creative SENZ3D	Depth	30 Hz	USB 2.0 or 3.0	US\$79 [98]

Figure 1: Comparison to other sensors on the market Guzsvinecz(2019)

The Leap Motion Controller (onwards referred to as L.M.C.) is a sensor designed for hand motion tracking, it has three infrared LED's to detect infrared light and two cameras to create grayscale stereo image. The LMC collects raw data and utilises algorithms to calculate the data of the hand and provide real time tracking of skeletal motions in a 3D space. (Guzsvinecz et. al, 2019) The Creative SEENZ3D even though is a similar price it can be controlled by voice as it has a microphone and can capture facial expressions as well, however these are all additional features I didn't need. In a study of 100 participants McCartney et al. (2015) found that L.M.C. had an accuracy rate of 92.4%, so I decided to go with the higher sampling rate and accuracy of the L.M.C.

What is my instrument?

As someone who prefers to create instrumental music, my initial thought was to make my own analogue synth instrument, however, after listening to all the crazy ideas (silence as an instrument, electrodes on plants etc.) in lectures I decided I wanted to think outside the box and push myself. I use instrumental music to convey a message without lyrics, I prefer to let people find their own meaning in a piece of music. When I want people to be emoted about a specific topic I will incorporate spoken word recordings: scientific lectures, interviews or news reports etc. so I wanted to see if it was possible to create a live performance instrument that could create its own narrative of spoken word with hands free manipulation.

At first I had considered creating a bank of phrases or pre-recorded audio clips, or using the most common words in the English language, but after a few brainstorming sessions I decided to look into singular words and find the ones most commonly used in the news in 2020. Even though this makes it very niche and time sensitive, I felt drawn to make an instrument that would reflect the realities of the time in which it was made and be emotive due to the weight of the words in our present day. Ultimately the samples could be replaced with other words or audio samples, but for this instrument now I decided to use topical words.

According to Shariatmadari (2020) from the Guardian, Linehan (2020) from the Irish Times and the “words of the week” section from Merriam-Webster (2021) some of the most popular words in the news last year can be seen in the fig. 2 table below:

Source	Words								
The Guardian	pandemic	Herd immunity	Lockdown	zoom	Megxit	Antifa	BLM	Karen	Simp
Merriam Webster	Cusp	Hindsight	sclerotic	Override	turbocharge	loony	ovation	singularly	antivax
	extremism	recall	Hate crime	Vaccine	unwitting	Scorched earth			
The Irish Times	covid	facemask	progress	truth	Social distancing	Trade war	sustainability	Flatten the curve	Lockdown
	Climate change	unemployment	outbreak	symptoms	trump	migrants	quarantine	progressive	Identity politics
	Hand sanitiser	Fake news	Self isolate	New normal	wuhan	stimulus	Work at home	coronavirus	

Figure 2 Words from the news 2020

From these words I narrowed it down to 10 words that could evoke a stream of consciousness about a certain topic, but without putting any pre conceived connotations on it. A performance instrument that would simply put a word out there and let the listener do with it mentally as they please. My 10 chosen words were: progress, freedom, isolation, compliance, misinformation, quarantine, truth, symptomatic, vaccination, totalitarian.

My first hurdle

The sensor was meant to arrive within three business days, however thanks to the combination of Brexit and Covid restrictions it took almost three weeks to arrive which meant I lost a lot of time in learning how to use it and program it. While waiting for the L.M.C. to arrive I researched the words (above) and recorded my samples. While waiting for the sensors arrival I also tried to skill up on programming from absolute beginner. I enrolled on a course with Wright (2021) to learn how to use the Max/MSP program with Stanford University, it was very beneficial, but trying to squeeze such an intense course meant my brain was learning a lot of new things in a very short space of time and it was very overwhelming to try and learn what felt like an entirely new language and way of communicating with my computer. I had to take regular breaks to allow myself to untangle what I was learning – awaiting the brains intuitive action of out of sight information processing to piece it all together.

The L.M.C. didn't arrive until the 17th March, and even though I had spent the time waiting trying to learn how to use Max/MSP, it was a completely different experience trying to program the sensor. Until I had all the components together I wasn't able to confirm if I could achieve what I hoped to do with my limited time and skills. I wondered if I had bitten off more than I could chew, but I persevered as I believed in the concept and was excited to see what innovative things I could achieve with it musically if I was able to complete the project in time.

Utilising the sensor

As the sensor is pre-made I anticipated having access to various drivers and patchers that other people may have created to draw inspiration from. Mueller (2014) stated technology users should be aware that sensors are designed to provide a numeric input to the computer, and that to gain an input from the sensor there may be a need to write a low level code if it is not supported by the platform or needs a specific driver, which I found to be true immediately. There were much less resources on how to utilise the L.M.C. with Max/MSP than I imagined and the majority of them were to be used with mac, in fact I only found two projects where people used it to create digital instruments on PC drivers, but both were with old versions of the software which were no longer compatible with each other. After many laborious hours I found a way for my L.M.C. communicate with my Max/MSP, using a program called "Reach" that opened a port between the two devices to allow them to speak the same language.

Deciding how to use it for my instrument

As can be seen in figures 1-4 in Appendix A, the L.M.C. has a variety of real time parameters, gestures and motions etc. that can be recognised, such as: palm position, palm velocity, finger extension, pinch strength, grip strength, rotation, finger spread, swiping motion, tapping motion and many more. I decided to reduce my gestures to a more manageable set of data to be manipulated, I thought it was best to isolate ones that were easy for me to carry out and less likely to get mistaken for other gestures. I quickly vetoed pinch, swipe etc. as parameters as I wanted more concrete ones to be used for triggering tracks, so I decided to utilise finger extension as a way to trigger audio playback. Thankfully the program I found when I got the Reach portal to work between L.M.C. and Max/MSP had a patcher designed (see fig.5) to recognise whether the fingers were extended or not, I was very grateful to be able to use it. I had trialled many hand gestures and got a peer to try them out as well (see feedback 1 in Appendix E) which helped eliminate some gestures as we weren't both able to extend certain fingers easily, so was good to factor that in to the decision process. Table 1 in Appendix B shows my chosen trigger gestures based on finger extension.

Alongside recognition gestures and motions the sensor can also pick up position in relation to the sensor, so I knew I would want to program these XYZ positions to some control parameter to manipulate the triggered sound.

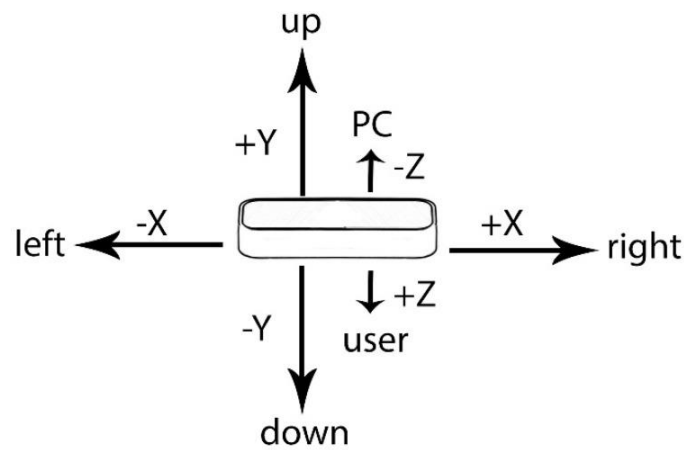


Figure 3. The different axes of the LMC (Guzsvinecz 2019)

After deciding what data collection I was going to use for my instrument I had to plan what I wanted each one to control. Fig 4. Shows an outline of what I wanted each thing to do

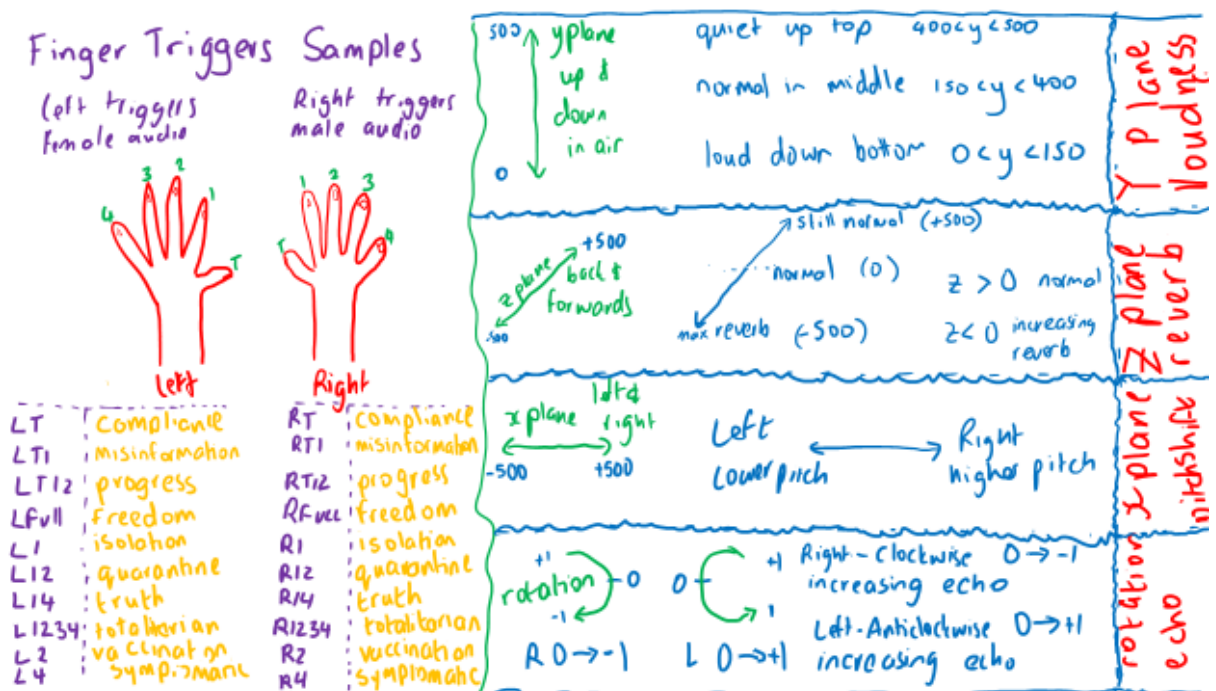


Figure 4 Visualisation of how my instrument works

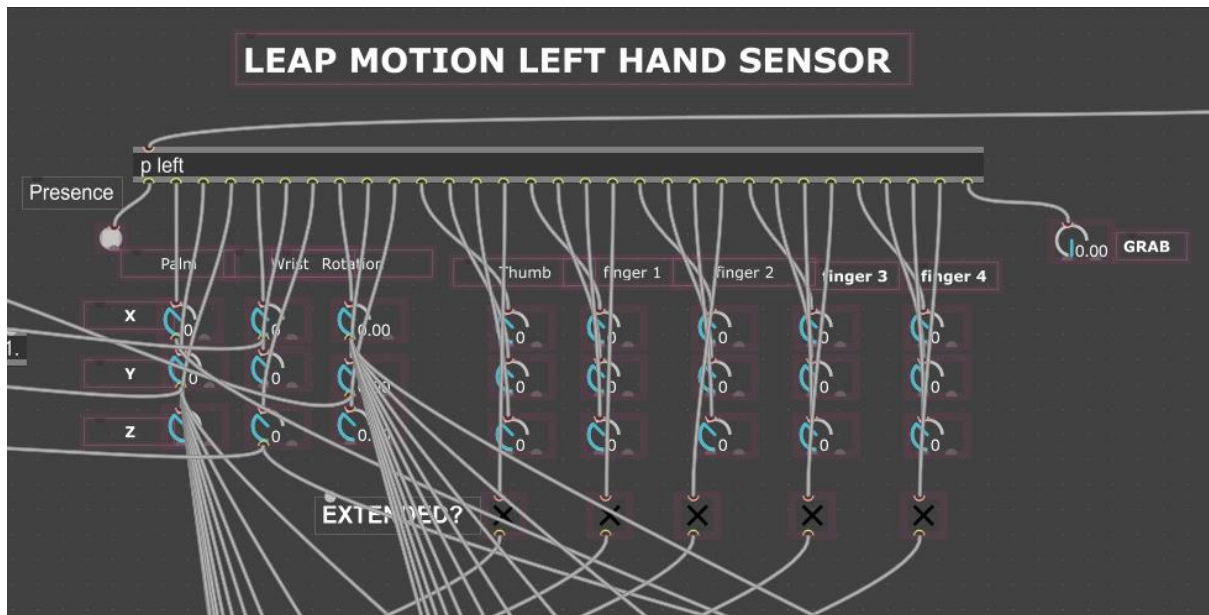


Figure 5 Data collection from left hand

As well as recognising finger extension and the XYZ position, I also decided to utilize the rotation of the hand and the recognition of a grabbing motion (closed fist). I decided rather than using the y-plane to change volume or velocity of audio sample I would split it into three regions which specified which version of the audio sample was triggered based on the hands gesture. In the x plane I wanted the pitch in relation to how far left or right I was from the sensor. For the x plane, the further I pulled back from the sensor the more I wanted to increase the reverb. And the rotation of my hand I wanted to control the echo of the triggered sample.

My first adaptations

At first I had planned to trigger one sample and use the y-axis adapt the velocity to make it sound quieter and louder, but ultimately I recognised that simply manipulating one sample wouldn't give a different timbral effect. Instead I decided for each word I would record three different versions, using the timbre of the human voice to portray different sounds. So even though they all said the same word in the same "key" so to speak, they had different properties: a whisper, a normal voice and a raised voice (not shouting as I didn't want it to seem aggressive, just noticeably louder)

I had also originally planned to use one hand to trigger and manipulate words and the other to trigger and manipulate sounds, however I decided for I would rather have a pre-recorded track that I used this instrument to perform over the top of rather than try to create a track whilst simultaneously trying to create a narrative, it seemed unnecessarily overwhelming.

I also decided it would be more impactful to have a female voice recorded as well as my own, so that it gave the aspect of shared experience, sometimes triggering the same word sometimes others, sometimes one whispering and one shouting. I felt that this would have a lot more poignancy, so I programmed one hand to trigger

female and one hand to trigger male, but with the same gestures triggering the same words.

When I was first planning this I had considered also using the tilt of my hand to add a high pass filter or low pass filter on the triggered sample, however I decided against this as when I applied them to the sample it took away the authenticity of the voice and I wanted it to still sound like human speech.

Programming and testing the parameters

After deciding what I wanted each parameter to manipulate it was time to start trying to program them in. The first gesture I wanted to define was a positional one in each of the three axes shown below

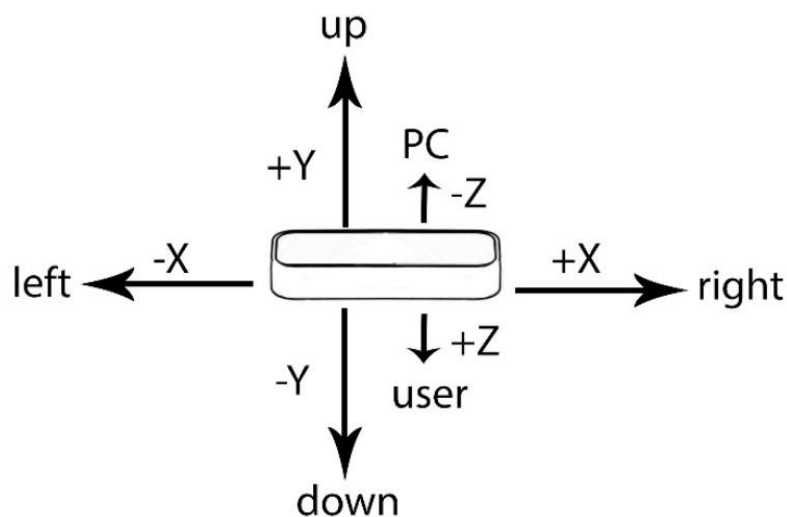


Figure 3. The different axes of the LMC (Guzsvinecz 2019)

The L.M.C. collects the positional data in the XYZ plane, each outputting raw data on a scale of -500 to +500. Fig.6 shows how that is output on dials in presentation mode of my instrument

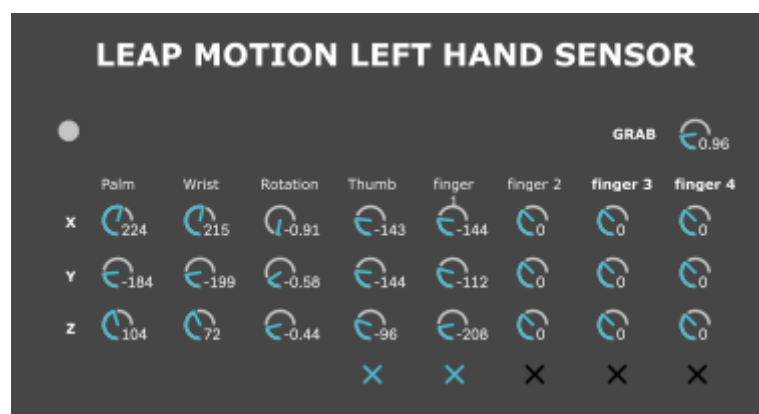


Figure 6 Raw data collection represented on dials

I tested the boundaries of each of these to ensure I was scaling the parameters appropriately and ended up adapting the range as stated below:

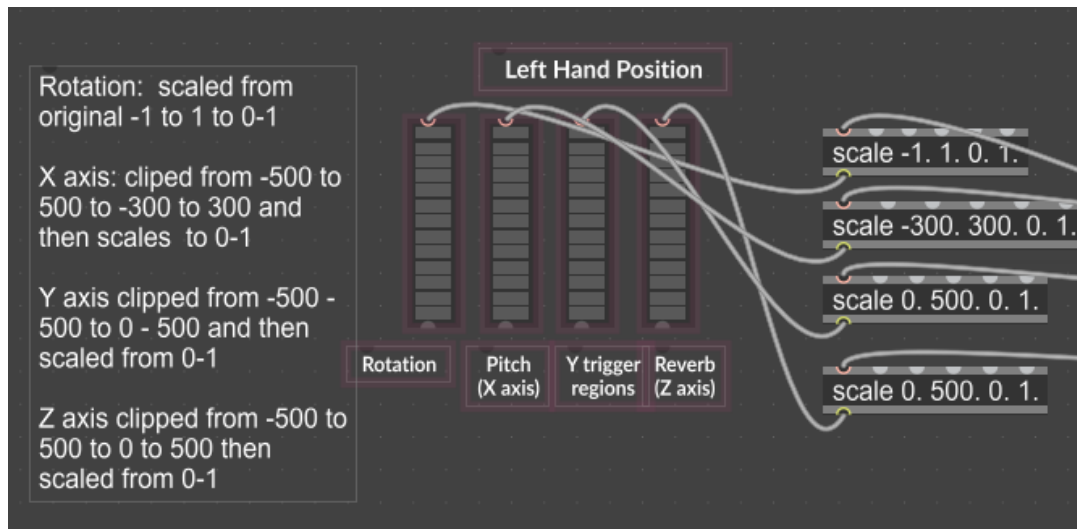


Figure 7 The clipping & scaling from raw data to visual output

X-axis: I realised due to my natural arm extension left and right I typically never went beyond the range of 300, so when I got someone else to try it out (see feedback 3. In Appendix E) they mentioned the extreme ends of the pitch not sounding good, so I adapted the parameters to deal with this and ensure it didn't go beyond my natural reach by clipping the original range of -500 to 500, to a range of -300 to 300.

Y-axis: As the sensor sits on the table I never went below into the negative range so adapted parameters to reflect this. I clipped from original range of -500 to 500 to a range of 0 – 500

Z-axis: I realised I only wanted the Z to be the pullback motion towards myself not pushing forward into the monitor so didn't need the negative range so I clipped from original range of -500 to 500 to a range of 0 – 500.

I then scaled the ranges to float values between 0-1 to satisfy OSC standard for their visual output metre. I ensured these metres were shown in presentation mode so that the user could see a visual representation of the position of their hand as well as seeing the numerical dial value.

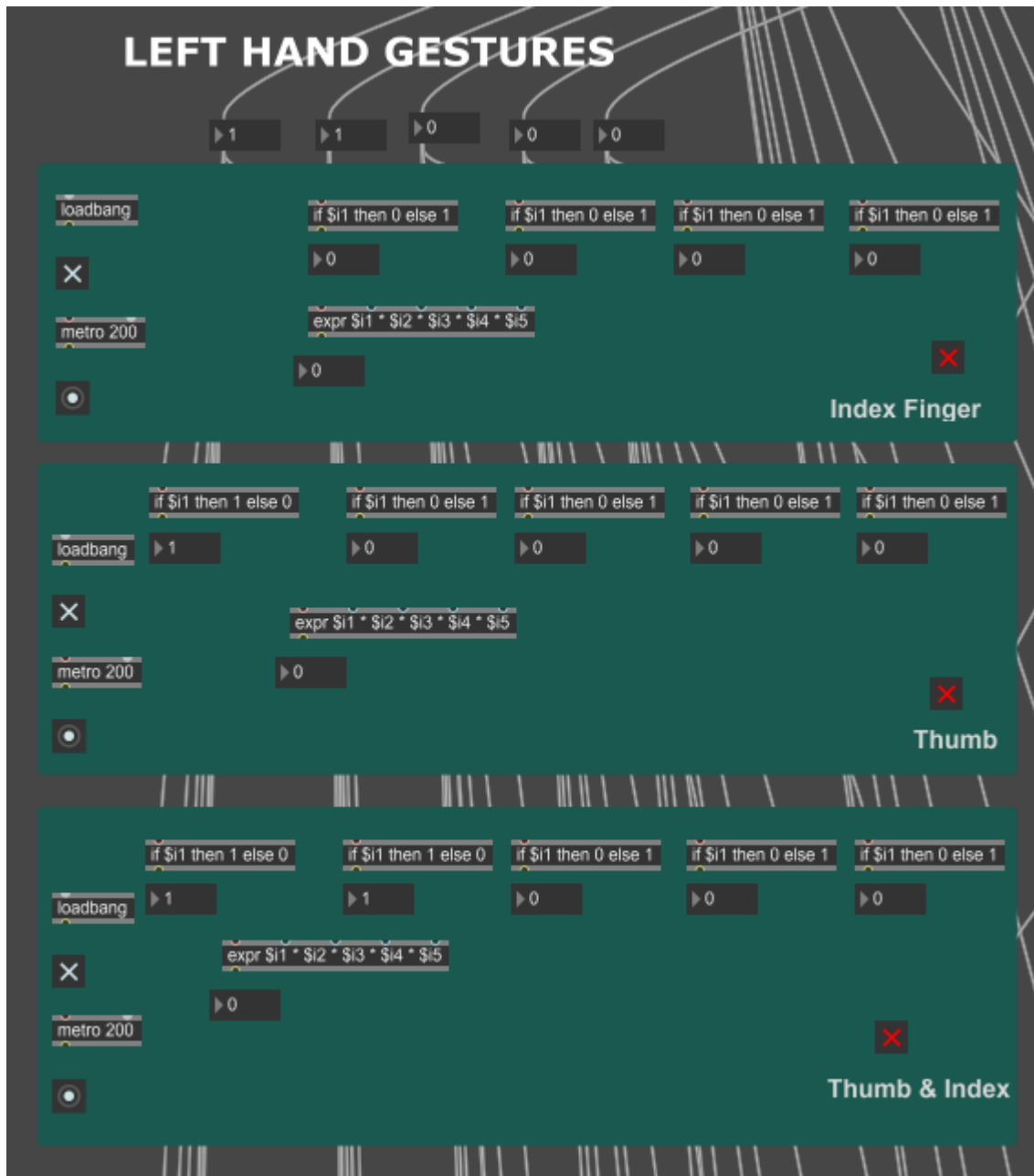


Figure 8 programming the gesture triggers

Fig. 8 shows the code I designed to recognise the trigger gesture. For the 10 different gestures some had a combination of fingers, so I had to program each gesture individually, because at first when the index finger was up and the thumb it would play three different samples, when I wanted it to only play the one associated with both being up. So to get around this I had to do it ten different times with its own if statements to only output a bang if it had the fingers I desired alone, and if any others were up it would output a zero. Then the command would be sent to the associated audio samples (see Fig.9).

As you can see on the left of fig. 8 there is a metro 200 to stop continuous streams of triggers being sent (see feedback 2 in appendix E) the new samples were getting triggered continuously when moving between different regions of the y-axis, so this

bought a little bit of time to prevent the signal being sent immediately as moving into next zone.

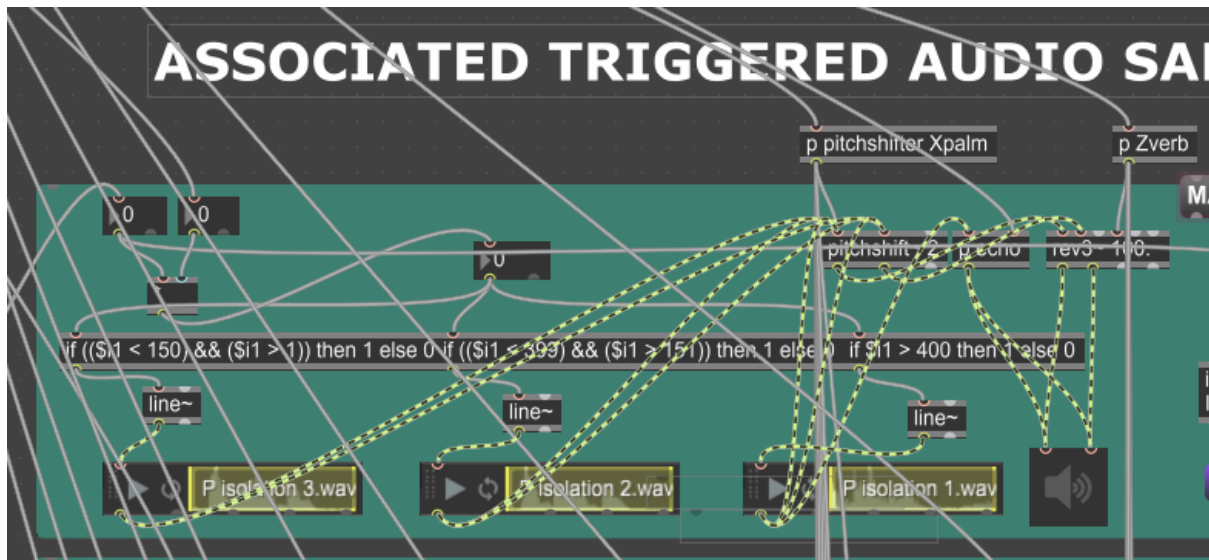


Figure 9 Programming the sound manipulations

In fig. 9 when a “1” was sent to this box it would multiply with the y-position data and the ‘if’ statement would interpret this to play sample. So if the position of the hand on the y-axis was: < 150 it triggers the loud sample, up to 399 triggers the normal sample and over 400 triggers the whisper. Once the track has been triggered the signal goes into the pitch which has the position of the x-axis input into it, so the pitch will shift up or down to match its detected position. The same process happens with the reverb receiving input from the hands position on the z-axis and increasing/decreasing the reverb dependent on that position.

One problem I encountered here was it triggering tracks simultaneously, eventually I realised if I dragged the output of my audio into the pitch shift and another one into the reverb and then put both into the ezdac I was getting two samples playing, which I didn’t want to happen so I had to make it a linear process where the shift was applied to the sound, then the reverb and then the echo, if applicable. The echo only applies to the whisper as I felt when it was on the more prominent samples it cluttered the sound scape.

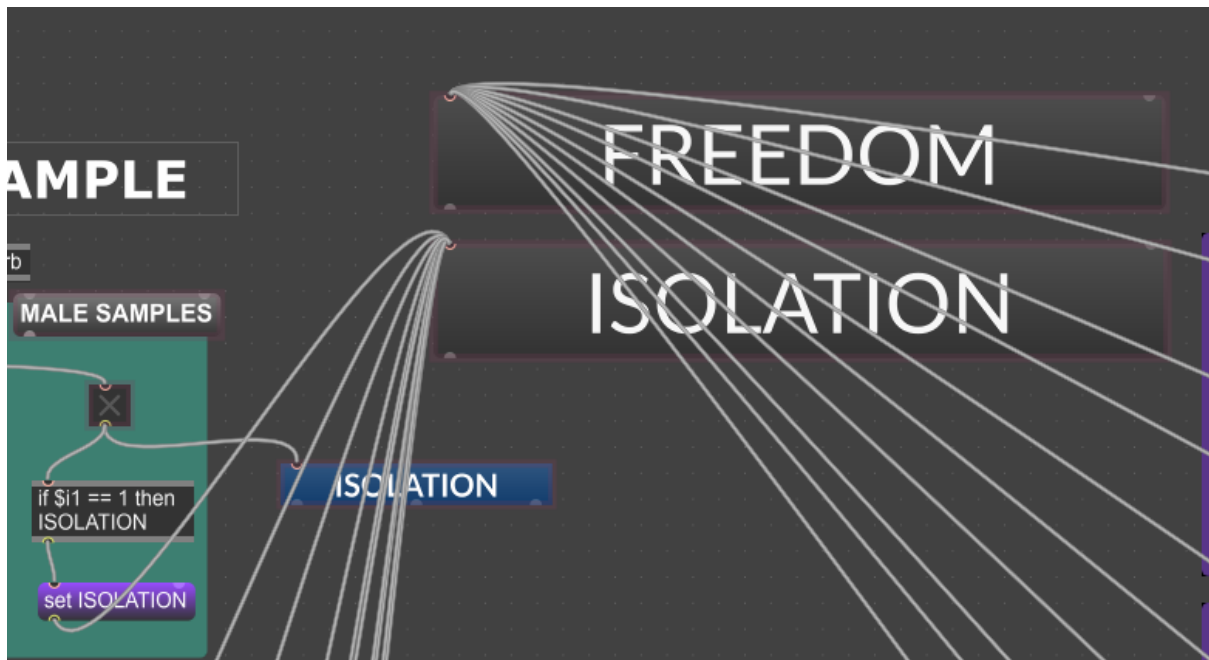


Figure 10 Programming the word pop ups

As can be seen in fig.10 above, I used an 'if' statement to trigger the word if it was the chosen sample (=1) and if 0 it doesn't display the word. Then dependent on which word has been chosen it would display in the large message box. Even though technically this has nothing to do with the sound aspect of the instrument this was a visual choice so that the user/ audience could see which word had been triggered, as can be seen in Figure 11.

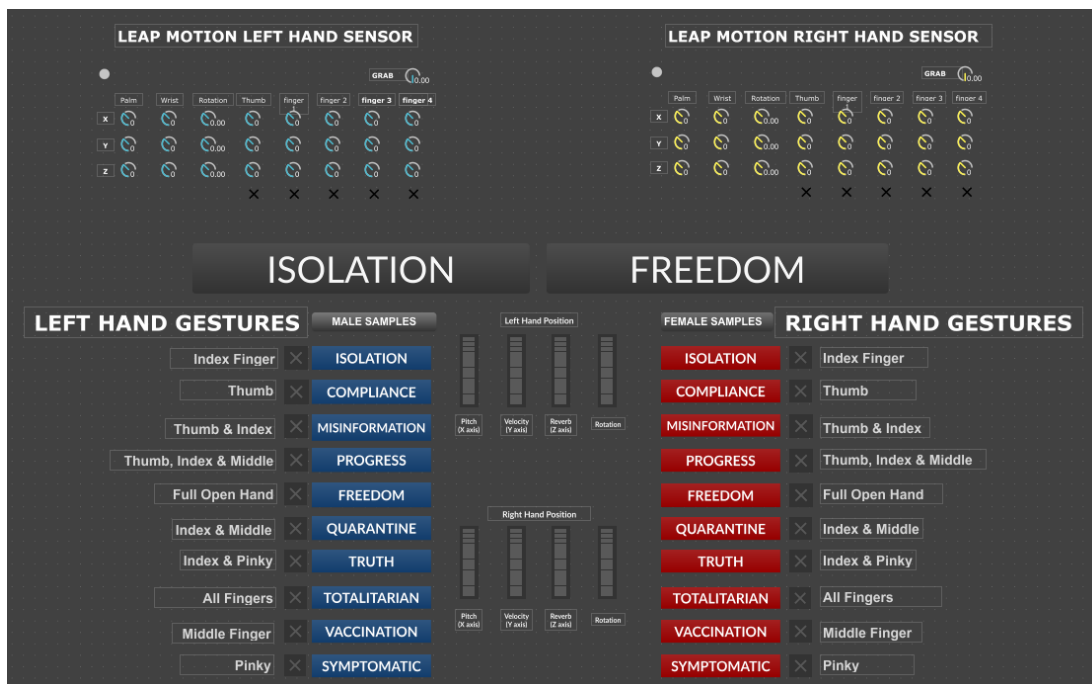


Figure 11 The presentation mode of my instrument

When performing using this instrument the user interface could be displayed for the listeners on a screen, so even if the word was an echoing inaudible whisper you

could see which word had been triggered (male/female) in the message box. I also chose to have the gesture trigger in the presentation mode so that if the performer knew they wanted to trigger a specific word the instruction on how to do so was there.

Issues

Throughout this essay I have tried to address issues that I had to deal with during this process, and acknowledge adaptations I made based on feedback and my own perception, but here I will address a couple more problems I encountered where I had to give up what I was doing.

Programming the words to pop up proved more challenging than anticipated. They kept overwriting each other and flashed continuously. My gestures triggered the reoccurring bang, so to stop it refreshing constantly I had to get it output a one so I could create an 'if' statement to trigger my word display once. However, during this process of solving this, something went very wrong and ended up ruining my program. After hours of trying to problem solve, hoping to find a misplaced cable or number somewhere I had to admit defeat as nothing was triggering anymore, the more I tried to fix the worse it became. Thankfully, I had uploaded a copy of the program earlier in the day to the cloud so I had to go back to the old version and start from there. Writing that portion of the program again was soul destroying, but I understood it better the second time.

Another issue I came up against was trying to record the vocal samples, as I live in a van and the weather has been terrible I couldn't record the audio in my van due to rain noise and waiting for the conditions to improve was fruitless, so, I reached out to a family member who allowed me to come and record the samples inside their closet with lots of soft furnishings to dampen the noise.

Comparisons

When researching my idea I found some other people using sensors as a musical instrument:

- Rerunner (2009) Graphics tablet Theramin which has 5 degrees of high res expressive freedom: x-axis pitch, y-axis volume, pen pressure=vibrato, Tilt-axis y=vibrato rate, tilt-axis x=detune semitone.

This similarities of this instrument is that it uses positional data mapped to a software instrument to control properties of a sound. In this case it is a sound created by the software rather than a pre-recorded one, I liked the detuning and vibrato aspects, but not for a spoken word sample.

- Virtual DJ (last performed 2009) uses tracking capabilities to allow users to use the space as a performance tool, they wave their arms to access audio loops, play samples and open synthesizers, all the while robotic lights follow and dynamically change with the audio.

I liked how the listeners were the performers in this instrument, the thought of adding visualisations such as lights that dynamically change with the music is very intriguing.

- Earnshaw (2014) uses a L.M.C. sensor to capture the position of his drumstick, the app he created sends MIDI messages to a Theremin style synthesizer. He mapped the X-axis to the pitch of the synthesizer, and the Z-axis to volume and Y-axis to depth of added LFO Oscillation.

He used positional data to control a synthesizer, similar to the basic principles of my instrument

- Fujimoto (2013) uses his right hand to control effects and left hand does synth sounds on the spot, all the while he beatboxes.

Even though none of these directly contributed to the development of my idea it was good to see it positional sensors being utilized, so I knew it was possible on some level. None of the aforementioned people have produced it as an instrument, just something they were playing about with themselves. However, Joseph Sprint (Damn Done Deal (2014)) has released his own interactive instrument free under creative commons and it has inspired me tremendously in ways to improve mine going forward. He uses a L.M.C. sensor to create his own digital Theremin in Max/MSP as his University Project which he calls "Theremax"

Reflection & Improvements

Looking at what I have created, I am proud of what I achieved in the time with zero prior knowledge. The entire process was one giant problem solving expedition. Early on I hated everything about it and regretted thinking I could do it, but as I overcame each small hurdle and little by little every small programming victory made me enjoy it more and more. When I look at the Theremax instrument (fig.12) that Sprint (2014) created I am ashamed of how limited my presentation mode is in comparison, however I feel inspired to try improve mine further. Creating this instrument was more about actually getting it working and doing what I wanted it to do in the limited time I had, however I feel with more time now that I have more knowledge I could now implement some of the feedback I received and draw inspiration from similar instruments to develop mine. For example in Feedback 4. (See Appendix. E) The user mentions being able to insert their own audio tracks, in its current state if somebody wanted to replace my audio tracks with their own they would have to open up the max code and swap them out an improvement would be if in the presentation mode the person was able to do that themselves, but I am not sure how I could do

that. However one adaption that I feel confident I could improve on now is what each of the control parameters does. As you can see in the Theremax (fig.12) the user is able to choose different options about the instrument before they play. I would like to design it so that the user could choose what they want each of the parameters to control. For example rotation is currently echo, but they could choose from a drop down menu of three possible effects/filters etc.

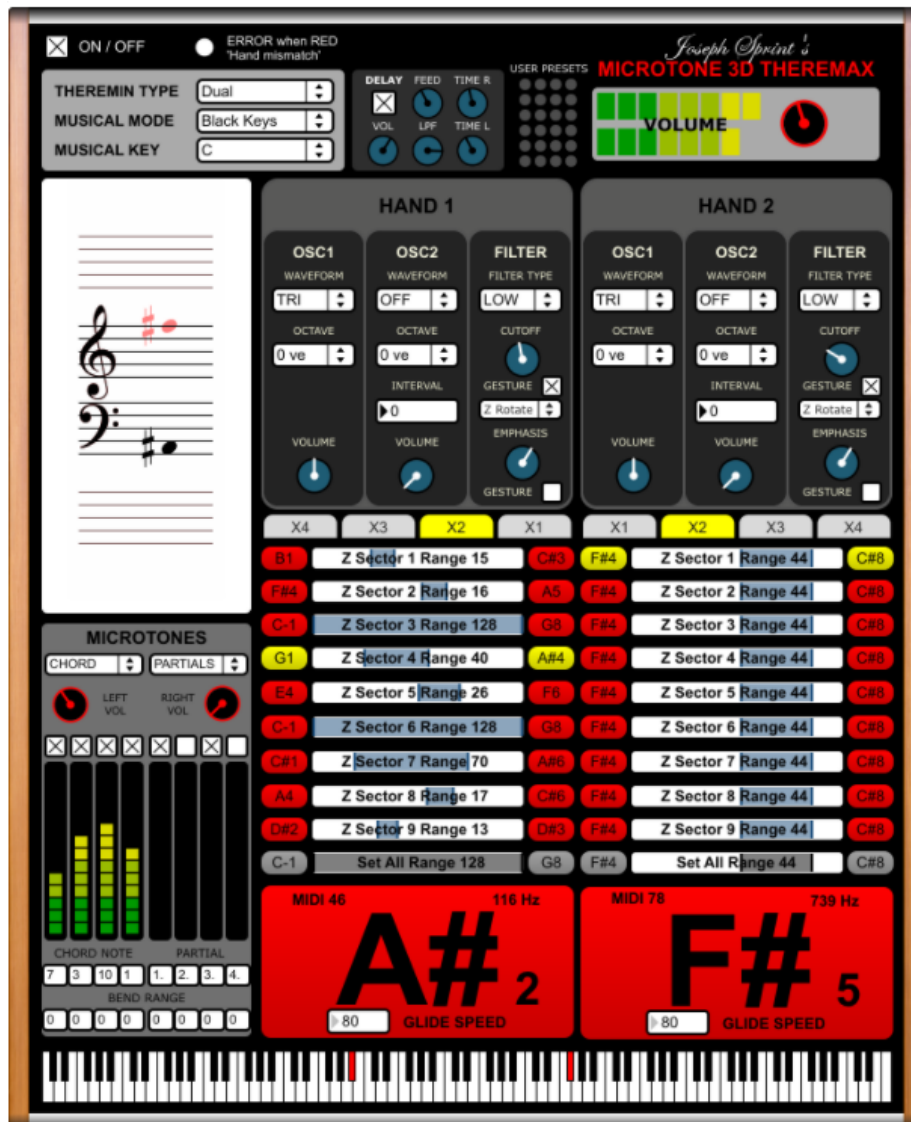


Figure 12 Theremax Presentation Mode

The Theremax has such a user friendly interface, it looks like an actual plugin whereas mine is very amateur. As well wanting to increase users ability to control the parameters and triggers I also like how visual the Theremax is, showing where the notes are etc. which links to Feedback 5 (see appendix E) where the user said a graphical output would bring it to the next level. I have small metres (see Fig. 13) that show the positional and rotational outputs visually, however a proper graphical output of the sounds would be incredible, with the word appearing in front of the coloured graphics on a video output display.



Figure 13: Visual output metres

Conclusion

Overall I am pleased with the instrument I have created, I am aware it has a lot of room for improvement, but it has fulfilled my creative vision. I have performed with it twice for friends/family and people have been moved by it. In this current world we live in something as simple as an echoed whisper of “isolation” over an instrumental synthwave track can move someone to tears. As Schulze (1995) said, the human voice really is the most natural and most emotional instrument we have, and a voice speaking of shared experiences resonates.

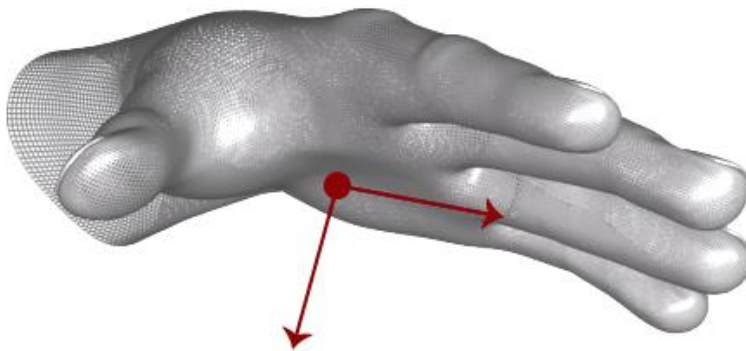
Appendix A

Leap Motions Controllers Capabilities

Hands

The hand model provides information about the identity, position, and other characteristics of a detected hand, the arm to which the hand is attached, and lists of the fingers associated with the hand.

Hands are represented by the `Hand` class.



The `Hand` `PalmNormal` and `Direction` vectors define the orientation of the hand.

Figure 1 Hand Parameters from Leap Motion Controller (Developer API)

Fingers

The Leap Motion controller provides information about each finger on a hand. If all or part of a finger is not visible, the finger characteristics are estimated based on recent observations and the anatomical model of the hand. Fingers are identified by type name, i.e. *thumb*, *index*, *middle*, *ring*, and *pinky*.

Fingers are represented by the `Finger` class, which is a kind of `Pointable` object.



Finger `|Finger_tipPosition|` and `|Finger_direction|` vectors provide the position of a finger tip and the general direction in which a finger is pointing.

Figure 2 Finger Parameters from Leap Motion Controller (Developer API)

The following movement patterns are recognized by the Leap Motion software:

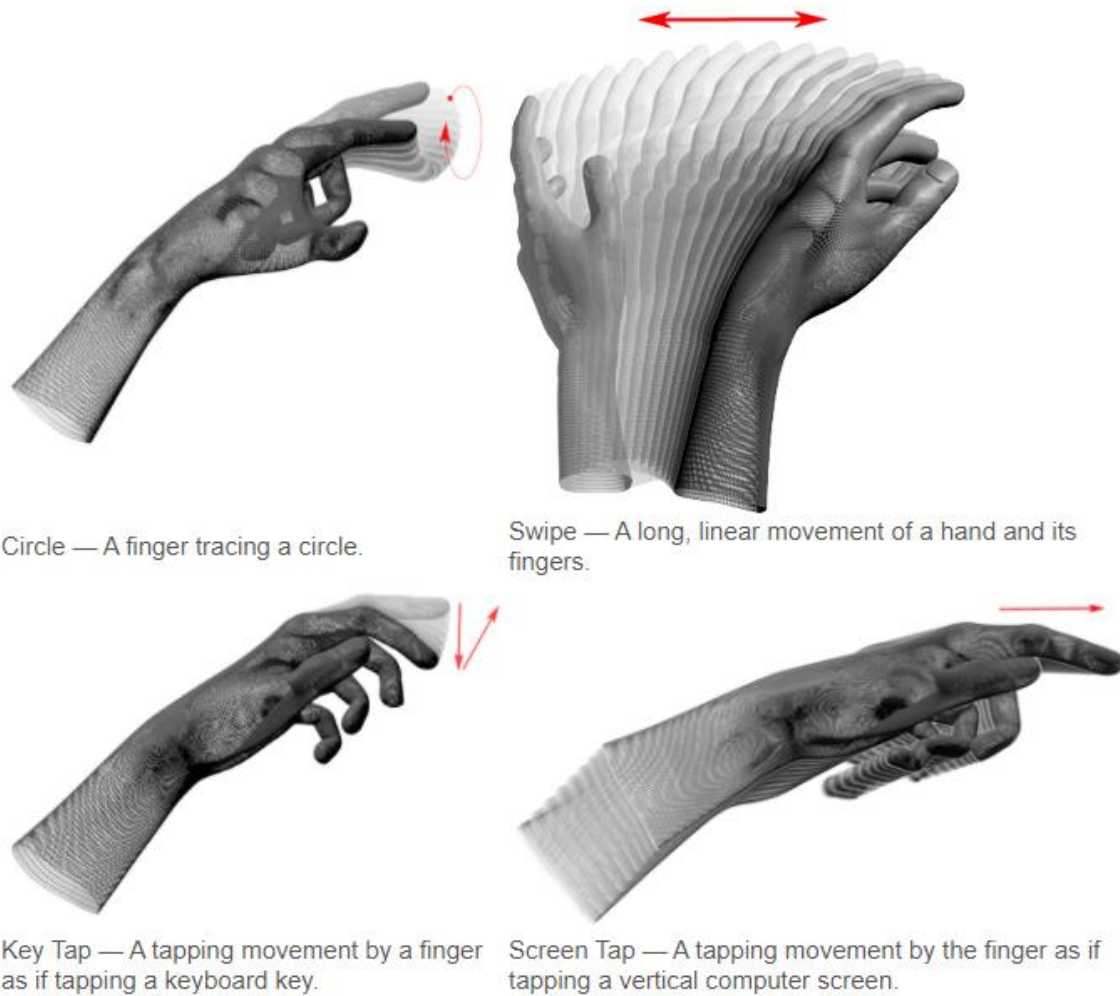
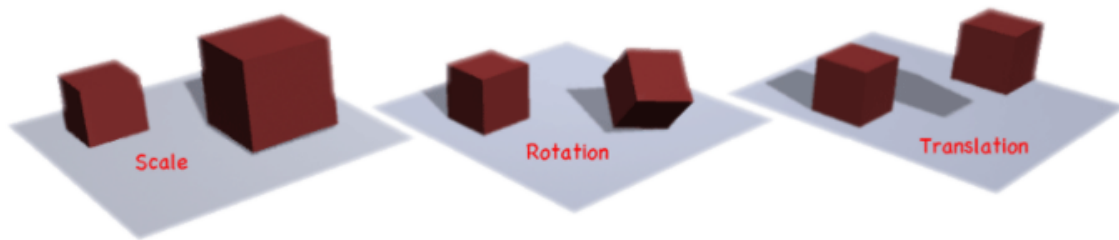


Figure 3 Movement Patterns from Leap Motion Controller (Developer API)

Motions

Motions are estimates of the basic types of movements inherent in the change of a user's hands over a period of time. Motions include scale, rotation, and translation (change in position).



Motions are computed between two frames. You can get the motion factors for the scene as a whole from a [Frame](#) object. You can also get factors associated with a single hand from a [Hand](#) object.

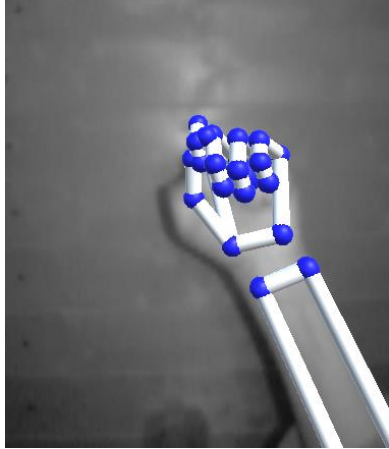
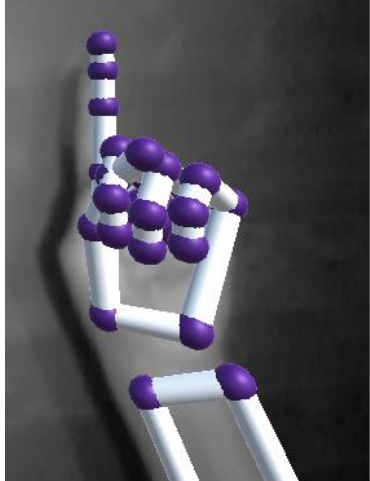
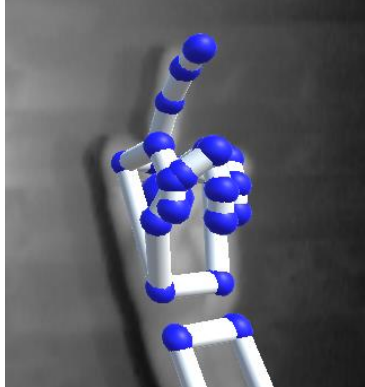
You can use the reported motion factors to design interactions within your application. For example, instead of tracking the change in position of individual fingers across several frames of data, you could use the scale factor computed between two frames to let the user change the size of an object.

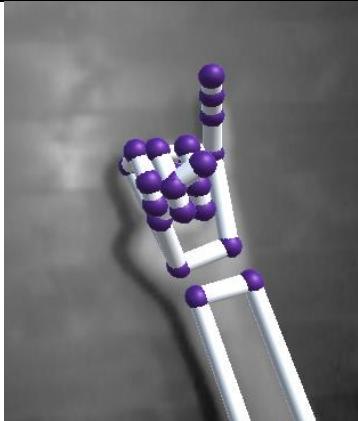
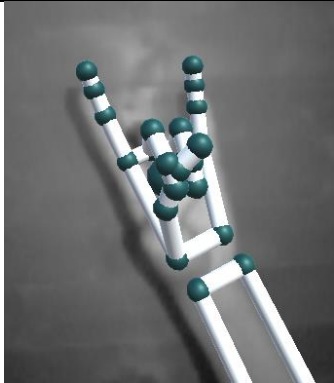
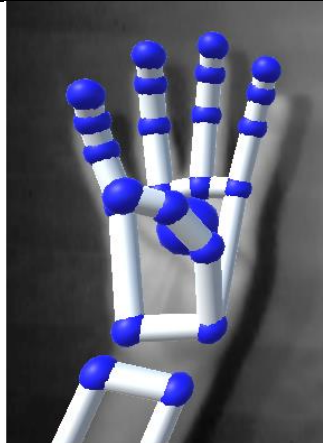
Motion Type	Frame	Hand
Scale	Frame scaling reflects the motion of scene objects toward or away from each other. For example, one hand moves closer to the other.	Hand scaling reflects the change in finger spread.
Rotation	Frame rotation reflects differential movement of objects within the scene. For example, one hand up and the other down.	Hand rotation reflects change in the orientation of a single hand.
Translation	Frame translation reflects the average change in position of all objects in the scene. For example, both hands move to the left, up, or forward.	Hand translation reflects the change in position of that hand.

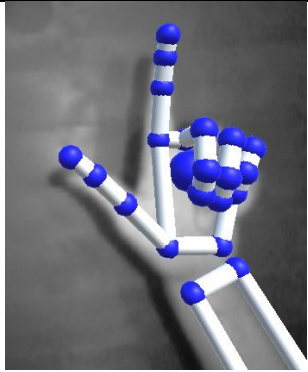
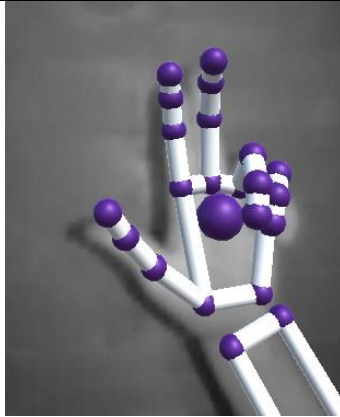
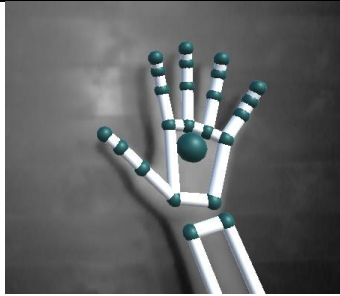
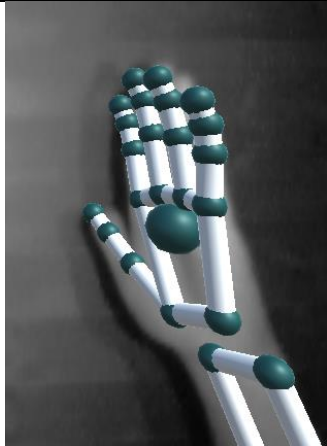
Figure 4 Motions from Leap Motion Controller (Developer API)

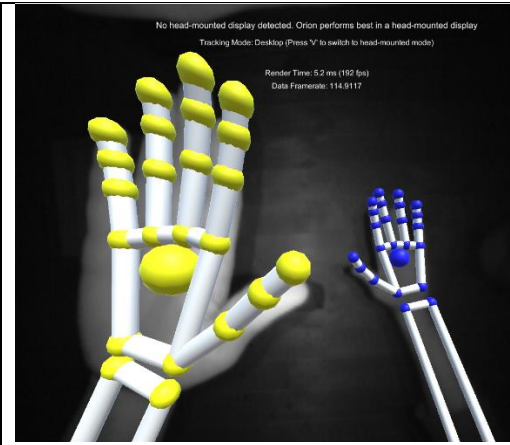
Appendix B

My programmed gestures and what they trigger

Left / Right Hand Gesture	Name	Triggers
	Grab	STOP ALL TRACKS
	Finger 1 (Index)	Isolation
	Finger 2 (Middle)	vaccination

		Finger 4 (Pinky)	Symptomatic
		Finger 1&2	Quarantine
		Finger 1&4	Truth
		Fingers 1,2,3&4	Totalitarian

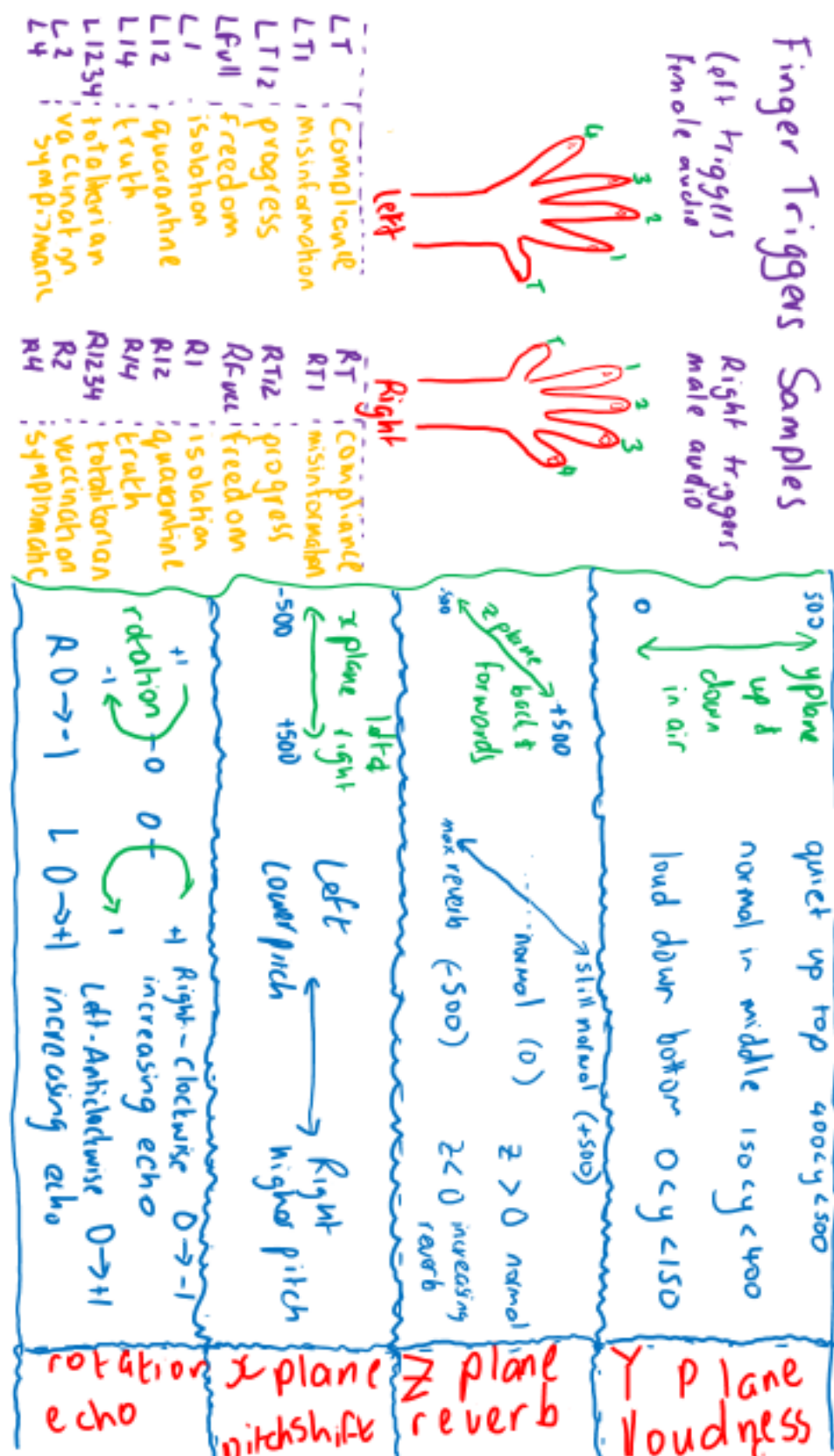
		Thumb & Finger 1	Misinformation
		Thumb, Finger 1 & 2	Progress
		Full Hand	Freedom
		Rotation	Increases echo



Both up and low on
y axis

Not triggering
anything, just visually
showing you that it's
possible to pick up
both hands
successfully.

Overview of what my instrument does



Appendix D

Source	Words									
The Guardian	pandemic	Herd immunity	Lockdown	zoom	Megxit	Antifa	BLM	Karen	Simp	
	Cusp	Hindsight	sclerotic	Override	turbocharge	loony	ovation	singularly	antivax	
Merriem Webster	extremism	recall	Hate crime	Vaccine	unwitting	Scorched earth				
	covid	facemask	progress	truth	Social distancing	Trade war	sustainability	Flatten the curve	Lockdown	
The Irish Times	Climate change	unemployment	outbreak	symptoms	trump	migrants	quarantine	progressive	Identity politics	
	Hand sanitiser	Fake news	Self isolate	New normal	wuhan	stimulus	Work at home	coronavirus		

Appendix E: Beta Testing Feedback

Constructive Criticism requests from peers after using beta version of sensor:

Feedback 1: "I struggled to keep my third finger up independently and even combined with the second one, only could extend it properly with all four other fingers, not sure if it's just me, but maybe not have a third digit trigger"

Feedback 2: "It kept cutting off the previous word when I triggered a new one, would be better if you could let the old word finish first"

Feedback 3: "The pitch shift is a bit extreme, maybe make a more gradual pitch shift to stop it sounding squeaky and comedic when it's quite a moody atmospheric instrument"

Feedback 4: "This is a sick concept, would be awesome if it was easy to insert my own audio samples if I wanted to use it to perform as I'm not familiar with max so would be hard for me to program my own ones in without potentially screwing it up"

Feedback 5: "This is a really innovative idea for live performance, maybe to take it to the next level you could use max to output a graphic visual of the sound manipulations, really blow people's minds when we are all finally allowed out to venues to watch live music."

Bibliography

Aroca, R. et al (2012) "Method for Reading Sensors and Controlling Actuators Using Audio Interfaces of Mobile Devices" *Sensors* 12(2) pp. 1572-1593. Available at: <https://doi.org/10.3390/s120201572> (Accessed 24th Feb 2021)

Chen, J. et al (2017) 'Multiple sensors based hand motion recognition using adaptive directed acyclic graph', *Applied Sciences*, 7(4) pp. 358

Damn Deal Done (2014) *Leap Motion – Theremin (Max/MSP) – Music Technology Gestural Instrument*. Available at: <https://www.youtube.com/watch?v=6S86PbXtCNM&list=PLE-gauBqVR8HwYcyCL3TFZcwKRjaEJ49d>. (Accessed 20 March 2021)

Developer Archive (2021) API Overview Leap Motion. Available at: https://developer-archive.leapmotion.com/documentation/v2/csharp/devguide/Leap_Overview.html (Accessed 20 March 2021)

Gibson, S. and Love, J. (2011) 'Advanced Media Control through Drawing: Using a Graphics Tablet to Control a Complex Audio and Video Data in Live context' Available at: <https://core.ac.uk/download/pdf/5901215.pdf> (Accessed 10 March 2021)

Gratoo (2013) *The Geek Tune (Rocking the Leap Motion)* #Original by Gratoo

Available at: <https://www.youtube.com/watch?v=xp6v96DhzwU>. (Accessed 2 March 2021)

Guzsvinecz, T., Szucs, V., & Sik-Lanyi, C. (2019). 'Suitability of the Kinect Sensor and Leap Motion Controller-A Literature Review'. *Sensors* 19(5) pp. 1072.

<https://doi.org/10.3390/s19051072>

Han, J. & Gold, N. (2014) "Lessons Learned in Exploring the Leap Motion™ Sensor for Gesture-based Instrument Design" Available at:

<https://core.ac.uk/download/pdf/21617813.pdf> (Accessed 15 March 2021)

Lacey, L. (2020). '5 of the Best Platforms to Make Music Software'. *Mac Pro Video*.

24 May. Available at: <https://www.macprovideo.com/article/audio-hardware/5-of-the-best-platforms-to-make-music-software> (Accessed 26 Feb 2021)

Lee, B, Lee, D, Chin, S, (2017) 'Structural Motion Grammar for Universal Use of Leap Motion: Amusement and Functional Contents Focused' *Hindawi Journal of Sensors*. Volume 2018, pp. 1 - 15 Available at:

[//core.ac.uk/download/pdf/206713433.pdf](https://core.ac.uk/download/pdf/206713433.pdf) (Accessed 10 March 2020)

Linehan. H. (2020). 'What's the top word of 2020? Oh you guessed that one easily' *The Irish Times*. 7 July. Available at: <https://www.irishtimes.com/culture/what-s-the-top-word-of-2020-oh-you-guessed-that-one-easily-1.4298315> (Accessed 15 March 2021)

McCartney, R., Yuan, J., Bischof, H. P. (2015) Gesture Recognition with the Leap Motion Controller. International Conference on Computer Vision and Pattern Recognition. Available at: <https://scholarworks.rit.edu/cgi/viewcontent.cgi?referer=https://www.google.com/&httpsredir=1&article=1877&context=other>. (Accessed 12 March 2021)

Merriam-Webster (2021) 'The words of the week -1/1/21' *Words at Play*. Available at: <https://www.merriam-webster.com/words-at-play/the-words-of-the-week-january-01-2021> (Accessed 15 March 2021)

Merriam-Webster (2021) 'The words of the week -2/5/21' Available at: <https://www.merriam-webster.com/words-at-play/the-words-of-the-week-february-05-2021> (Accessed 15 March 2021)

Mueller. J, (2014) 'Tools to Help You Write Apps That Use Sensors.' *SmartBear* Available at: <https://smartbear.com/blog/tools-to-help-you-write-apps-that-use-sensors/> (Accessed 28 Feb 2021)

Nico Earnshaw (2014) *Drums + Leap Motion Theremin experiment*. Available at:
<https://www.youtube.com/watch?v=4ZG9V0MNeh4> (Accessed 8 March 2021)

Phillips, M. (2012) "Max/MSP/Jitter For Music: A Practical Guide To Developing Interactive Music Systems For Education And More". *Journal of the Art of Record Productions* 1(06)

Rerunnermusic (2009) *Max for Live – the Wacom Tablet Theremin – Using Max MSP in Ableton Live, with Akai APC40* Available at:
<https://www.youtube.com/watch?v=V5FiC563GgM> (Accessed 10 March 2021)

Resch, T. (2013) "note~ for Max – An extension for media arts & music". In: *Proceedings of the e international conference on new interfaces for musical expression*, Daejeon, Republic of Korea, pp. 210-212

Reveillac, J.M. (2018) 'Introduction to MAX/MSP' *Musical Sound Effects Analog and Digital Sound Processing*, 1. pp. 209 -311. Available at:
<https://onlinelibrary.wiley.com/doi/pdf/10.1002/9781119467014.app3> (Accessed 26 Feb 2021)

Ryo Fujimoto Youtube Channel (2013) *Humanelectro x Leap Motion* Available at:
<https://www.youtube.com/watch?v=p9EyDes62sc> (Accessed March 2 2021)

Shariatmadari, D. (2020) 'Pandemic, lockdown and Megxit: the most influential words of 2020'. *The Guardian*. 26 November. Available at: <https://www.theguardian.com/books/2020/nov/26/doomscrolling-in-lockdown-the-most-influential-words-of-2020> (Accessed 15 March 2021)

Smith, J. (2008) 'MAX/MSP A Software Tool for Percussionists'. *Arizona State University* pp. 64-69. Available at: http://www.public.asu.edu/~idjbs/max_msp.pdf (Accessed 26 Feb 2021)

UKEssays. (2018) '*Electronic Music Origin And Effects Music Essay*.' [online]. Available from: <https://www.ukessays.com/essays/music/electronic-music-origin-and-effects-music-essay.php?vref=1> [Accessed 12 March 2021].

Virtual DJ (2021) Available at: <https://www.telebody.ws/VirtualDJ/sub%20index.html>

Wright, M. 2021. "Session 3: What's Really going on inside Max" *Programming Max: Structuring Interactive Software for Digital Arts* Stanford University