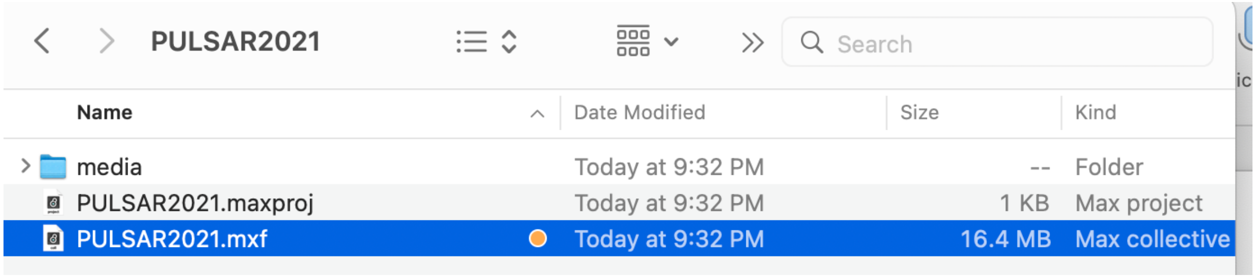
Pulsar (2021) – A Quick Guide

Pulsar is an electroacoustic instrument for solo or group laptop improvisation. It was made using Max/MSP and is shared using a Max collective. This is similar to making a standalone desktop application and often results in a much smaller file, but does require a copy of Max. It is simple enough to use without any musical training, but this document will discuss some ideas that require some background in music technology/Max/MSP. This is not a comprehensive guide but does offer some information about the intended design and use of this instrument.

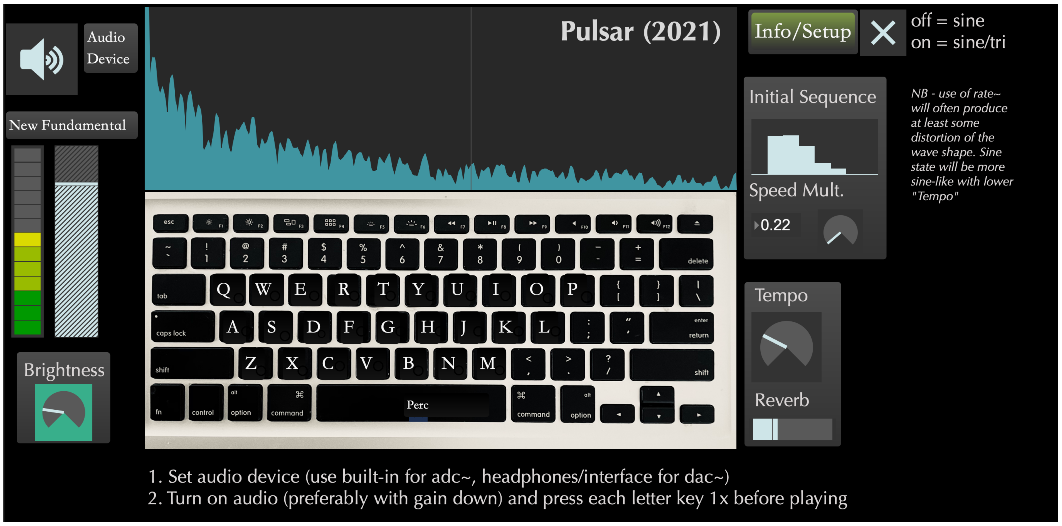
The .mxf file is what you use to open the instrument. Note that it can take 5-10 seconds to open.



PULSAR2021

Name	Date Modified	Size	Kind
media	Today at 9:32 PM	--	Folder
PULSAR2021.maxproj	Today at 9:32 PM	1 KB	Max project
PULSAR2021.mxf	Today at 9:32 PM	16.4 MB	Max collective

Here's what the interface looks like (turquoise wavy display only shows when audio is running through the instrument):



off = sine
on = sine/tri

NB - use of rate~ will often produce at least some distortion of the wave shape. Sine state will be more sine-like with lower "Tempo"

1. Set audio device (use built-in for adc~, headphones/interface for dac~)
2. Turn on audio (preferably with gain down) and press each letter key 1x before playing

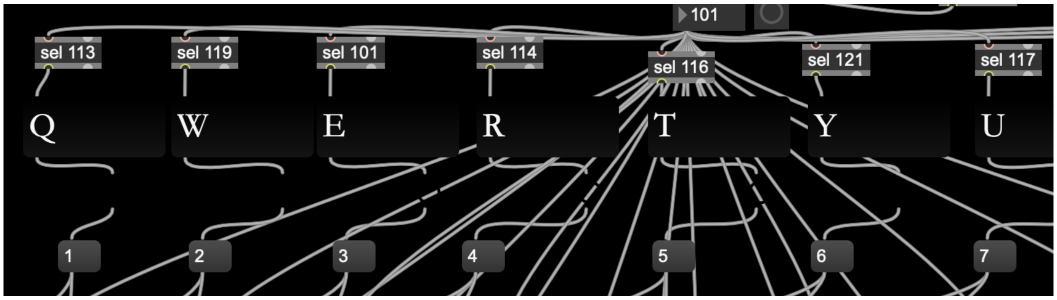
Before you begin to improvise…

1) You should use the "Audio Device" button on the top left to make sure that your that your output is set properly (presumably headphones or an audio interface), and that the input is set to your built-in microphone.

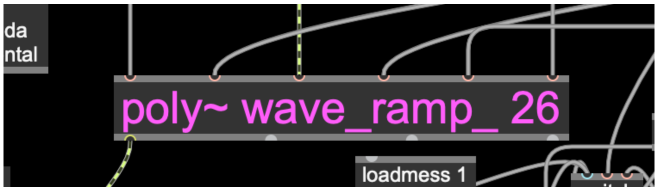
2) The ` key (below the ~ tilde) turns off the pairing of the master gain slider (bottom left, just above the “Brightness” dial) with the vertical position of your cursor. Clicking with your track pad turns this feature back on. You need to turn the dac~ on (speaker logo button, top left) and press each letter key on your keyboard once, using lower case keys. You can use your mouse to lower the gain level. This part of the setup process will avoid clicks/signal discontinuities after you begin to improvise.

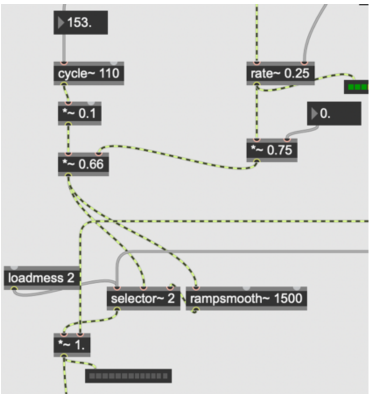
How *Pulsar* Works

This instrument uses the idea of the QWERTY keyboard (lower case letters only) as the main tool for play. A fundamental frequency value is set and each letter key multiplies that fundamental frequency using consecutive integers. The Q key multiplies by 1, the W key by 2, the E key by 3, and the integer multiples of the fundamental increase until you get to the M key (fundamental x 26). In this sense, you have the harmonic series of the fundamental frequency as decided pitch material. But the instrument shouldn’t be thought of as primarily pitch oriented.



Each letter corresponds to an integer multiple (this is from inside of the main patch)

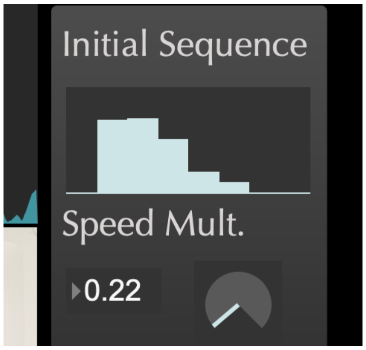




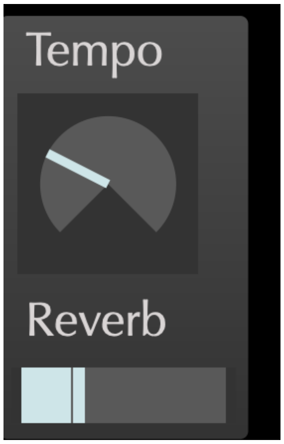
The sounds generated from this instrument rely on one primary and another secondary source. The primary source can be found in the “poly~ wave_ramp_ 26” object – which is visible if you leave presentation mode and enter patching mode. This object contains a subpatcher that has a single sine wave oscillator (this is the cycle~ in Max). The oscillator’s wave shape can be distorted in 2 ways. 1) using the rampsmooth~ object, 2) using the rate~ object. rampsmooth~ is often helpful for resolving audio errors, and rate~ is great for sample-accurate sequencing, but here they’re used to distort a sine wave so that it creates an interesting timbre.

Drum-like sounds can also be produced by using the space bar. Once you hit the space bar, you will hear the sound of your keystrokes being passed through the internal mic into filters and pitch shifters to get a tom-tom like acoustic sound. This can be disabled by simply hitting the space bar once again.

There are some useful features of this instrument that make it less of a simple matter of assigning frequencies to an oscillator, and more oriented around texture/rhythm as a driver for improvisation. For example, the instrument is able to hold chords – kind of like stuck MIDI notes. If you press multiple keys simultaneously, they will hold the frequency (because there are 26~ instances of the poly~ object, hence you could even use all 26 oscillators if you wish to do so). If you want to release notes from the "chord," you simply press the individual QWERTY key that you wish to release.



Also, if you hold a key for a couple seconds, a sequencer (reflecting the shape in the drawn below "Initial Sequence" in the GUI) will begin to transpose the Hz value of a particular key's oscillator. The sequence can be redrawn by clicking and dragging, and speed of the sequence (with 1.0 at normal speed) can be multiplied using the dial below. If you hold the key for especially long then the sequence will become randomized.



The Tempo dial can be used to create both rhythmic and timbral effects. It controls the "multiplier" (essentially speed) argument of the rate~ object. At slower tempos the effect is more rhythmic, and higher tempos will start to produce a timbral effect akin to ring modulation.

The oscillator type toggle box (see top right) allows you to use either a regular sine or a sine that has been distorted with rampsmooth~ as the basic waveform inside of the poly~ object.

This toggle applies to a specific instance (i.e. the last key you pressed) of the poly~ subpatcher. This way you can mix regular and distorted sines by affecting specific keys. Since the "Tempo" dial causes some sidebands/modulation when using higher values, the "sine" setting will be most sine-like with slower Tempo. The sine can give the occasional click with slow Tempo, which can be absorbed into the sound of the reverb object that connects to the Reverb wet/dry slider. In any case, there's a spectrum of timbral possibilities given this relatively simple poly~ patch and the controls of the interface!