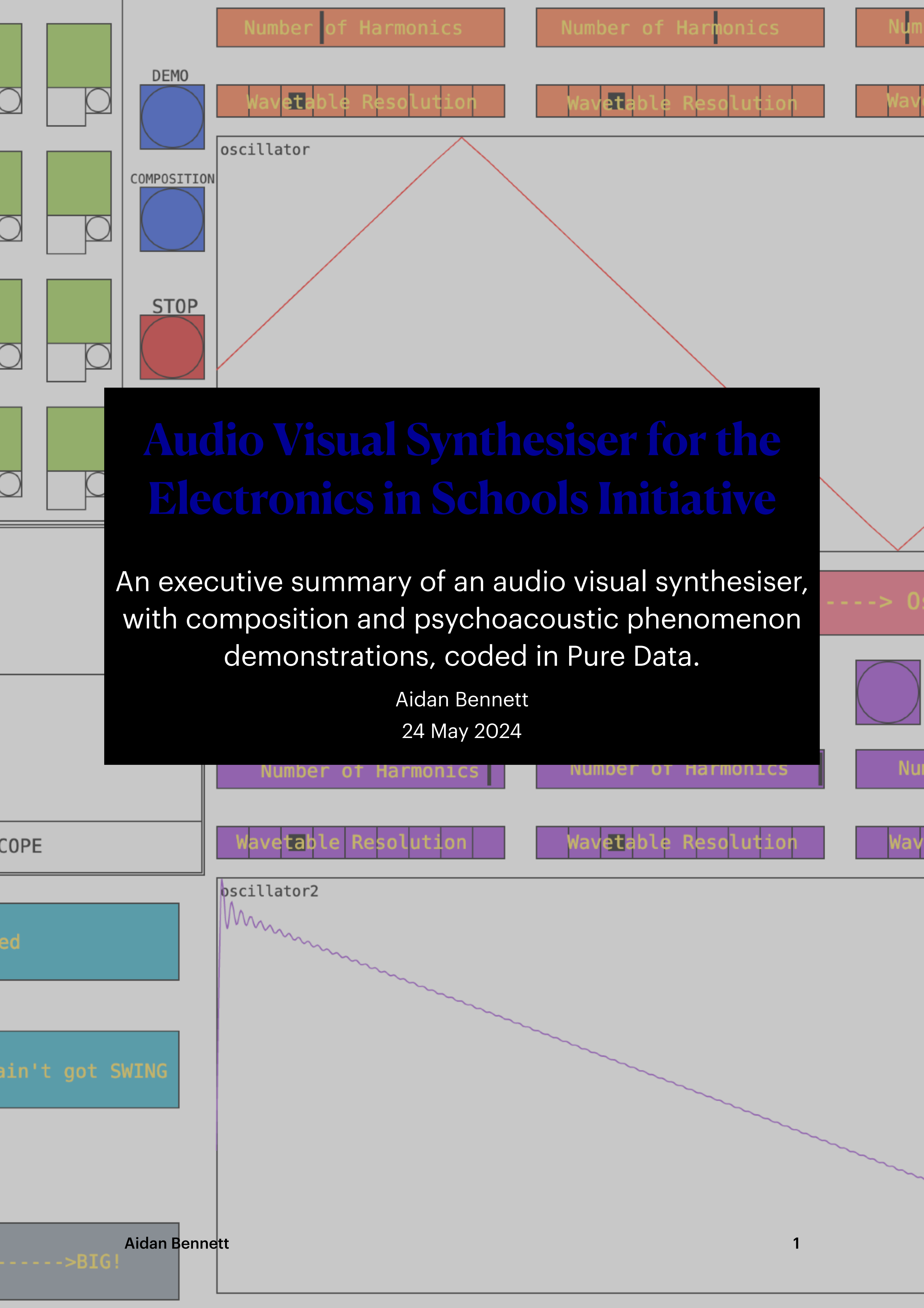

Audio Visual Synthesiser for the Electronics in Schools Initiative

An executive summary of an audio visual synthesiser, with composition and psychoacoustic phenomenon demonstrations, coded in Pure Data.

Aidan Bennett

24 May 2024

Abstract

This report is an executive summary of the technical aspects of the *16BitWavetableSequencer* Purr data patch. It gives an overview of the synthesis method, the context and inspiration behind the audio and visuals, the design rationale, and some brief user testing. It then explains the "streaming" psychoacoustic phenomenon, including how it was implemented technically speaking, an evaluation, and some spectral plots of the demonstration output. The report concludes by evaluating the final work and suggesting improvements for future iterations of the software.

1. Introduction

Response to the briefing; How your approach combines synthesis and psychoacoustics

The synthesiser summarised in this report was in response to the brief from SynthAcademy Inc. to create an audio visual synthesiser, aimed at sixth form students, which allows them to explore synthesis via an intuitive user interface, and to explore a psychoacoustic phenomenon with the same interface. In response to this briefing, the synthesiser has a sequencer at the core of it; this allows students to generate repeating patterns of tones so that they are free to explore the wide timbral aspects of the synth as they play. The idea behind the visual aspect of the synthesiser was to represent the waveforms produced in an aesthetically pleasing way, somewhat like a more artistic and abstract version of putting a signal through a standard laboratory oscilloscope, that responds in real time to the final signal produced and it’s timbral characteristics.

The psychoacoustic phenomenon that the 16-bit sequencer based synthesiser lends itself to most intuitively; streaming, which will be detailed in section 3, is a phenomenon that manifests itself most obviously when the speed of a repeating pattern of alternating high and low frequency notes is slowly increased. The sequencer therefore aims to combine synthesis and psychoacoustics by offering a automated demo which speeds up the sequencer linearly after programming high and low tones so students can judge the threshold of the steaming phenomenon, and provides a sequencer speed slider such that students can explore this threshold manually.

2. Synthesis Report

2.(1) Introduction to the Synthesiser

The synthesiser uses wavetable synthesis to produce a waveform with frequencies that are programmable as MIDI notes into 1 of 16 individual "bits" of a sequencer. The sequencer speed, and other parameters, can be controlled with a colourful graphical user interface.

Wavetable synthesis is when samples are loaded into one or more wavetables, which are essentially arrays of amplitude values. The frequency of the outputted tone depends on the rate at which the index of a pointer, which reads individual samples in the wavetable, is iterated through this array; the number of samples it needs to iterate with each sampling period is given by the equation in figure 1, where i is the integer of samples to iterate, f is the desired frequency, N is the total number of samples in the wavetable array, and r is the sampling rate in Hertz [2].

$$i = \frac{f * N}{r}$$

Figure 1 - Sample increment equation[2]

Figure 2 shows the part of the synthesiser patch which accomplishes this: the two "tabos4~" object each read from one of two respective wavetable arrays, oscillator1 and oscillator2, which can be loaded with different waveforms as shown in figure 3. The frequency is taken from the sequencer bit currently being activated. These objects are sent to "*~" objects, which have "vline~" objects attached to them, so that the amplitudes of the oscillator1 and oscillator2 can be faded between using the pink "wavetable mixer" slider in figure 3.

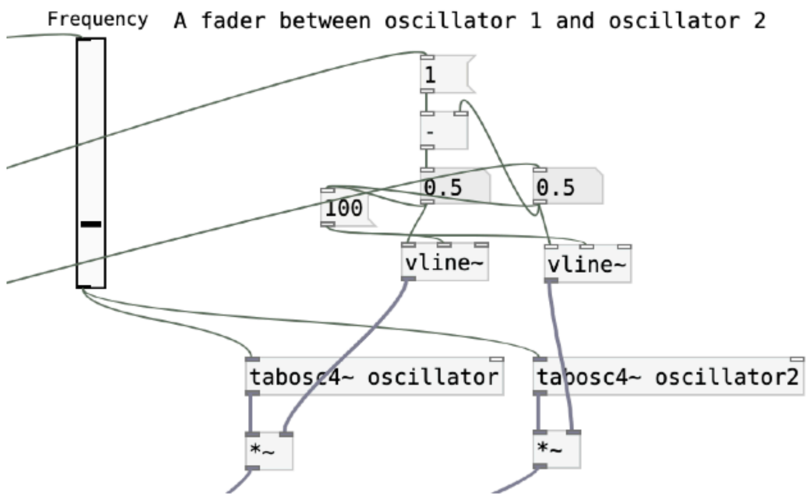

Figure 2 - "tabosc4~" object patch

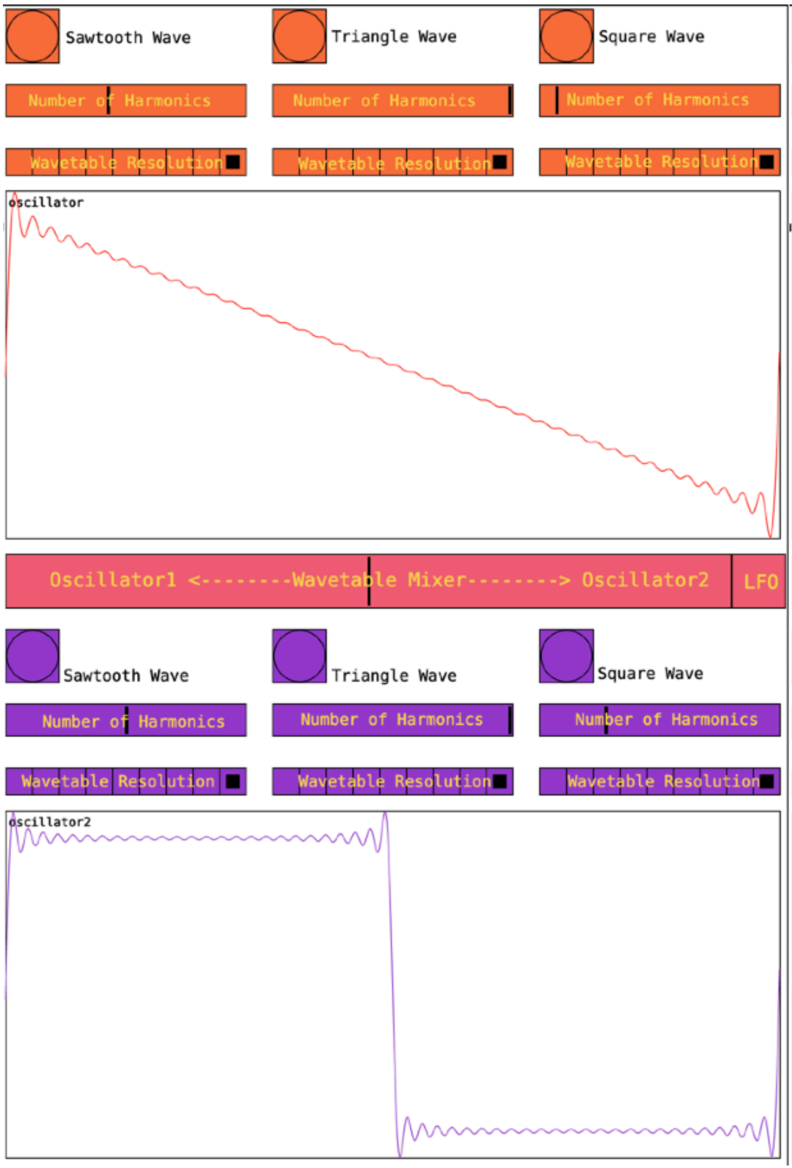

Figure 3 - Wavetables example 1

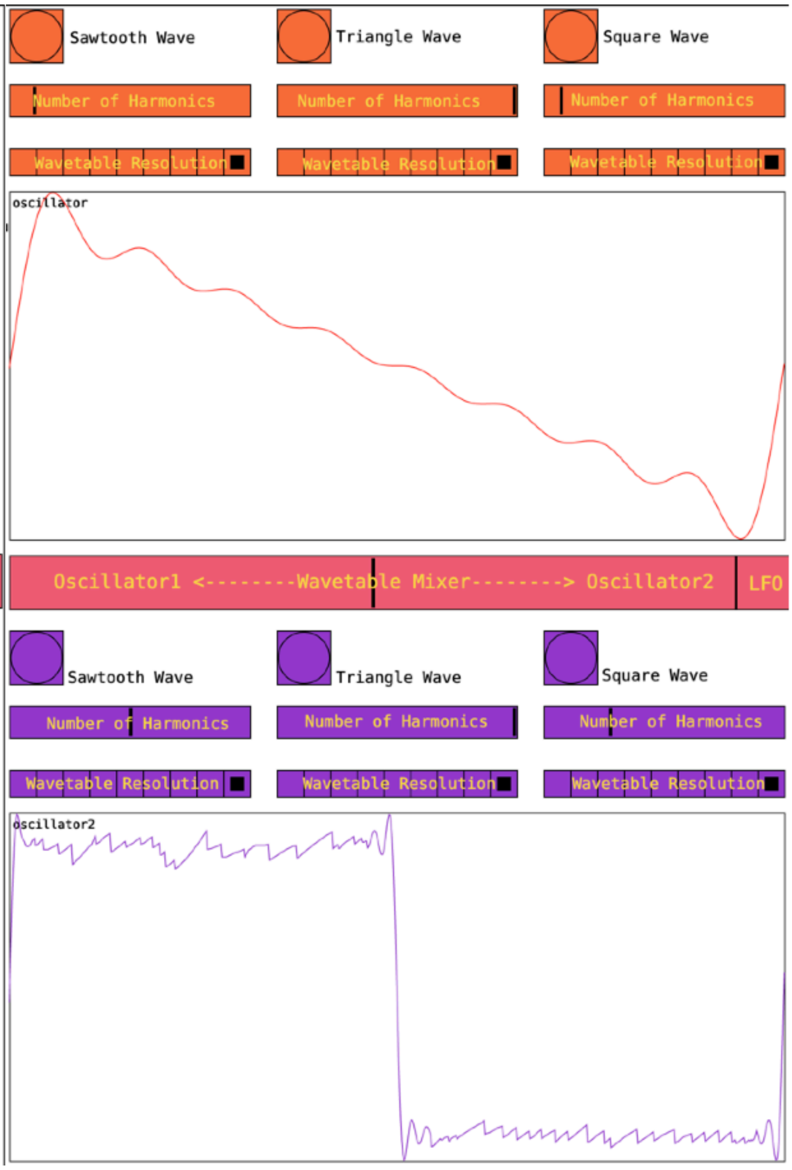

Figure 4 - Wavetables example 2

The two wavetables can be drawn on to produce “buzzier” timbres, and sawtooth, triangle, and square waves can be loaded into them with a variable number of harmonics. For example, contrast the oscillator1 sawtooth wave in figure 3 with that of figure 4 which has fewer harmonics but is the same sawtooth shape. Also compare the square wave in oscillator1 in figure 1 with the “buzzier” sounding square wave that has been drawn on to the wavetable by the user.

2.(2) Pure Data Design Rationale

The inspiration behind the Pure Data sequencer patch was the range of “Pocket Operators” by the company “Teenage Engineering” [3]. To match the basic operation of these pocket operators, the synthesiser interface has a 16 bit sequencer which enables programming of 16 bit rhythms that activate in sequence depending on the sequencer speed, and which bits of the sequencer are currently selected. See the patch screenshot in figure 5 including comments for details of how this works.

The sequencer works in tandem with a piano keyboard style layout of tones ranging from C2 (65.406Hz), to C4 (261.63Hz), in a chromatic scale [4]. With one of the 16 bits of the sequencer selected for editing, either the physical piano keyboard, or the labelled software keyboard on the GUI, can be used to program that sequencer bit with a frequency. Once the “START Sequencer” toggle is selected, the sequencer cycles through each bit at the sequencer speed, and activates the stored frequency with all the timbral elements the synth is currently set to.

The keyboard layout has also been routed to a duplicate of the “tabosc4~” object based patch shown in figure 2 such that tones can be played with the computer keyboard, or GUI software keyboard, along with the sequenced tones.

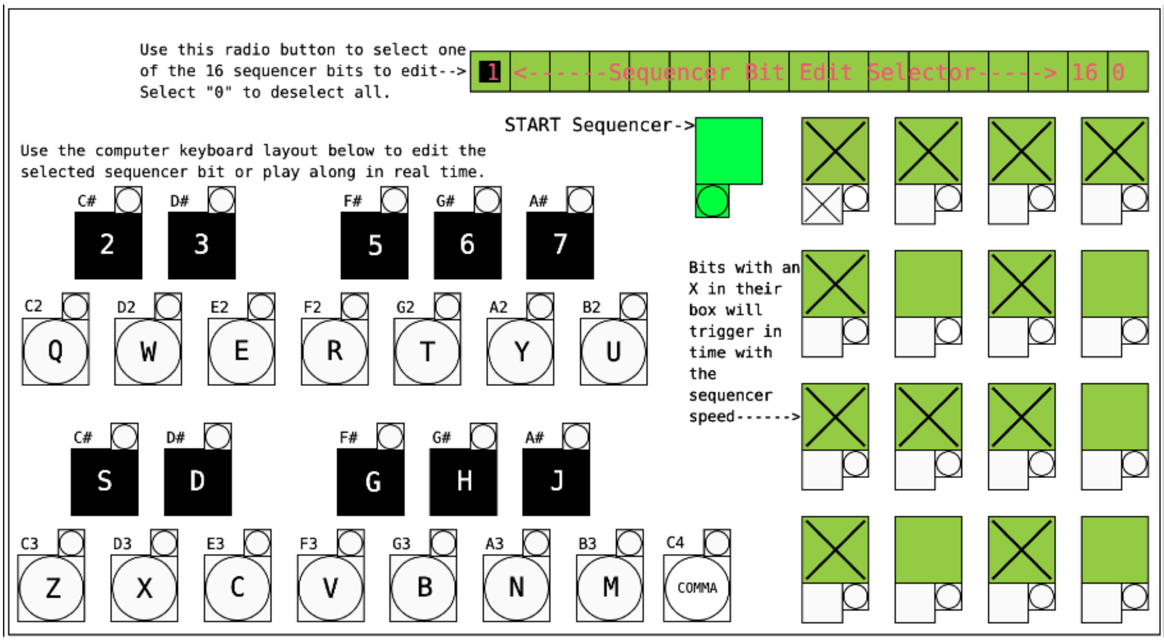

Figure 5 - The sequencer GUI

Figure 6 shows how the sequencer was designed: a “counter” object cycles from 0 to 15 in increments of 1, then resets to 0 at a rate specified by the “sequencer speed”. Each integer is fed into the “select” object which sends a bang to one of 16 depending on the index of the sequence. The “spigot” object is used next in the signal chain such that this bang message is only allowed through if there is a value of 1 on it’s right-hand inlet; each of the 16 bits arranged in the grid in figure 5 is routed to a corresponding on of these “spigot” objects so that if a bit’s toggle box is selected, it’s bang signal is allowed through which triggers the note stored in it.

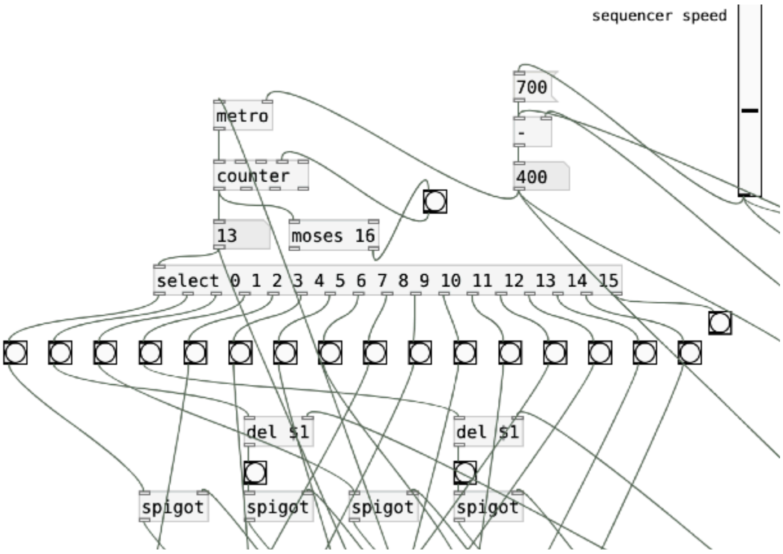

Figure 6 - Sequencer counter: (most spigot objects have been cut off the bottom of the image)

2.(3) Audio-Visual Demonstration

The visual aspect of the synth uses 3 “scopeXYZ” GEM objects. These produce 3 oscilloscope patterns in a gem window. Each of them combines a 20Hz oscillator with a signal such that an oscillating trace of this signal is drawn on the screen, with colours that depend on the current timbre of the synth; the top left oscilloscope visualises the output of the sequencer, the top right the output of the software keyboard, and the third one visualises the final output waveform on the bottom of the gem window. The size of the top two are mapped to the envelope of each tone, so that they reinforce amplitude at any time visually.

The demonstration uses “line” objects to vary the sequencer speed, whilst a set of random number generators randomise the timbre of the synth every sequencer pulse.

2.(4) Testing and Evaluation

A user tester was shown the audio visual demonstration and had these comments: The cacophony is quite daunting, and due to the randomness and clashing tones, but this goes with the chaotic and dark visuals. When there were harsher sounds there were lighter colours, and mellower sounds meant darker colours, and the waveform “danced” along [5].

A second user tester said the following: “it felt futuristic, and like something that could be used at concerts to make experiencing music feel more encompassing.” [6]

The sequencer is very effective at cycling through the tones in a visually pleasing way, with an intuitive user interface that gives easy control of the sequence. The visuals map well to the tone produced and seem to dance along. A wide range of timbres can be produced, and random presets can be generated for more creative exploration. However, the internal code is mostly messy and poorly commented; splitting areas of the code into more sub-patches to make it all self commenting would make it less confusing and facilitate ongoing development and collaboration.

3. Psychoacoustics Report

3.(1) Explanation of Psychoacoustic Phenomenon

The psychoacoustic phenomenon demonstrated in my synthesis patch is a version of auditory streaming in the galloping paradigm. The phenomenon occurs when three alternating tones - a low, followed by a high, then back to the low tone before repeating - are far enough apart in frequency that at a certain speed of iteration they will be perceived as a single stream of notes in the sequence low->high->low, but at another speed they will be heard as two streams: a low tone pulsing at a regular frequency, and a simultaneous stream of the higher tone pulsing at half of this frequency. The single stream is heard at slow iteration speeds and the streams perceptively separate at a subjective high speed value [7].

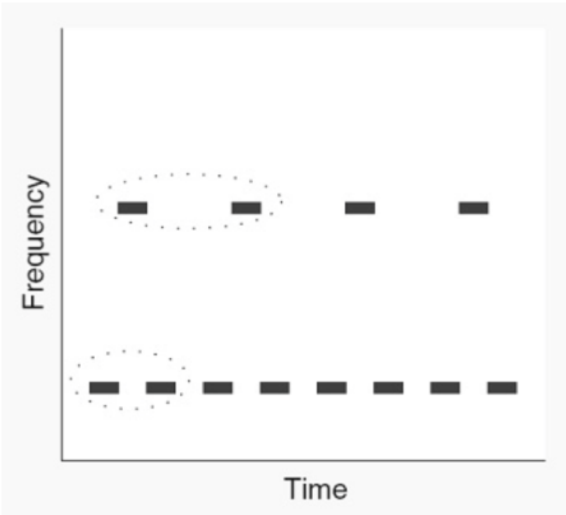

Figure 7 - Galloping rhythm perceived as simultaneous lower and higher streams [7]

3.(2) Design and Implementation

The psychoacoustic demonstration in the synthesis patch uses a “line” object to vary the speed of the sequencer from 610 ms periods to 100ms periods over a 45 second time interval. Somewhere close to the 100ms sequencer period mark the two streams will perceptually separate.

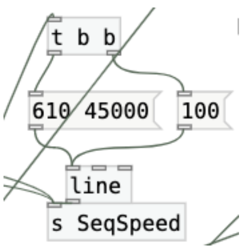

Figure 8 - “line” object for sequencer speed

The idea is then that whilst the sequence is still ongoing a student can vary the sequencer speed manually around this threshold using the "sequencer speed" slider to find the point at which the two streams perceptually separate subjective to them. They can also use the sequencer keyboard to input different tones into the galloping sequence to experiment if this has any effect on the phenomenon; when the interval between two tones is small enough, it is more likely that a single stream of tones will be perceived [7].

3.(3) Evaluation of Effectiveness

The demonstration is very intuitive and easy to use as it is initiated with the push of a button which is clearly labelled on the GUI. The interactivity with the phenomenon is also built in to the interface of the sequencer itself, which makes exploring it further technically very simple. However, there is no guidance on how to proceed with this exploration, so without external guidance on the nuances of auditory streaming 6th form students may be unlikely to try editing the parameters in the correct way; adding a guided script to the GEM interface as the demonstration is playing with tips of what parameters to try editing next would make the demonstration more interactive.

The auditory streaming effect could be distinctly heard by developer of the patch when they tested the automatic demonstration, and experimenting with the sequencer speed slider resulted in the perception of streams changing in real time.

4. Conclusion

In conclusion the 16 bit wavetable sequencer demonstrates a wide range of timbral possibilities which can be easily and intuitively altered with a colourful and neat graphical user interface. These timbres can be creatively explored with the help of randomisation which will be beneficial to sixth form students' initial learning as it flattens the initial learning curve with visual feedback.

The visual mappings of the synthesiser co-ordinate with the output in an aesthetically pleasing way which makes it even more engaging to use.

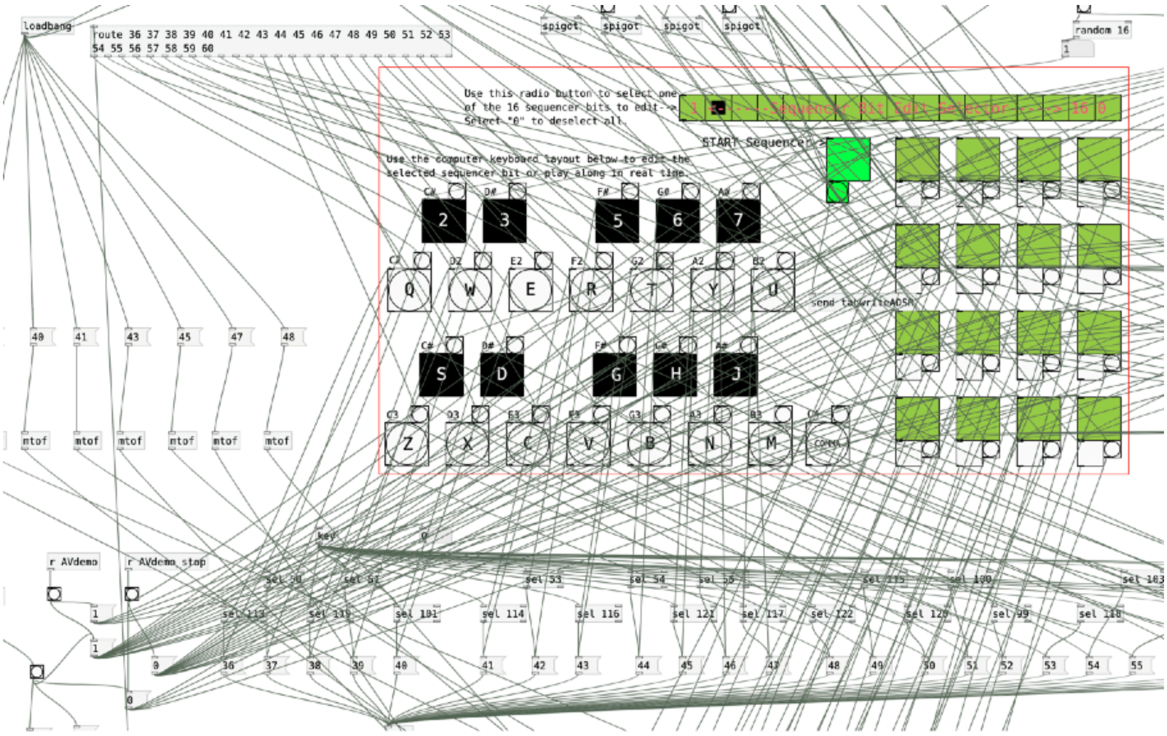

Figure 9 - The messy sequencer sub-patch

However, the internal code is poorly commented and is very messy, especially in areas like the sequencer sub-patch shown in figure 9. Therefore, the first bit of future work would be to separate the code into a lot more sub-patches and neaten all the objects so that continuous improvement is not stunted by difficulty reading the code. This would also allow more audio engineers to collaborate with the original developer to make an even more effective product for the sixth form students.

5. References

[1] IEEE Periodicals, Transactions/Journals Department. “IEEE. REFERENCE GUIDE” ieeeauthorcenter.ieee.org. https://ieeeauthorcenter.ieee.org/wp-content/uploads/IEEE-Reference-Guide.pdf (accessed Jan. 16, 2022)

[2] https://digitalsoundandmusic.com/6-3-3-type-of-synthesis/ (accessed May. 23, 2024)

[3] https://teenage.engineering/products/po (accessed May. 23, 2024)

[4] https://audiodev.blog/midi-note-chart/ (accessed May. 23, 2024)

[5] User tester “Josh” was shown the audio visual demonstration in full on May. 23, 2024. He was asked to comment on what came to mind, and the notes from this conversation have been paraphrased.

[6] User tester “Anushka” was shown the audio visual demonstration in full on May. 23, 2024. She was asked to comment on what came to mind, and dictated exact quotes which are shown here.

[7] https://auditoryneuroscience.com/scene-analysis/streaming-galloping-rhythm (accessed May. 23, 2024)