

## txcttm

+ + + + +

this document contains:

1. first thing you may want to do
2. quickies
3. basic operations
4. sequencing
5. advanced operations

+ + + + +

1. first thing you may want to do:

type in [?] and press ENTER to see the help file.

+ + + + +

2. quickies:

set osc 1's amplitude to 1 and frequency to 2000Hz  
[ osc 1 a 1 f 2000 ]

sequencing osc 2's amplitude  
[ osc 2 a 1 0 1 0 1 0 0 0 ]

+ + + + +

3. basic operations:

txcttm has three kinds of sound generators:

[ osc ] - sine wave oscillator with frequency modulation  
[ wtl ] - wavetable synthesiser with frequency modulation  
[ sml ] - sampler

each kind has four generators defined by id number right next.  
for exmaple, if you want to call #2 of wavetable synthesiser [ wtl ], it would be,  
[ wtl 2 ]

each kind has its own unique set of parameters.

[ osc ]  
a = amplitude (0. - 1.)  
f = frequency (0. - 20000.)  
m = midi note number (0 - 127) (60 = center c)  
h = fm frequency rate (0. - oo)  
i = fm depth (0. - oo)  
p = panning (0. - 1.)

[ wtl ]  
a = amplitude (0. - 1.)  
f = frequency (0. - 20000.)  
m = midi note number (0 - 127) (60 = center c)  
h = fm frequency rate (0. - oo)  
i = fm depth (0. - oo)  
p = panning (0. - 1.)  
open = open waveform  
wf = waveform (sine, tri, saw, pulse, rand, glitch)

[ sml ]  
a = trigger and amplitude (0. - 1.)  
f = frequency (0. - 20000.)  
p = panning (0. - 1.)  
open = open waveform  
load = load sample

here are some examples:

```
set osc 1's amplitude to 1
[ osc 1 a 1 ]
set osc 2's amplitude to 0.5, frequency to 440, fm ratio to 2, fm depth to 0.5
[ osc 2 a 0.5 f 440 h 2 i 0.5 ]
set osc 3's frequency to midi note number 60 (center c), amplitude to 0.5, pan to the left
[ osc 3 m 60 a 0.5 p 0 ]
set wtl 1's amplitude to 1 and frequency to 5000Hz
[ wtl 1 a 1 f 5000 ]
show wtl 2's waveform
[ wtl 2 open ]
set wtl 3's waveform to pulse wave
[ wtl 3 wf pulse ]
basic sequencing with sml 1
[ sml 1 a 1 0 0 0 ]
more sequencing with sml 2
[ sml 2 a 1 0 1 0 1 0 0 0 f 0 0 -12 -12 p 0 1 ]
show sml 3's waveform
[ sml 3 open ]
load smaple to sml 4
[ sml 4 load kick1 ]
set bpm to 140
[ bpm 140 ]
redo operations 2 and 6
[ >> 2 6 ]
halt operations
[ hlt 1 ]
un-halt operations
[ hlt 0 ]
```

+ + + + +

#### 4. sequencing:

in txcttm, any parameter can be sequenced, while there are some exceptions.

a simple example would be amplitudes and frequencies.

```
[ osc 1 a 0.5 0 0 0 f 1000 2000 3000 4000 ]
```

notice in this example that you only hear 1000hz, because when osc 1 plays 2000hz, 3000hz and 4000hz, osc 1's amplitude is 0.

if you do this instead,

```
[ osc 1 a 0.5 0 0 0.7 0 f 1000 2000 3000 4000 ]
```

you will hear 1000hz and 3000hz alternatively while 3000hz being a bit louder.

now let's put different number of inputs,

```
[ osc 1 a 0.5 0 0 0.7 0 f 1000 2000 3000 ]
```

in this example you have 4 inputs for amplitude and 3 inputs for frequency.

the result you'll hear would be 1000hz - rest - 3000hz - rest - 2000hz - rest - 1000hz... and so on.

txcttm is, by design, easily accommodating to poly-rhythm.

for example, if you can create poly-rhythmic pattern within one sound generator.

```
[ osc 1 a 0.5 0 0 0 0.5 0 f 10000 h 1 0 0 1 0 i 0 0 0 5 0 3 1 0 0 0 0 ]
```

in this example, there are 6 inputs for amplitude, 5 inputs for fm rate, 11 inputs for fm depth, resulting in poly-rhythm of 6:5:11.

this idea can be effortlessly expanded to multiple sound generators.

for example,

```
[ osc 1 a 0.5 0 0 0 0 0.5 0 0.1 0 f 8000 p 0 ]
```

```
[ osc 2 a 0.5 0.5 0 0.5 0 0 0 0 0 f 9000 9000 4500 4500 p 1 ]
```

+ + + + +

#### 5. advanced operations:

redo-ing and halting

in txcttm, you may not control multiple sound generators at once.  
for exmaple, following command is NOT CORRECT.

```
[osc 1 a 0.5 0 0 0 osc 2 a 0.5 0 0.5 0 0]
```

in order to do the above, you use redo and halting.  
here is how.

```
[ hlt 1 ]  
[ osc 1 a 0.5 0 0 0 ]  
[ osc 2 a 0.5 0 0.5 0 0 ]  
[ hlt 0 ]  
[ >> 2 3 ]
```

the first line halts all following commands. i.e. line 2 and 3 are ignored.

line 4 undo the halting.

redo command (>>) redo the line 2 and 3 and this time they won't be ignored.

adding to sequence

extra inputs can be added to the end.

for example,

```
[ osc 1 a 0.5 ]
```

once this is established, if you put the following,

```
[ osc 1 0.1 ]
```

this will result in the same as

```
[ osc 1 a 0.5 0.1 ]
```

here's more complex example.

```
[ osc 1 f 10000 a 0.5 0 0 0.5 0 0 0.5 0 0.5 0 0 0 0.5 0 0 0 ]
```

```
[ osc 1 0.5 0 0 0.1 ]
```

these two lines will result in,

```
[ osc 1 f 10000 a 0.5 0 0 0.5 0 0 0.5 0 0.5 0 0 0 0.5 0 0 0 0.5 0 0 0.1 ]
```