

# PureData Synthesis and Composition

ELE00006C

<b>Submission</b>	<b>1</b>
<b>Composition</b>	<b>1</b>
<b>Synthesisers</b>	<b>1</b>
WAVETABLE SYNTHESISER	1
FM SYNTHESISER	1
<b>Final Piece</b>	<b>2</b>
SEQUENCING	2
CRITICAL ANALYSIS	2

## SUBMISSION

<code>recording.wav</code>	<b>The recording of the run through the composition.</b>
<code>master.pd</code>	<b>The main master controller where you can start the composition/recordings.</b>
<code>8step_note.pd</code>	An 8-step sequencer control for an individual pitch.
<code>8step.pd</code>	A programmable 8-step sequencer with 8 notes.
<code>adsr_control.pd</code>	An ADSR envelope interface that draws the step response of the envelope.
<code>adsr~.pd</code>	An ADSR envelope generator that is triggered by incoming notes.
<code>fm_voice_carrier~.pd</code>	A carrier operator for the FM synthesiser.
<code>fm_voice_modulator~.pd</code>	A modulator operator for the FM synthesiser.
<code>fm_voice~.pd</code>	An individual voice for the FM synthesiser (multiple of these used for polyphony).
<code>fmsynth~.pd</code>	The FM synthesiser main patch.
<code>pingpong~.pd</code>	A stereo 'ping pong' delay that alternates between channels.
<code>reverb~.pd</code>	An interface for the 'freeverb~' object.
<code>wavetable_voice~.pd</code>	An individual voice for the wavetable synthesiser.
<code>wavetable~.pd</code>	The wavetable synthesiser main patch.

## COMPOSITION

My idea was a soundscape that changes from a thick, sinister, and dark start to something serene and elevating. I also wanted to have some fragmented melodies to develop an otherworldly feeling.

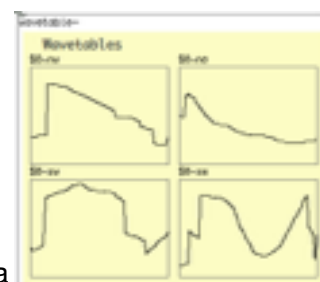
I wanted an expansive sound so I worked on integrating a reverb effect as well as a panning 'ping pong' delay that alternatively feeds back through stereo channels and creates a long chain of sound.

After experimenting I found that FM synthesisers could create quite deep and textured dark sounds, whereas wavetable synthesisers seemed to be good at creating sounds with a lot of shifting high frequencies which seemed suited to the more ascendent part of the composition.

## SYNTHESISERS

### Wavetable Synthesiser

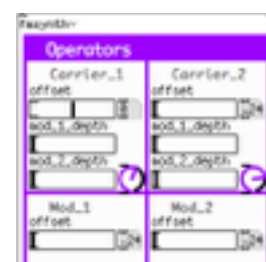
I started out by building a simple wavetable synthesiser with two wavetables which shifted between the two waveforms based on the ADSR envelope. I thought the sound was too static so I added another dimension with two more waveforms that would be panned with an LFO. The result is a synthesiser with four wavetables where the particular vector set by the ADSR and the LFO controls the output waveform.



In order to achieve polyphonic operation, the synthesiser is split up into several 'voice' sub-patches which share inlets and are routed using the poly object and route object.

### FM Synthesiser

When designing the FM synthesiser, I found that multiple operators are needed to create more interesting sounds. Because of this I had to divide the FM synthesiser into further components, and it is structured with sub-patches for modulators and carriers, both of which are positioned in a voice sub-patch

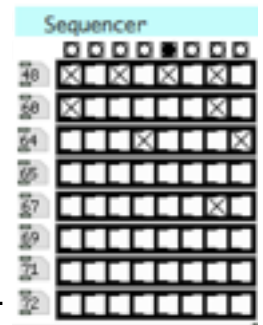


which works as part of the polyphonic dispatch process in the same way as the wavetable. Then there is a master patch where the parameters can be set by the user.

## FINAL PIECE

### Sequencing

Whilst developing the synthesisers, I used a MIDI keyboard to experiment as easily as possible. Unfortunately it's not quite so straightforward to sequence a composition in PureData as it is to play on a keyboard. Ideally I would have liked to have created a system that recorded the MIDI events that came in and played them back, however I elected for something simpler in my piece to try and get some more emergent behaviour.



I created an 8-step sequencer so that I could quickly experiment with combinations without having to manage a complicated `qlist` arrangement.

I also used this as the opportunity to include my exam number as part of the composition. There is a sequencer for each number in my exam number, and the clock speed for each is dictated by the number followed by some number of zeros. E.g. '3000' and '700' for the first and second digits. I couldn't do this for the final number in my exam number ('0'), however I interpreted this number to be the end of the composition, and therefore it comes to a rapid close at the 1 minute mark.

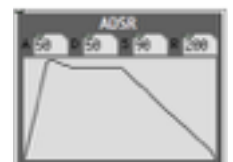
### Critical Analysis

The final piece suffers from lack of modulation of other parameters of the synths during the composition, e.g. LFO frequency, depth. Whilst these do change between instances of the synths, they remain static throughout the composition. I had problems with PD the workspace became very jittery and laggy and would not respond properly after I created my sequencer, and it became very difficult to develop the composition further at this point. I would like to investigate the performance of PureData applications and learn how to do things in a way that won't result in any unresponsiveness, so that I could develop the ideas further.

In terms of the sound design, I feel like the composition achieves its goals in outline. The piece does start off sinister and develops into an ascendant rush. However I would have liked to develop the texture of the sinister sounds a lot more, and try experimenting with pitches and ratios that are not part of standard musical scales.

I feel like a sound that stands out as being effective is the sound that is generated using the wavetable synth sequenced at incredibly fast speeds (the 8-step sequencer is progressing at one beat every 8 milliseconds). This created an incredibly dense array of shifting harmonies from the notes as they fade in and out rapidly, almost like having a third kind of synthesiser made out of the sequencer itself.

I was very pleased with my ADSR control patch which automatically draws the envelope for the user to inspect. In hindsight though it took me a long time to figure this out which took away from development time for the synths.



One other thing to note is that as I developed the composition I discovered much more effective ways to route data around PD. I would love to make improvements to the parts of the solution I did not have time to alter to use these techniques. Particularly using `route` to respond to particular symbols sent into the inlets seems much cleaner than filling the screen with connecting lines.