

Name: James Burns
Student Number: 2106300
MA Commercial Music Production
Module Title: Development ADAD7005
Assignment: Element 2b Essay

Introduction

The purpose of this essay is to demonstrate the development and research surrounding the creation of my instrument, RV's Warped Quartet. I will detail the process of creating the instrument, alongside my considerations for the graphic user interface (GUI), and my response to feedback during the developmental stage. I will also compare the instrument to similar products on the market and detail potential future changes.

Overview of RV's Warped Quartet

RV's Warped Quartet is a Kontakt library consisting of a string quartet, recorded through a Sanyo cassette player, and then processed and manipulated using a modular synthesiser. The user can navigate the four velocity layers to produce either a low fidelity warped sound with frequent audio drop-outs, or a distorted, rich, string quartet with synthesised overtones, allowing them to quickly adjust the sound without having to move any knobs or switches.

Should the user wish to gain further control over the sounds they produce, I have coded a variety of interactable parameters using Kontakt script. Beginning with an attack and release knob, this allows the user to control the envelope of the instrument, giving them the opportunity to create swelling pads with long tails, staccato stabs, or anything in between.

The user can toggle three buttons in order to activate different effects. Influenced by the Roland Juno-60 (Reid, 2021), I installed a button that toggles a chorus effect, the user cannot tweak the rate or wet/dry levels as I wanted to keep the instrument's interactivity as simple as possible.

Below the chorus is the degrade button, this is a compressor and overdrive unit that adds further grit and distortion to the instrument. The Crunch knob controls the wet/dry level of the degrade, allowing the user to blend in the level of distortion without it being too overbearing on the overall sound.

The reverb button toggles a bright room reverb, giving width and space to the string sound whilst also making the warped quartet sound more akin to a clean, traditional string plugin. The reverb knob controls the wet/dry send as I believed this gave the user the most control over the sound whilst still maintaining the simplicity of one knob.

Finally, the cutoff control is a low-pass filter with a slight increase in resonance. Allowing the user to create dramatic filter sweeps, and giving the user the option to filter the strings, thus making them sound closer to an electronic synthesiser instead of string samples. The cutoff is also mapped to the mod-wheel of a keyboard, allowing the user to have more versatility and ease

whilst playing with RV's Warped Quartet.

Instrument Development

A frequent part of my musical process is manipulating physical instruments and turning them into unique sounds using a variety of destructive processes. I wanted a new development on the countless clean and polished sample libraries of string quartets, as I knew I couldn't reach the same quality of Spitfire's Sacconi Quartet (Spitfire, 2015), I decided I wanted to make a string library that was unique and captured the sounds that I love.

The instrument began with engineering a string quartet session at Tileyard Education's Commercial Studio. I used an AEA R88 ribbon microphone in the centre of the room as "ribbons have a suitably wide frequency range and the fast transient response necessary to capture the very complex sound of an orchestra." (Robjohns, 2006)". Below the R88 was a pair of Neumann KM184's in an ORTF positioning whereby "The mics are spaced 17 cm (about 6 3/4") apart, at an angle of 110 degrees" (Sweetwater, 1997), thus creating an ideal stereo capturing of the quartet. I placed two Schoep's Colette MK2 omnidirectional microphones in a spaced pair on either side of the room to further increase the stereo capturing. Finally, I pointed a KM184 at each violin with a distance of two feet and a pair of Neumann U87 microphones on the Cello and Viola, thus resulting in a clear recording of each individual instrument for additional reinforcement when mixing. I chose condenser microphones over dynamic microphones for their accuracy, sensitivity, and detail in capturing the subtle sounds of a string quartet. (see picture of setup below)



Neubert states that "The primary reason why composers chose the orchestra instead of individual instruments in the early stages of developing the instrumental tape medium is that the timbral

content of the orchestra contained enough diverse timbral choices to be readily combined with complex electronic effects.” (1982 p.542-543). I wanted my instrument to feature the whole frequency spectrum, through the low frequency tones of a cello, and the delicate treble of a violin, hence why I recorded the quartet as one collective sound. After getting the microphones and levels setup, I recorded the quartet playing each note for three seconds in a three octave range, from a C1 to a C4.

After recording the quartet, I placed the stems of the session into Ableton and began balancing the mix. I consciously decided against using any processing aside from EQ and panning as I wanted to do this on cassette and modular synthesisers. I soon came across an issue as I found out the cassette recorders I owned are only mono compatible, initially I attempted to record the left channel and right channel of the cassette separately and then sync them together in my DAW (digital audio workstation), however due to the unpredictability of the playback speed on my cassette players, this proved to be a problem with both pitch and timing, therefore I decided to record in mono.

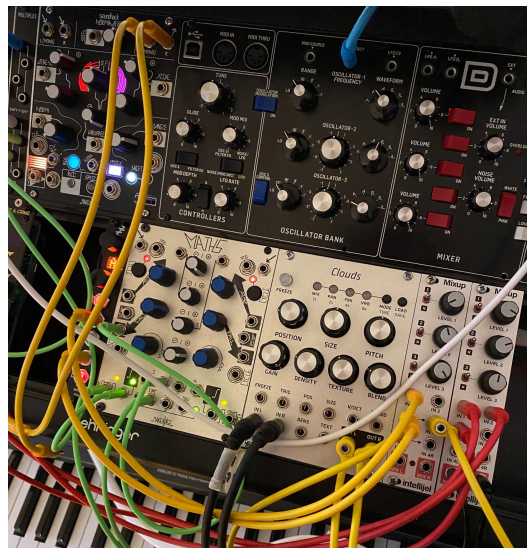
As my cassette player is circuit bent with a 10k potentiometer (see below image), I carried out a variety of experiments whereby I recorded a note, and then decreased the playback speed to half, resulting in a longer, lo-fi and octave-lower recording of the original sound, however i believed this dramatically changed the source sound, losing the qualities that made it sound like a string instrument. Therefore I finally recorded each of the notes onto a mono cassette with the playback set to the original speed of 3.75 inches per second, I added a low shelf frequency boost up to 250Hz prior to recording to increase the cello’s lower presence as “Cassettes, on the other hand, start to attenuate bass tones below about 40 Hz, so instruments with ultra-deep notes—such as grand piano, large kick drums, and synth bass—lose some of their sonic power.” (Butterworth, 2021), thus reducing the loss of lower frequencies in the final recording.



I originally planned to send the sounds from the cassette player directly into my modular synthesiser, however the short spacing between the individual notes was causing issues with the delay and reverb tail of the synth, therefore I had to record the cassette player into Pro Tools,

using the tab-to-transient and cut feature to precisely extract each note, and finally, I added ten seconds of silence between the notes.

The final stage of processing involved me using my modular synthesiser. For context, a modular synthesiser is an instrument made up of numerous modules, each with their own function, “leaving it up to the user to arrange and connect them however they want to create their own unique synthesizer sounds and behaviours.” (Intellijel, 2022). My patch for the instrument was largely based around the Mutable Instruments’ Clouds, “Clouds is a granular audio processor like no other. Clouds is focused on the realtime granularization of incoming audio signals, and their transformation into amorphous textures.” (Mutable Instruments, 2022), I used the Clouds to add an additional tone an octave above the original note alongside adding additional audio dropouts to the original signal. I sent the cassette notes through the Clouds four times, reducing the amount of signal dropouts, and increasing the presence of the additional tone each time. This acts as the different velocity layers for the sample instrument as the instrument ranges from a broken, distorted recording of the strings, to a smoother, richer, and dense recording. Finally, I took these recordings of four layers of the four octaves and bounced a recording of each separate note (as a 48kHz, 24 bit, .wav file) resulting in a folder of 144 individual notes. I then carefully labelled which velocity layer and which pitch each note belonged to.



Kontakt development and comparison to competitors

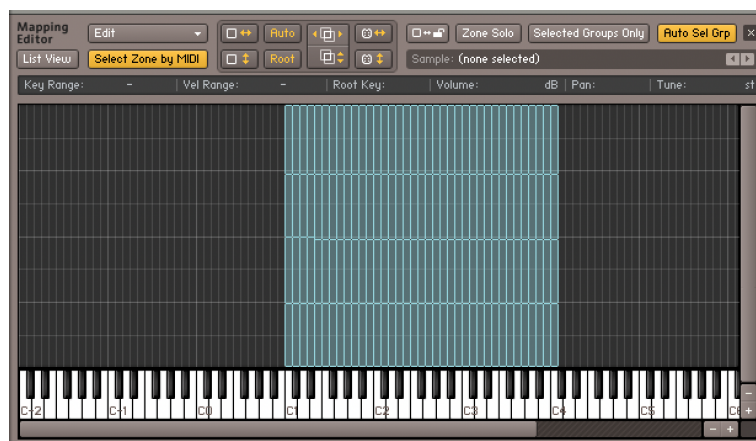
The next stage of development involved loading the instrument into Native Instrument’s Kontakt sampler, “Kontakt ...has since become a major player amongst software samplers. To recap briefly, Kontakt is a sophisticated software sampler, offering sample creation from compatible file formats, sample manipulation and performance capabilities similar to programs such as Tascam's Gigastudio, Steinberg's Halion and Emagic/Apples EXS24.” (Magnus, 2005). Since its

release, Kontakt has become the industry standard for the development and release of sample libraries, due to its compatibility with most DAWs and large user base.

I explored a variety of Kontakt's competitors beginning with Logic Pro X's sampler. Logic's sampler is free (excusing the £175 for Logic) compared to Kontakt's £359 price point, and offers most of the same features as Kontakt with lower CPU usage, however it is only compatible with Logic thus restricting any usage across other DAW's, and it lacks a scripting language similar to Kontakt script, thus reducing the customizability of the users instrument.

Another competitor is HISE, "HISE is an open source framework for building sample based virtual instruments." (HISE, 2019). HISE is free, open source (allowing users to add new, and more advanced functions than those available in Kontakt if the user knows C++) and doesn't charge for users to release sample libraries (unlike the \$1000 fee Kontakt can charge if the creator wishes to licence their instrument for the free Kontakt player) thus making it a strong competitor to Kontakt. However, HISE is still a relatively new software, thus meaning it doesn't have the same market size as Kontakt, and those who have already purchased multiple Kontakt libraries may be reluctant to switch to HISE. Alongside this, from my research and brief exploration of HISE, it is a far more complex operation to develop and release a library compared to Kontakt "If you're a Kontakt developer thinking about switching to HISE and you have limited knowledge of writing software applications then you're in for an adventure" (d.healey, 2022), therefore I decided to develop using Kontakt.

After bouncing each of the samples and labelling them with the note and velocity layer, I had the simple task of assigning the samples to their relative keys. For the velocity layers, I knew the upper limit for MIDI is 127, therefore I rounded this up to 128 and divided it by 4 and placed each velocity layer every 32 velocity values (e.g, layer 1 is between 1 and 31, layer 2 is between 32 and 63, etc). This gave a well balanced response when I tested playing back the samples.



After, I played back each sample and added their looping point, thus allowing the notes to be held for as long as desired without the volume decreasing. Alongside this, I added a mod so that

the volume is controlled by the velocity of the note, giving the player further control over the sound.

Following this, I added the different instrument insert effects. I chose this over group insert fx as it would affect each track in a polyphonic way instead of adding an effect to each note, thus minimising the CPU usage. I also chose insert over sending to an auxiliary channel as I wanted the different effects to run through each other (for example, the distortion will pass through the reverb as opposed to the signal being sent to the two effects separately), thus giving the user more creative options for sound design.

On the effects chain, I began with a cutoff filter (ladder low-pass filter) with a slight boost in resonance thus exaggerating the effect when the user does a filter sweep. Following this, I added the ‘SuperchargerGT’ effect which acts as a saturator and tube compressor, this compresses the audio and enhances the cassette’s hiss and distortion, whilst also saturating the sound. The chorus effect is applied after the distortion, the slow rate and maximum depth results in a thicker sound. Finally, a room reverb is applied to add stereo width and space to the instrument.



GUI (Graphical User Interface)

For the GUI I had to familiarise myself with Kontakt script, ‘The KSP (Kontakt Script Processor) is a powerful feature in Kontakt that lets developers or users build custom instruments by addressing and manipulating the internal workings of Kontakt using a dedicated scripting language.’ (Pitman, 2012). This proved to be particularly useful when developing my GUI.

```
declare ui_knob $ReverbWetDry (0,1000000,1)
set_text($ReverbWetDry, "")
hide_part($ReverbWetDry,$HIDE_PART_BG .or.$HIDE_PART_VALUE)
move_control_px($ReverbWetDry, 400, 80)
```

To begin, I had to create these initial lines of code within the ‘on init’ section of my code, this means this section of the script will run on initialisation (when the instrument is first loaded).

‘declare ui_knob \$ReverbWetDry’ creates the variable of the knob and labels it ‘ReverbWetDry’, the numbers following this mean the knob will go from 0, when turned to the left, to 1,000,000, when turned to the right. I set the text accompanying the ‘ReverbWetDry’ variable to “”, meaning it will be blank in the preview window. On line 3, I hide the grey background that normally displays with the knobs, and I remove the value box as well for a clearer and minimalistic GUI. Finally, ‘move_control_px’ is where the knob will be displayed on the GUI. I repeated this code, creating a variable for the cutoff filter’s frequency, the wet/dry of the SuperGT distortion, and the attack/release of the instrument. I followed this text with ‘make_persistent’ for each variable, meaning when the user recalls a session, the knobs will remain in the same position of which they left them. Finally I added ‘end on’, closing the section of the script that occurs on initialisation.

```
on ui_control($ReverbWetDry)
    set_engine_par($ENGINE_PAR_SEND_EFFECT_OUTPUT_GAIN, $ReverbWetDry,-1,3,1)
end on

on ui_control($Saturation)
    set_engine_par($ENGINE_PAR_SUPERGT_MIX,$saturation,-1,1,1)
end on

on ui_control($cutoff)
    set_engine_par($ENGINE_PAR_CUTOFF, $cutoff,-1,0,1)
end on
```

In the following part of the script, I added sections of the code that update the parameters of the effects upon their movement (as seen in the image above).

Returning to the ‘on init’ part of the script, I added buttons to my GUI so that the user could toggle the bypass of the instrument. Beginning with ‘declare ui_switch \$ReverbButton’, I declared the switch and assigned it to the ‘ReverbButton’ variable. Similar to the knobs, I removed the text, changed the positioning, and adjusted the size of the knobs so they’re better suited to my GUI. (See appendix A for full code)

For the artwork, I researched a variety of pre-made Kontakt instruments and examined their art style. The majority were either realistic images of instruments (such as pianos, violins, etc) or cartoon drawings. I decided to lean into the ‘warped cassette’ approach, by drawing a cassette in procreate, loading it into Adobe Photoshop, and then bending and morphing the shape until it looked ‘warped’. I chose a simple colour scheme of #982713 (Red) #d2c683 (White) #8198cf (Blue), as I believed the simplicity would complement the rest of the GUI as “When designing a UI, it’s recommended that the interface is designed to be predictable, concise and consistent.” (Berezhnoi, 2019). I labelled the cassette after the instrument and added text with blank spaces underneath for the knobs. I cropped the image down to fit the Kontakt window’s sizing (632 x 272px) and was happy with the result.

Prototype & Beta Testing

Prior to the development of my instrument, I created a prototype of RV's Warped Quartet in Ableton's sampler. Due to the ability to quickly mock-up a rough version of the instrument, I used this to experiment with different effects and receive feedback from users.



The samples for the prototype were recorded louder to the cassette resulting in further tape distortion, and there weren't any velocity layers at this stage. Alongside this, I added the cutoff, reverb, echo, high pass filter, and drive knobs whilst experimenting with different effects. I then reached out to a handful of users to test this version of the instrument for feedback (See appendix B). I amended the instrument according to this feedback by: re-recording the strings to tape to lower the distortion, removing the high-pass-filter, adding a chorus button, and adding additional octaves and velocity layers.

Following the development of my instrument in Kontakt, I returned to the same users for beta testing and additional feedback and testing for any errors (See appendix C). One of the users reported that the chorus rate was too high, therefore I adjusted the rate to a much slower 1.1Hz. Another issue was the volume level, as this user wasn't familiar with Kontakt, they didn't know that you could adjust the level within Kontakt therefore I didn't install an additional volume knob. Finally, two of the beta testers complained that they didn't use or own Kontakt, this is a problem that would involve recreating the instrument in Ableton or Logic's sampler, therefore I will potentially use this as a future development.

Comparison to Commercial Competitors

The market for string based sample instruments is both competitive and dominated by teams of developers with large budgets to record renowned quartets in world famous studios. With this information in mind, I knew that I had to approach my instrument from a different perspective by offering something with a unique selling point. In order to prepare for this, I explored popular sample instruments and carried out a SWOT analysis on them. A SWOT analysis is "the identification and assessment of strengths, weaknesses, opportunities, and there to yield strategic

insights.” (Valentin, 2001, pp. 54), this allowed me to see where my competitors succeed and where their weaknesses are, so that I can adapt my instrument accordingly.

One of the most popular quartet libraries is Spitfire Audio’s Sacconi String Quartet. Recorded with the renowned Sacconi Quartet in London’s Wigmore Hall, Spitfire mentions “We offer you six microphone positions and two mixes for ultimate control, with easy-to-use playable patches, and four types of true legato.” (Spitfire, 2015) showing they pride themselves on customizability whilst still being user friendly. Due to my desire to keep my GUI simplistic and easy to use, RV’s Warped Quartet has fewer parameters to adjust, thus somewhat limiting the user’s ability to shape parameters of the sound (such as microphone position, vibrato, and legato.) This could potentially be an aspect that I would add in future versions of the instrument.

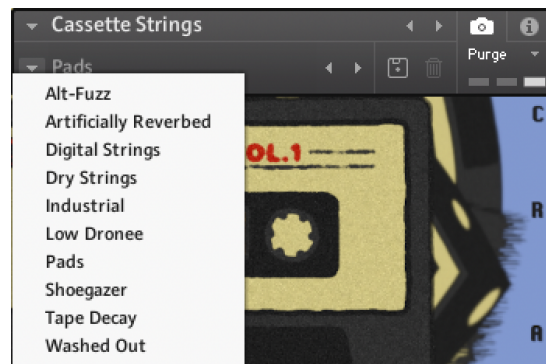
Whilst Sacconi String Quartet boasts its numerous parameters, creatively it is limiting and not particularly unique from other string quartet libraries, therefore it may struggle to stand out amongst other string quartets on the market. This informed my decision to explore the use of cassette tape on strings and the additional application of modular synthesisers.

After exploring string libraries, I wanted to find instruments that involved cassettes and lo-fi sounds. I soon discovered Peter.Flint’s Relic Keyboard, an independently made sample instrument, where he recorded “ a selection of the sustained sounds of the keyboard onto microcassette, and trying out the vocal sampler that sets this series of keyboards apart. The results are distant, distorted, and imperfect” (Peter.Flint, 2022). To begin with, this instrument doesn’t have velocity layers thus limiting the performability of the instrument. RV’s Warped Quartet has four separate velocity layers, thus making it useful for when the user wants quick dynamic changes or a different sound without tweaking the settings. Alongside this, whilst the unique selling point of Relic Keyboard is the lo-fi sound, the user doesn’t have any control over adding more or less noise thus restricting the user’s ability to shape the sounds. My instrument has an additional knob, allowing the user to add further distortion and noise should they choose to do so, resulting in greater creative control.

Relic Keyboard’s minimalistic GUI is similar to RV’s Warped Quartet, preferring to offer a simplistic and user friendly interface compared to something more daunting with plenty of menu diving such as Spitfire’s Sacconi String Quartet. Relic Keyboard’s interface does offer control over the decay and sustain of the instrument’s envelope, this could potentially be something I would adapt for my instrument in the future as it gives the user further control over the envelope of the instrument and could result in the shaping of short plucked sounds or a slow decay could enhance early transients in the sample.

Presets can be found on both Relic Keyboard and RV’s Warped Quartet, this allows the user to quickly recall new sounds and access pre-made settings. Relic Keyboard’s presets will load an

entirely new bank of samples, ranging from flutes and choirs, to pianos and organs, thus giving a much wider range of sounds for the user to choose from. RV's Warped Quartet's different presets (see below) will alter the various parameters on the instrument whilst still keeping the same samples, thus showing the drastic customizability from the various knobs within my instrument. The opportunity to load pre-made collections of samples is a potential future development, allowing me to offer additional sounds such as a warped piano and a cassette xylophone, to further broaden the timbres available from my instrument.



Conclusion

In conclusion, I'm content with the progress of RV's Warped Quartet. User feedback and my own frequent testing has allowed me to refine and develop the instrument to it's current form. The range of effects and simplistic interface allows the user to have plenty of customisable options to shape the sound. The cassette and eurorack processing gave me an interesting foundation to build upon to create a truly unique sample library within the string library market.

The incorporation of a round robin feature on the instrument would be a potential development. A round robin "enables us to assign two or more samples to a single key before playing them back either sequentially or in random order" (Computer Music, 2015), thus making each performance with the instrument unique, and reducing the 'machine gun' effect whereby a sample sounds fake when repeatedly played.

Another potential development would be using a convolution reverb combined with an impulse response from my modular synth. Sterne mentions, "An 'impulse response' (or IR), is created by introducing a short burst of sound in a room and recording the echo" (2015, p.124). By taking an impulse response of a manipulated reverb from my eurorack, I could further add to the 'warped' element of my instrument. The use of the reverb's wet/dry knob would allow the user to control the level of the warped reverb, thus adding further depth and uniqueness to my instrument.

As mentioned prior, during my beta testing many users complained about the instrument being only available through Kontakt, therefore another future development would be recreating the

instrument within other samplers. By recreating the instrument within Ableton's Sampler, Logic Pro X's Sampler, and HISE, I should have full coverage across all DAW's, allowing me to maximise my audience and userbase.

To summarise, I have successfully created a polyphonic instrument of a string quartet, processed through cassette, and modular synthesisers. The instrument features 4 velocity layers, a chorus, reverb, and degrade effect, alongside an attack and release knob. The GUI is simplistic yet effective, and the instrument features a number of presets.

Bibliography

Berezhnoi, R (2019) 'What Is UI Design And Why Is It Important?', *F5 Studio* , 12 April. Available at: <https://f5-studio.com/articles/what-is-user-interface-design-and-why-is-it-important/> (Accessed: 23 March 2022)

Butterworth, B. (2021) 'Don't Call It a Comeback: Cassettes Have Sounded Lousy for Years (And Still Do!)', *New York Times*, 26 October. Available at: <https://www.nytimes.com/wirecutter/blog/cassettes-comeback-sound-lousy/> (Accessed: 16 March 2022)

Computer Music (2015) 'How to avoid the machine gun effect using round-robin sampling', *musicradar*, 29 December. Available at: <https://www.musicradar.com/tuition/tech/how-to-avoid-the-machine-gun-effect-using-round-robin-sampling-632302> (Accessed 23 March 2022)

d.healey (2022) For Those of Us Considering Jumping From Kontakt To HISE. *VI Control*. 12 February. Available at: <https://vi-control.net/community/threads/for-those-of-us-considering-jumping-from-kontakt-to-hise.90765/page-6> (Accessed: 22 March 2022)

HISE. (2019) *Introduction* Available at: <https://docs.hise.audio/introduction/index.html> (Accessed 17 March 2022)

Intellijel (2022) *Eurorack 101* Available at: <https://intellijel.com/support/eurorack-101/> (Accessed 17 March 2022)

Magnus, N. (2005) 'Native instrument's Kontakt 2', *Sound on Sound*, July. Available at: <https://www.soundonsound.com/reviews/native-instruments-kontakt-2> (Accessed: 17 March 2022)

Mutable Instruments. (2022) *Mutable Instruments | Clouds* Available at: <https://mutable-instruments.net/modules/clouds/> (Accessed 17 March 2022)

Neubert, D. (1982). 'Electronic Bowed String Works: Some Observations on Trends and Developments in the Instrumental/Electronic Medium.' *Perspectives of New Music*, 21(1), pp-542-543

Peter.Flint (2022) *Relic Keyboard*. Available at:

<https://www.pianobook.co.uk/packs/relic-keyboard/> (Accessed: 23 March 2022)

Pitman, T (2012) 'Introduction To Scripting in Kontakt, Part 1', *macProVideo.com*, 1 October. Available at:

<https://www.macprovideo.com/article/audio/introduction-to-scripting-in-kontakt-part-1>

Reid, G (2021) 'Roland Juno 60', *Sound On Sound*, May. Available at:

<https://www.soundonsound.com/reviews/roland-juno-60> (Accessed: 29 March 2022)

Robjohns, H. (2006) 'Recording A String Section', *Sound on Sound*, May. Available at:

<https://www.soundonsound.com/techniques/recording-string-section> (Accessed: 16 March 2022)

Spitfire (2015) *Spitfire Audio - Sacconi Strings Quartet*. Available at:

<https://www.spitfireaudio.com/shop/a-z/sacconi-strings-quartet/> (Accessed: 23 March 2022)

Sterne, J. (2015) Space within Space: Artificial Reverb and the Detachable Echo. *Grey Room*, 60, 110–131

Sweetwater. (1997) 'What is "ORTF"?', *Sweetwater*, 17 April. Available at:

<https://www.sweetwater.com/insync/ortf/> (Accessed 17th March 2022)

Valentin, E. K. (2001). 'Swot Analysis from a Resource-Based View', *Journal of Marketing Theory and Practice*, 9(2). pp. 54.

Appendix A

My full Kontakt script for my instrument with additional labelling.

on init {when the instrument is first loaded}

```
make_perfvew {make performance view}
set_script_title("Knobs") {created the title of this script}
set_ui_height_px(200) {Sets the user interface's height}
declare $count {creates the 'count' variable}

declare ui_knob $cutoff(0,1000000,1) {declares a knob as 'cutoff' sets range from 0 to 1000000}
set_text($cutoff, "") {makes the knob's text disappear}
hide_part($cutoff,$HIDE_PART_BG .or. $HIDE_PART_VALUE) {hides knob's background and value box}
move_control_px($cutoff, 400, 15) {moves knob on user interface}

declare ui_knob $Saturation (0,1000000,1)
set_text($Saturation, "")
hide_part($Saturation,$HIDE_PART_BG .or. $HIDE_PART_VALUE)
move_control_px($Saturation, 535, 15)

declare ui_knob $ReverbWetDry (0,1000000,1)
set_text($ReverbWetDry, "")
hide_part($ReverbWetDry,$HIDE_PART_BG .or.$HIDE_PART_VALUE)
move_control_px($ReverbWetDry, 400, 80)

declare ui_knob $Attack (0,1000000,1) {declares attack knob}
set_knob_unit($Attack, $KNOB_UNIT_MS) {sets the display's units to be in milliseconds}
set_text($Attack, "") {hides the name of knob in display}
hide_part($Attack,$HIDE_PART_BG ) {hides the background of the knob}
$Attack := get_engine_par($ENGINE_PAR_ATTACK, 0, find_mod(0,"ENV_AHDSR"), -1) {returns the
value of the attack }
move_control_px($Attack, 400, 160) {positioning of attack knob}

declare ui_knob $Release (0,1000000,1)
set_knob_unit($Release, $KNOB_UNIT_MS)
set_text($Release, "")
hide_part($Release,$HIDE_PART_BG )
$Release := get_engine_par($ENGINE_PAR_RELEASE, 0, find_mod(0,"ENV_AHDSR"), -1)
move_control_px($Release, 535, 160)

declare ui_switch $ChorusButton {declares the chorus button switch}
set_text ($ChorusButton,"") {sets text as blank}
declare $ChorusButtonid
$ChorusButtonid:= get_ui_id ($ChorusButton) {gets the ID number of callback}
move_control_px($ChorusButton, 490, 60) {sets button positioning}
set_control_par($ChorusButtonid,$CONTROL_PAR_WIDTH,15) {button width}
set_control_par($ChorusButtonid,$CONTROL_PAR_HEIGHT,15) {button height}

declare ui_switch $DegradeButton
set_text ($DegradeButton,"")
declare $DegradeButtonid
$DegradeButtonid:= get_ui_id ($DegradeButton)
move_control_px($DegradeButton,490, 85)
set_control_par($DegradeButtonid,$CONTROL_PAR_WIDTH,15)
set_control_par($DegradeButtonid,$CONTROL_PAR_HEIGHT,15)

declare ui_switch $ReverbButton
set_text ($ReverbButton,"")
declare $ReverbButtonid
$ReverbButtonid:= get_ui_id ($ReverbButton)
move_control_px($ReverbButton,490, 111)
set_control_par($ReverbButtonid,$CONTROL_PAR_WIDTH,15)
set_control_par($ReverbButtonid,$CONTROL_PAR_HEIGHT,15)

make_persistent($DegradeButton) {makes it so the value remains after closing/reopening}
make_persistent($ChorusButton)
make_persistent($ReverbButton)
make_persistent($Release)
make_persistent($Attack)
make_persistent($ReverbWetDry)
make_persistent($Saturation)
make_persistent($cutoff)
```

```
read_persistent_var($Attack){reloads the value of the variable}
set_knob_label($Attack, get_engine_par_disp($ENGINE_PAR_ATTACK, 0, find_mod(0,"ENV_AHDSR"), -1))
{sets knob value to the attack time}
read_persistent_var($Release)
set_knob_label($Release, get_engine_par_disp($ENGINE_PAR_RELEASE, 0, find_mod(0,"ENV_AHDSR"), -1))

end on {closes the script for upon initialisation}

on ui_control ($Attack) {this section of the script runs when the attack knob is moved}
$count := 0
while ($count < $NUM_GROUPS) {when count value is less than the number of groups}
    set_engine_par($ENGINE_PAR_ATTACK, $Attack, $count, find_mod(0,"ENV_AHDSR"), -1)
{sets the length of the attack time, }
    inc($count)
end while
set_knob_label($Attack, get_engine_par_disp($ENGINE_PAR_ATTACK, 0, 0, -1))
{updates knob label value}
end on

on ui_control ($Release)
$count := 0
while ($count < $NUM_GROUPS)
    set_engine_par($ENGINE_PAR_RELEASE, $Release, $count, find_mod(0,"ENV_AHDSR"), -1)
    inc($count)
end while
set_knob_label($Release, get_engine_par_disp($ENGINE_PAR_RELEASE, 0, 0, -1))
end on

on ui_control ($ReverbWetDry)
    set_engine_par($ENGINE_PAR_SEND_EFFECT_OUTPUT_GAIN, $ReverbWetDry,-1,3,1)
{when reverb knob is moved, set the reverb wet to the value}
end on

on ui_control ($Saturation)
    set_engine_par($ENGINE_PAR_SUPERGT_MIX,$saturation,-1,1,1)
end on

on ui_control($cutoff)
    set_engine_par($ENGINE_PAR_CUTOFF, $cutoff,-1,0,1)
end on

on ui_control ($ChorusButton)
    set_engine_par($ENGINE_PAR_EFFECT_BYPASS,$ChorusButton,-1,2,1)
{when chorus button is pressed, sets chorus bypass to off or on}
end on

on ui_control ($DegradeButton)
    set_engine_par($ENGINE_PAR_EFFECT_BYPASS,$DegradeButton,-1,1,1)
end on

on ui_control ($ReverbButton)
    set_engine_par($ENGINE_PAR_EFFECT_BYPASS,$ReverbButton,-1,3,1)
end on
```

Appendix B

Asking three producers for feedback on the Ableton Prototype of my instrument

NAME	POSITIVES	NEGATIVES
Daniel Miller	'Sounds really unique' 'Really versatile, you can really change the source sound to something new'	Far too noisy HPF seems a bit pointless Velocity layers are needed
Vittorio Tondo	'I like the ability to change the attack and release' 'The cutoff knob is good for performance'	Choose different effects (Chorus, or phaser) Too distorted
George Bone	'I like that you can still hear the string quartet clearly' 'Being able to change the reverb wet/dry is useful'	Only 1 octave is restricting Doesn't use ableton so can't use it

Appendix C

Asking the same producers for feedback after Beta testing my Kontakt Instrument

NAME	POSITIVES	NEGATIVES
Daniel Miller	'Love the <u>colour</u> and artwork' 'The different velocities are really interesting'	Chorus effect sounds weird
Vittorio Tondo	'Sounds much better than the previous version' 'The simplicity makes it easy to play for the first time'	Can't control the overall volume Doesn't have Kontakt so can't use it
George Bone	'Good that it has the multiple octave range' 'The distortion is much more usable on this version'	Doesn't own Kontakt, would like it to be in Logic More customisable options could work better