

Kevin McKenna

Student Number: 1909923

Module: Development

Element 2b Instrument Essay

The aims of the instrument

The aim of this instrument is to be a single voice, multi oscillator drone instrument with an authentic rich textural sound.

Drones by definition “are essentially sustained or continuous tones that establish a tonal centre to accompany melodic instruments or voices. A drone can simply be the root note of a particular key, or a root and its octave. The fifth or fourth is often added to establish more harmonic content” (Arnold, 2013).

It is to contain a range of pre-sets and an easy start “creator patch”.

It is to act as a creative hands-on platform for both live and studio musicians. Especially those used to having access to all controls without menu diving or having to access multifunction button sets.

This will be useful to those familiar with the signal flow, controls and behaviour of analogue machines. Who want to experiment with a number of oscillators. With the option of blending a looped sample (either user or embedded).

The platform chosen to create this instrument was Reaktor Primary. And the instrument was named Binary Star. A reflection of its spacey, ambient context.

The original aims consisted of several key elements to enhance the individual character of the instrument.

1.) Analogue emulation elements

A saturation element to simulate analogue warmth and distortion.

An oscillator drift algorithm linked to each oscillator. Creating the slightly detuned feel of a vintage analogue synthesiser.

2.) Subharmonic Generator

For each group of oscillators. To add more depth and presence to the low end between 30-90Hz.

3.) Embedded or user samples

Long sustained samples could be looped after the initial attack to provide a more continuous sound. Or attack transients could also be included in the loop. A small variety of embedded samples with selectable loop points would allow for a variety of wave shaping. Through research I found that Gamelan sound-sets from Indonesia could be a useful element in creating patches with entrancing musical textures.

“Research has documented oscillations in neuronal activity that entrain with rhythmic stimuli, linking observed behavioural entrainment to neuroscience”(McPherson, 2017).

The instrument will also give the user the option to drag and drop their own samples into the palyer.

4.) Solfeggio Frequency pre-sets.

A more esoteric idea for experimental sound design.

“Solfeggio frequencies have been identified by a set of six tones that were used long ago in Gregorian chants and Indian Sanskrit chants.” They “have been popularized and marketed for their healing qualities.” (Joseph, 2019)

At present its a pseudo-science as there is no real evidence with any scientific research.

They do appear to have been derived by Guido of Arezzo from the 10th Century. “He was a Benedictine monk and music theorist, who wanted a better way to teach songs to other monks.” (Good, 2015).

The use of these frequencies is popular in the new age movement as can be seen in play lists on YouTube and Spotify found in Appendix 1.

There is a market for those in the new age music and ambient genres to explore more detailed coverage of this within an instrument

A full list of Solfeggio major and mirror frequencies and their alleged properties can be found in Appendix 2.

5.) Delay

Provided as an optional effect to give more movement and depth.

6.) Convolution deep space reverb

This would use embedded algorithms to emulate actual physical spaces. These sample sets are normally called "Impulse Response" samples or IR. These could then be manipulated with internal equalisation and an envelope response filter.

7.) Ring Modulation Effect

Capable of modulating the original signal with another by a controllable amount and mix. The most intriguing unit that had relative key elements to examine was a Blocks version called Ring Mod (Tim, 2015). This had a variety of modulation waves.

8.) 27 Oscillators

With individual pitch control and a variety of waveforms such as Sine/Triangle/Pulse/Saw.

These can be sub mixed in groups of nine to pitch control, filter and low frequency oscillator (LFO) stages.

9.) Dual pitch control

As well as each group of 9 oscillators having pitch control +/- 24 semitones there will be a master pitch control that will create further bend of +/- 12 semitones.

The possible approaches that could have been taken

As this aims to emulate an authentic analogue drone synthesiser one possible approach would have been to build a hardware instrument.

The software route was taken for several reasons. Firstly; to create the same functionality with a hardware version would have involved making a hybrid digital/analogue synthesiser. Designing/debugging and blowing an EPROM (erasable, programmable, read only memory chip) for this and integrating it into an analogue structure would be quite complex.

Secondly due to the amount of oscillators and filters involved there would have been a lot of boards to make. Testing all the boards would take time.

However the hardware route would allow for easy customisation to create a unique sound (as opposed to learning Reaktor Core which is very in-depth!).

My original idea was to have 27 oscillators. This build idea was too central processing unit (CPU) intensive even on a modern computer. So the number of oscillators had to be scaled back.

With an analogue electronic circuit there would be no CPU load to consider.

Another approach was to mix the oscillators in pairs to a stereo mixer. One for the left and right input on each channel. This would have allowed for a more dynamic left and right placement between speakers. Which would have made routing to the 5 group filters difficult. The result would either be too many filters (one for each oscillator!) to be workable in the graphical user interface (or GUI). Or just one filter on the mixer output. It would also lessen the impact of the stereo harmonic structure by separating the feeds hard left and right.

But it would have created an interesting sonic structure for each speaker. Modulation could have been added to the effects returns to support them in gelling the signals into a stereo field.

A sequencer could have been used to allow creative patterns of individual oscillators assigned to different notes across a keyboard range.

This would have been another way to allow more movement in the patches.

It was decided against this direction as the limit of CPU was reached quickly with any increased polyphony (especially considering the envelopes of multiple oscillators with drone style sequences often have longer release times).

An option of switching off some oscillators relative to an increase in note polyphony and selecting shorter release rates on the attack, decay, sustain release envelope (ADSR) could have helped overcome this issue.

An audio input could have been added for a line or microphone signal featuring a vocoder to enrich the oscillators and samples.

A simple entry to the Reaktor user library called Vocoder 1.1 (MW, 2015) was examined for this as the prime modules could have been integrated into the ensemble.

It was decided against this approach as the complexity of multiple oscillators was already enough for a very rich signal. The addition of a vocoder could have made the signal too complex for coherent musical use.

There could have been an option to use it with a few oscillators only. This would increase the interactivity value of the instrument. Particularly to enable a musician to switch the instrument in and out as an effect to enrich particular chords. Or as part of a live set with a keyboard player triggering the midi and a vocalist or guitarist working through the same patch with the vocoder.

Details of the issues encountered while researching and creating Binary Star. And how I resolved these issues

Creating 27 oscillators with even just one voice of polyphony began pushing the CPU load to uncomfortable limits.

While this was working it already had a 45% CPU load on 32 GB of RAM with a 9th Gen i9 8 core CPU.

Although filters and a mixer had been applied at this stage, extra effect and modulation processing had yet to be added. It was clear this load would be unacceptable for use in projects involving many other plug-ins.

The oscillators were reduced to 18 to see how that worked with CPU load as other elements of the ensemble were created.

Having 18 oscillators with exactly the same functions might prove uninspiring. So the last two oscillators were changed to bi saw. With an adjustable pulse width. This gave an option of adding a different texture to the sound.

Looking at the high CPU usage also gave concern about adding a convolution reverb. After experimenting with Viva La Convolution (Brown, 2012) in the signal chain. It was apparent the CPU use would be too high. I looked at more economic CPU examples and with some experimentation found that using a delay based reverb was best. The one that sounded best was the Arena reverb (Lake, 2013). A simple reworking of Martin Zwartjez's metal reverb designed for Reaktor 5. The design and development of a bespoke reverb with this level of functionality within the time frame of this project would have been difficult. Mainly considering Arena was almost perfect with the chime sample set and multiple oscillators. So I integrated this into the effects chain.

I experimented with a sub bass harmonizer (Gabr, 2018). Sub bass synthesis can be used effectively for adding lower harmonics to singular audio sources such as a kick drum or bass line or even a cello. But it was found to be creating too many complex sub harmonic frequencies when the multiple oscillators were set to different pitches. This was causing distortion issues with the low end as the harmonizer tried to work with too many sources. Putting individual instances under each oscillator would take up too much CPU. It became apparent that it was always creating a very muddy unmusical effect with more than two oscillators on at the same time with varying tunings.

Of course just one instance could have been used with a selectable frequency at the end of the effects chain. But bass harmonics could still be created with this instrument by tuning down some of the many available oscillators. And by using the Geiger Oscillator. This will be discussed next.

I looked at making lower oscillator pitches by implementing the Geiger oscillator (a built in Reaktor Prime module). Where the main MIDI pitch went to the P input (the control input) of the Geiger oscillator. The “Clk” output (randomly generated clicks) from the Geiger oscillator was passed through an audio to event converter and then sent to the pitch input of one of the main variable wave oscillators. When the main oscillator was set to a saw wave using this method and combined with a regular saw wave at a higher pitch the results were quite pleasing. Giving more grit and pleasing harmonics between the combined saw waves. It was decided to use the first oscillator in each bank of four in the same way.

I studied the construction of the Analogue Drift Oscillator in the Reaktor user library (Aronson, 2018) and decided against including an oscillator drift algorithm (intended to emulate analogue oscillator detune behaviour). The drone instrument is intended as a set of constant tuned musical tones to play over.

Oscillator drift working between 2 or 3 Oscillators can have some pleasing harmonic movement if finely tuned. But with this drift function working over many oscillators (already with their own micro tunings) the results could quickly become chaotic and non-musical.

A rich harmonic palette could be created by focusing on micro tuning the oscillators and having them hold this tuning. Introducing modulation via effects could create movement with the harmonics in a more manageable way.

It was found that having saturation after each oscillator might overdrive the whole combined signal too much. So the plans were removed from the design and one single saturation stage for all oscillators was added.

This would also help to glue the sound together.

It felt more intuitive to add one control dial for the amount of pitch bend using a user defined range. Rather than using two pitch bend controls at different stages. This looked confusing and they could end up cancelling each other if the range values were conflicting.

Connecting the sampler with a divide setting to the main output from all the oscillators, created a pleasing distortion when combined with a signal from the oscillators.

In particular with the Geiger oscillator converted to audio added in at an earlier stage.

There was a real raw over driven rich analogue style sound.

I named this “The Mothership” setting!

The original idea was to switch between this for raw driving patches and then back to a regular direct output from the oscillators for more ambient use.

However no amount of attenuation on the dynamic range could control the signal between zero and total overload. Which rendered it inoperable in an open user environment unless the instrument would come with a warning to turn speakers right down first! In the end the regular direct oscillator output was used but with some added saturation to enhance rawness. If only a little compared to “The Mothership!”. This extreme setting would also have been outside of the original aims of the instrument.

I wanted to use more stereo field placement for the oscillators. With 18 oscillators as nine stereo pairs having individual left and right hand signals.

This created a problem as the ADSR, LFO, ring modulation and second stage filter all came at a group stage when the signal was still mono. To keep them would have meant having multiples which would have made the GUI too vast. Or using a complex modulation matrix which would have compromised the simplicity of use with the instrument. Not to mention the increased CPU power.

The power of having a rich harmonic mono signal over a compromised stereo field signal won over. To counter this, stereo panning was introduced at the group stage with the later inclusion of a stereo mixer for the sampler and oscillators.

My idea for the embedded sample set was to use Gamelan sounds. They have “Shimmering interlocking patterns, a seductive sound” (Broughton, 2001). However getting hold of a local Gamelan orchestra in Hampshire during lock down was not looking promising.

So I looked closer to home for a possible sample source and found one in the form of my front door wind chimes. There were only 3 chimes (of varying frequency).

I used an Audio Technica AT4033a to make a 24-bit 48KHz sample of each one. Then looked at combining and re-pitching them in different orders. I then tried a similar process with some very small chimes from a hanging mobile in the house. This made a useable collection of harmonics and tones. A workable solution considering the sound set would be used in most cases just to compliment the oscillators and then be effected in different ways.

I did not want users to be forced into using only the embedded samples. So a switch was added to go between the embedded chimes and a user sample drag and drop mode. This gave an extra angle for custom patch creation.

I also wanted to include pure sine samples as a set of Solfeggio frequencies within the embedded table. However in practise these pitches were only calibrating properly in the non-loop sampler.

I got round this problem by using hertz based tuning plug-in to calibrate the oscillators using a sine wave to the correct frequency. These were saved as snapshots to provide a selection of tuned pre-sets. Acting as a basis for patch creation using elements from the Solfeggio major and mirror frequency set (Appendix 2).

This worked out better than using the sound set in the sampler as it was now possible to have multiple tuned oscillators playing rather than one through the sampler (due to the limits of polyphonic voicing on CPU with large numbers of oscillators as detailed earlier).

Initially the layout of the instrument was somewhat confusing as there were many elements and it was hard to follow the signal flow visually. So the order of the instruments GUI was redesigned to represent the signal flow through the instrument.

So that sections were listed in the order of audio processing. Oscillators were grouped in their order with a sub group mixer and filter for that group under them. Then to the right of this came the ADSR, Filter and LFO for all the Oscillators. Followed by group dynamics and modulation. Next the sampler and its ADSR. Then the stereo mixer to combine the oscillators and sampler at the far right, along with the delay and reverb that were inserted in the two send and return fx groups with the stereo mixer. At the very far right the final 3 band parametric EQ for the final output. This was to make programming and playing the instrument more intuitive.

Although this worked on bigger HD screens there were problems with the more limited screen real estate on smaller screens.

So a version of the instrument with a more compact GUI was created to help address this and reduce scrolling.

After tuning all the oscillators to the different Solfeggio frequency set, it was apparent that unless the same key was used to trigger the instrument, the tunings would be out of pitch by the relative proportion of the key pressed versus the original intended key.

So I made middle C the reference point when programming pre-sets and put a graphic note on the GUI to “press middle C for all Solfeggio pre-sets”. The frequencies were then put in the pre-set names. Of course the entire keyboard range could still be accessed for those wishing to change key or create their own user patches.

With so much going on in the instrument I decided some user feedback would help.

So the instrument was sent to 12 users of Reaktor devices for feedback along with a brief instructional video (Appendix 3) and a selection of pre-sets.

The common feedback (Appendix 4) was that the device contained more oscillators than needed for most applications. To cut back on oscillators it was concluded that only one Geiger oscillator module was really needed to create usable lower frequency harmonics when combined with other saw waves.

So three Geiger oscillator to variable oscillator modules were removed. This freed up CPU usage, GUI real estate and made the interface easier to deal with.

Reflection on the results of Binary Star

Overall the instrument has many exciting possibilities but the initial daunting amount of functions that meet the users eye could be off putting.

With that in mind allowing user feedback was a great way to shape the device.

It helped to hone out some of the unnecessary elements within the sound design capacity. And to allow a greater working knowledge of end user perception with the interface.

Allowing users access to a range of Solfeggio frequency patches was a good creative addition.

While this particular set of frequencies and their alleged attributes are a concept without any real scientific proof. They can be used as a starting point for making artistic creative musical interpretations.

They create movement and texture as a fundamental when combined with some slightly detuned frequencies at higher harmonics and lower volumes.

I began to realise that having an ideal must be balanced by the consideration of its practical application. I had thought that using a software model would be far easier than creating a hardware model. A lot of the prime modules were borrowed and adapted from both the stock library (where possible) and user library where not. Yet here I reached a limit with what was practical in the design (without going into the deeper aspects of Reaktor Core). This was also reflected in my battle with CPU load. Although initial results were quicker than soldering components the end result was not as customised as I would have liked.

Yet the general sound was pleasing to the ear in other unexpected ways.

Perhaps by the process of having to find a work around, the adventure of creativity does yield more individuality after all. As the journey is not planned, it can be even more original.

I discovered that vision needs to be quantified by interaction in practise. Not just by me but using a control feedback group.

These valuable lessons can now go on to be applied to a variety of interactive product based design and development projects.

I learnt that a simple approach is often the best way. Especially when in doubt about which elements to include. The key is to make sure that each module is playing a strong role

It is a bit like cooking. Add just the right amount of each ingredient. Too much and it is over complicated. Not enough and it is uninspiring and bland.

There were many elements in the original aims. But I learnt that it is only through analysing and experimenting that each element could truly be assessed for its worth.

Playing with the built in modules led to unexpected and intriguing discoveries like “The Mothership” and feeding a Geiger oscillator into a saw wave oscillator to create slower oscillation rates.

The Reaktor user library and tutorials provided a useful resource for ideas. And supplied extensive details when looking for solutions to design problems. Helping me to expand on the original design and overcome limitations

Long established platforms can be useful as there is often an extensive amount of custom user plug-ins that can be examined, adapted, back engineered and integrated into a design.

Whilst creating the core concept of the instrument within the time constraints of this project was fine. It was then a compromise to develop other areas with the remaining time. In particular the deep spacey reverb and stereo expansion that I had envisaged giving an extra dimension to the sound.

At the beginning of designing such modules from the ground up, the programming would likely have been quite CPU hungry. Without time to streamline and develop each module so that it worked efficiently. So while it was great to find effects in the user library and to implement and edit them into the design, there was always a compromise in functionality.

However the overall sound was still unique, and sonically pleasing,

One of the great things about being able to create a pre-set bank is that there is a basic sound palette ready to dive into.

The only issue with the pre-sets is that rather than being tuned to a particular musical key, many of them were created as ambient textural sound scapes, or tuned to Solfeggio frequencies. Yet with a little tuning this could be altered. Plus there was an addition of a basic starter patch to help users with creating custom settings.

Consideration of how this could have been approached and what I would do differently next time

As the MIDI notes were held (by the instrument), both hands remained free to adjust pitches and tune effects and cross modulations.

It was great to have this access. But with more oscillators and modulation capacity than an average virtual synthesiser things could become quite daunting.

The resulting GUI was found not to be practical or necessary for all users. In response some oscillators were removed. But a further stage of development could have been implemented to create a clearer more intuitive GUI for those just wishing to access the most used controls and pre-sets.

There were a number of third-party customisable dials and text fonts plus graphic images available that could have been employed to direct the user more efficiently around the instrument.

Perhaps colour coding and enlarged dial sizes for the more commonly accessed controls would help.

With this in mind a more detailed customer feedback form about use of the GUI and commonly accessed functions would have been useful.

An area for future development would be to add a switchable display between Cent and Hertz for each oscillator. This would allow easy tuning to a particular scale or key/harmonic spectrum, without the need for a third party frequency analyser. Allowing more flexibility to tune the instrument in a live performance environment.

For any future project I would involve user feedback throughout all the different stages of development. From design, functionality, ease of integration with the GUI and shaping user pre-sets. Also assessing how often certain functions are accessed. This could develop where controls are placed and how big they are. Possibly merging lesser used functions within a separate menu for more advanced programming.

I have learnt how valuable constant feedback is to keep everything in context.

It highlighted to me why companies have such big research and development departments. And why prototypes for customer consultation are vital.

I would definitely delve further into developing a set of bespoke effects for any future project. In fact the deeper I could get into the coding of every element, the more customized and individual the sound would become. Creating a recognisable sonic character throughout. Plus streamlining CPU usage. These modules could then be included with other instrument builds. The effects could even be expanded to be realised as standalone plug-ins.

The requirement to constantly refine and hone a design will always be there. The key is to get better at the process.

Bibliography

Arnold, L. (2013) *How sound affects you: using drones in your music* [Online] Available from <https://ask.audio/articles/how-sound-affects-you-using-drones-in-your-music> [Accessed 8th June 2020]

McPherson, T. (2017) *Researching the brain through Balinese Gamelan* [Online] Available from <https://college.unc.edu/2017/08/musical-neuroscience/> [Accessed 14th June 2020]

Joseph, S. (2019) *Sound healing using Solfeggio frequencies* [Online] Available from https://www.researchgate.net/publication/333852911_Sound_Healing_using_Solfeggio_Frequencies [Accessed 20th June 2020]

Good, C. (2015) *Solfeggio frequencies* [Online] Available from <https://skeptoid.com/episodes/4474> [Accessed 20th June 2020]

MW. (2015) *Vocoder 1.1* [Online] Available from <https://www.native-instruments.com/en/reaktor-community/reaktor-user-library/entry/show/8745/> [Accessed 28th June 2020]

Aronson, A. (2018) *Analog drift oscillator* [Online] Available from <https://www.native-instruments.com/jp/reaktor-community/reaktor-user-library/entry/show/12223/> [Accessed 29th June 2020]

Brown, C. (2012) *Viva la convolution* [Online] Available from <https://www.native-instruments.com/jp/reaktor-community/reaktor-user-library/entry/show/6841/> [Accessed 30th June 2020]

Lake, B. (2013) *Arena Reverb* [Online] Available from

<https://www.native-instruments.com/jp/reaktor-community/reaktor-user-library/entry/show/7297/>

[Accessed 30th June 2020]

Broughton, B. (2013) *Indonesia: Sounds like* [Online] Available from

<https://www.telegraph.co.uk/travel/destinations/asia/indonesia/718144/Indonesia-Sounds-like.html>

[Accessed 12th June 2020]

Tim, H. (2015) *Ringmod* [Online] Available from

<https://www.native-instruments.com/en/reaktor-community/reaktor-user-library/entry/show/8747/>

[Accessed 30th June 2020]

Surrell, N. (2013) *Solfeggio Frequencies* [Online] Available from

<https://nevillesurrell.wordpress.com/2013/11/27/solfeggio-frequencies/>

[Accessed 14th June 2020]

Appendix 1

Solfeggio playlists

<https://www.youtube.com/playlist?list=PLsuCfYXzi5DIQJ1aHf3X4LKtmjKyj5Az>

<https://open.spotify.com/artist/5f1gvzToaopu6VOaBKXE2D>

Appendix 2

The Solfeggio Frequencies and their mirrors plus qualities (Surrell, 2013).

- 1- 147 Hz Harmonic Frequency ; Subliminal
- 2- 174 Hz Harmonic Mirror Frequency ; Foundation
- 3- 258 Hz Harmonic Frequency ; Concentric Ascension
- 4- 285 Hz Harmonic Mirror Frequency ; Quantum Cognition
- 5- 369 Hz Harmonic Mirror Frequency ; Inner Harmony
- 6- 396 Hz Major Solfeggio Frequency UT ; Liberation from Guilt and Fear
- 7- 417 Hz Major Solfeggio Frequency RE ; Undoing Situations and Facilitating Change
- 8- 471 Hz Harmonic Mirror Frequency ; Eternal Window
- 9- 528 Hz Major Solfeggio Frequency MI ; Transformation and Miracles (DNA Repair)
- 10- 582 Hz Harmonic Mirror Frequency ; Galactic Fire Strand
- 11- 639 Hz Major Solfeggio Frequency FA ;Connecting/Relationships Integrating Structures
- 12- 693 Hz Harmonic Mirror Frequency ; Violet Flame
- 13- 714 Hz Harmonic Mirror Frequency ; Reordering Perceptions
- 14- 741 Hz Major Solfeggio Frequency SOL ; Awakening Intuition , Third Eye Activation and Pineal Stimulation
- 15- 825 Hz Harmonic Mirror Frequency ; Silver Violet Fire
- 16- 852 Hz Major Solfeggio Frequency LA ; Returning to Spiritual Order

17- 936 Hz Harmonic Frequency ; Mastery

18- 963 Hz Harmonic Mirror Frequency ; Numinous Accord

Appendix 3

Instructional video for all end user testers for feedback

<https://youtu.be/WY7NSJbl4K0>

Appendix 4

User feedback contacted via personal Facebook contacts.

1	NAME	REQUEST	AGREED	SENT	FEEDBACK
2	Christian Schorschinski	YES	YES	YES	Very interesting. May well suit some of my projects.
3	Gemma 'Spiro' Kristina Bowen	YES	YES	YES	N/A
4	Jamie Tassell	YES	YES	YES	I really like it but would probably only use 8 Oscillators. But then again I'm making techno.
5	Igor Fabris	YES	YES	YES	Very nice for space music! good selection of patches. not sure all the low freq osc are needed! too many!
6	Jaroslav 'Orkus' Konečný	YES	YES	YES	strange but cool.
7	Tony Kuhn	YES	YES	YES	Yes like! Need time to explore this. 18 oscillators is alot!
8	Simon 'xetechno' Owen	YES	YES	YES	Will try this out in some studio sessions
9	Iain M Gill	YES	YES	YES	Just waiting on new laptop so will feed back then.
10	David Akahum	YES	YES	YES	Looks like an exciting monster! Will get back to you once I've had a proper play.
11	David Rayner	YES	YES	YES	N/A
12	Jason Baker	YES	YES	YES	N/A

