

# **First Steps in *RTcmix*~**

**Including Connections to *bach* and *poly*~**

**A MaxMSP Patching Odyssey**

**ENT3320 — *Interactive VR, Music, Sound, and Memory***

**New York College of Technology, CUNY City Tech  
Spring 2020**

**Professor Louis Goldford  
[LGoldford@citytech.cuny.edu](mailto:LGoldford@citytech.cuny.edu)**



*Computer Music Center, Columbia University*



*Creative Commons  
Attribution-ShareAlike 4.0 International*

# CONTENTS

<b>First Steps in RTcmix~</b>	<b>1</b>
CONTENTS	2
<b>1. Installation</b>	<b>2</b>
1.1 Download & Install	2
1.2 Path to Max /Library Folder	4
<b>2. Basic Synthesis With RTcmix~</b>	<b>5</b>
2.1 A Single STRUM	5
2.2 Repetition with a FOR LOOP	6
2.3 Repeated Random Notes	7
2.4 Continuous External Control	8
2.5 Continuous Looping Via MAXBANG()	9
2.6 Controlling <i>RTcmix~</i> Variables	10
<b>3. The WAVETABLE Instrument</b>	<b>11</b>
3.1 A Single Sine Tone	11
3.2 Using Maketable	12
3.3 Additive Synthesis	13
<b>4. Connecting to <i>bach.roll</i> and <i>poly~</i></b>	<b>14</b>
4.1 Dynamic WAVETABLE Messages	14
4.2 <i>poly~</i> with <i>RTcmix~</i> Synthesis Engine	15
4.3 Writing an <i>RTcmix~</i> Script	16
4.3.1 <i>poly~</i> Subpatch	17
4.3.2 Initialize.script Subpatch	18
4.3.3 Contents of <Text> Object	19
4.3.4 Running an <i>RTcmix</i> Script in Terminal	20
4.3.5 Example Terminal Output	21

# 1. Installation

## 1.1 Download & Install

The screenshot shows a web browser window displaying the RTcmix website. The URL is [rtcmix.org/rtcmix~/](http://rtcmix.org/rtcmix~/). The page title is "RTcmix". Below it, the sub-page title is "rtcmix~". The main content area contains text about RTcmix and its capabilities, followed by a list of download links for 32-bit and 64-bit versions. To the right, there is a sidebar with information about availability for Pd and links to binaries and source files. At the bottom, there is a note about removing older folders. Two red arrows point from the text below to specific parts of the website: one arrow points to the first download link, and another arrow points to the note about removing older folders.

RTcmix - rtcmix~

Not Secure | rtcmix.org/rtcmix~/

Home · Standalone · rtcmix~ · iRTcmix · Tutorials · Reference

# RTcmix

an open-source digital signal processing and sound synthesis language

about · links · contact

## rtcmix~

RTcmix is a complete sound synthesis and signal processing language, including a robust scheduler and large set of pre-compiled "instruments". The rtcmix~ object completely encapsulates RTcmix within the Max/MSP real-time music environment, extending the capabilities of Max/MSP.

- Download (32 bit): [RTcmix-2.00.zip](#)
- Download (64 bit): [RTcmix-2.099-64.zip](#)

rtcmix~ is also available for the Pd visual programming language, ported by [Joel Matthys](#)

- Binaries for OSX and Linux
- source files

Current versions are available for Max 5 (and above) for Mac OS X. Older versions are available [in the archives](#).

### Installation Instructions for Max 7 and 8

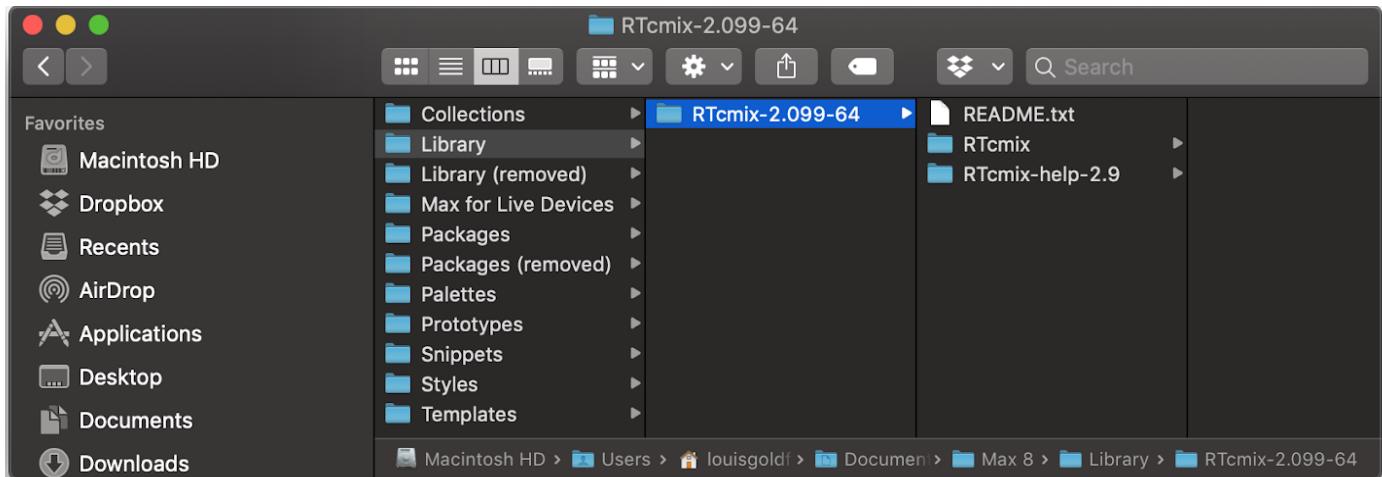
The archive will unpack to a folder titled "RTcmix-2.0xx" (with "-64" appended if it is the 64-bit version). Move this entire folder intact to the "/Users/Shared/Max 7/Library" or "/Users/Shared/Max 8/Library" folder. Restart Max/MSP and the new [rtcmix~] object should show in the Max console as:

```
RTcmix music language, v. 2.0xx (RTcmix-maxmsp-4.3.1)
```

**Note:** Be sure to remove any of your older RTcmix or RTcmix-help folders from these folders:  
/Users/Shared/Max 7/Library  
/Users/Shared/Max 8/Library

If you're running Max 7 on a City Tech laptop, chances are you will need the 32-bit version of RTcmix (first arrow points to it above).

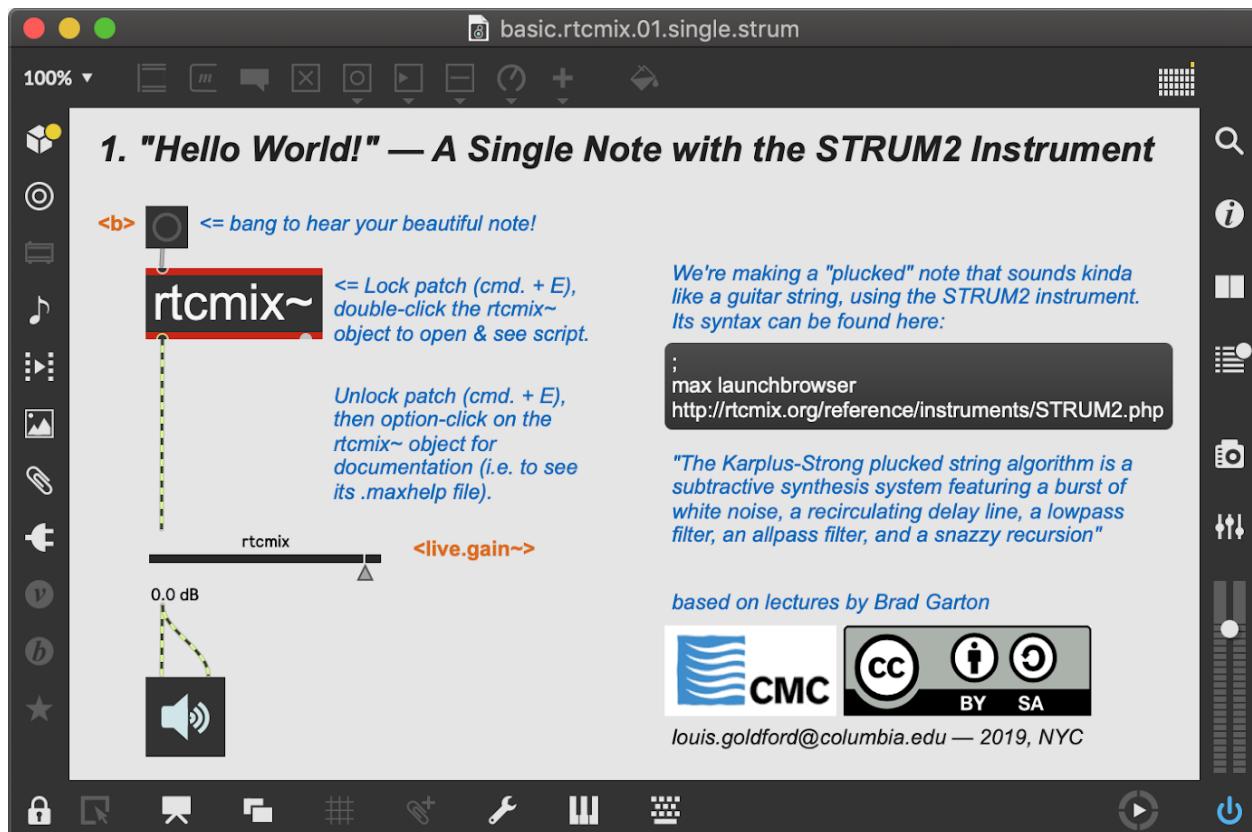
## 1.2 Path to Max /Library Folder



Unzip the RTcmix folder and place it your Max Library folder as shown above.

## 2. Basic Synthesis With *RTcmix~*

### 2.1 A Single STRUM

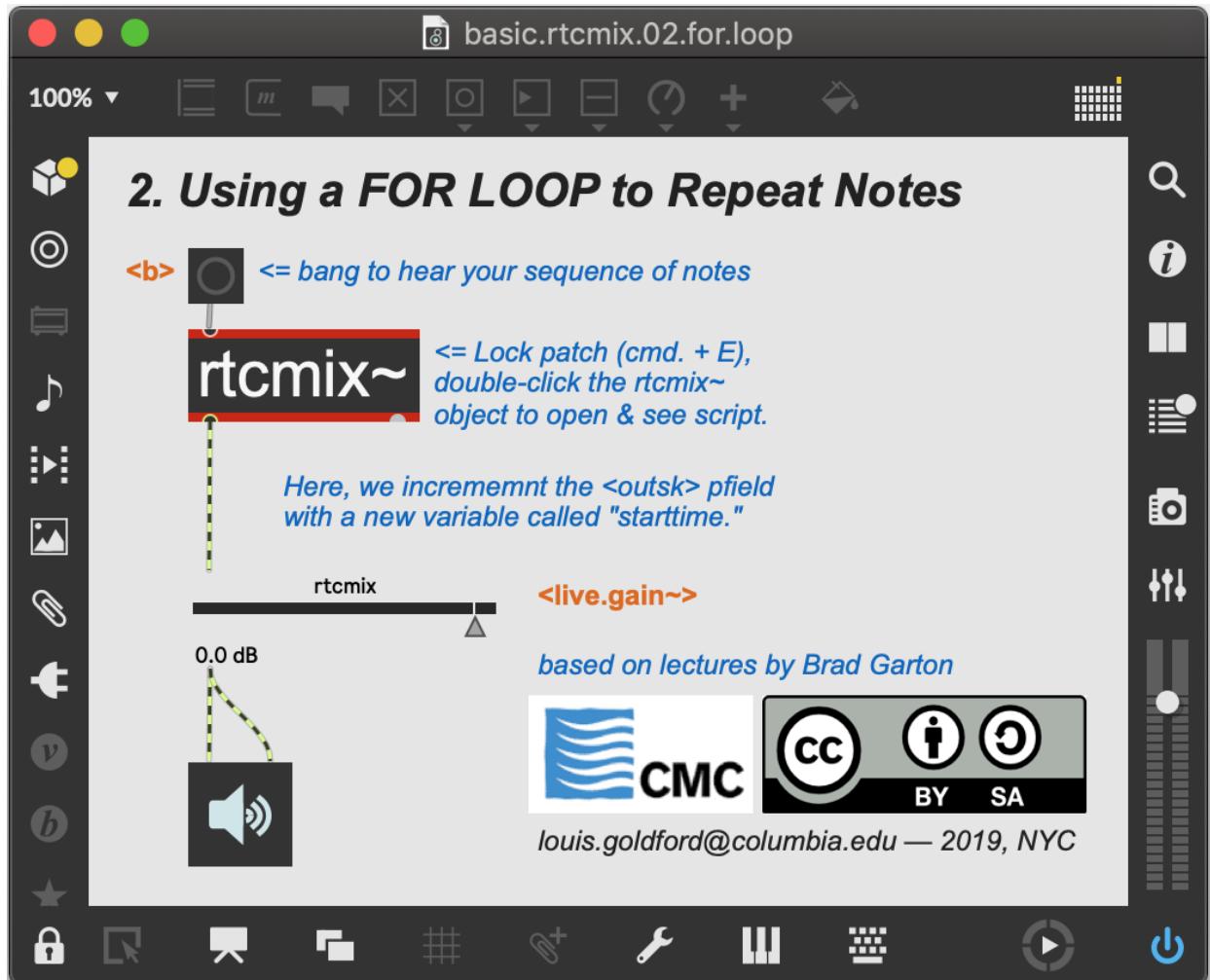


The screenshot shows the script editor window titled "script\_0" containing the following RTcmix code:

```
1 // STRUM2 Instrument Syntax:  
2 // STRUM2(outsk, dur, AMP, PITCH, squish, decay_time[, PAN])  
3 // source: http://rtcmix.org/reference/instruments/STRUM2.php  
4  
5 STRUM2(0, 3.5, 20000, 443.9, 2, 3.5)  
6
```

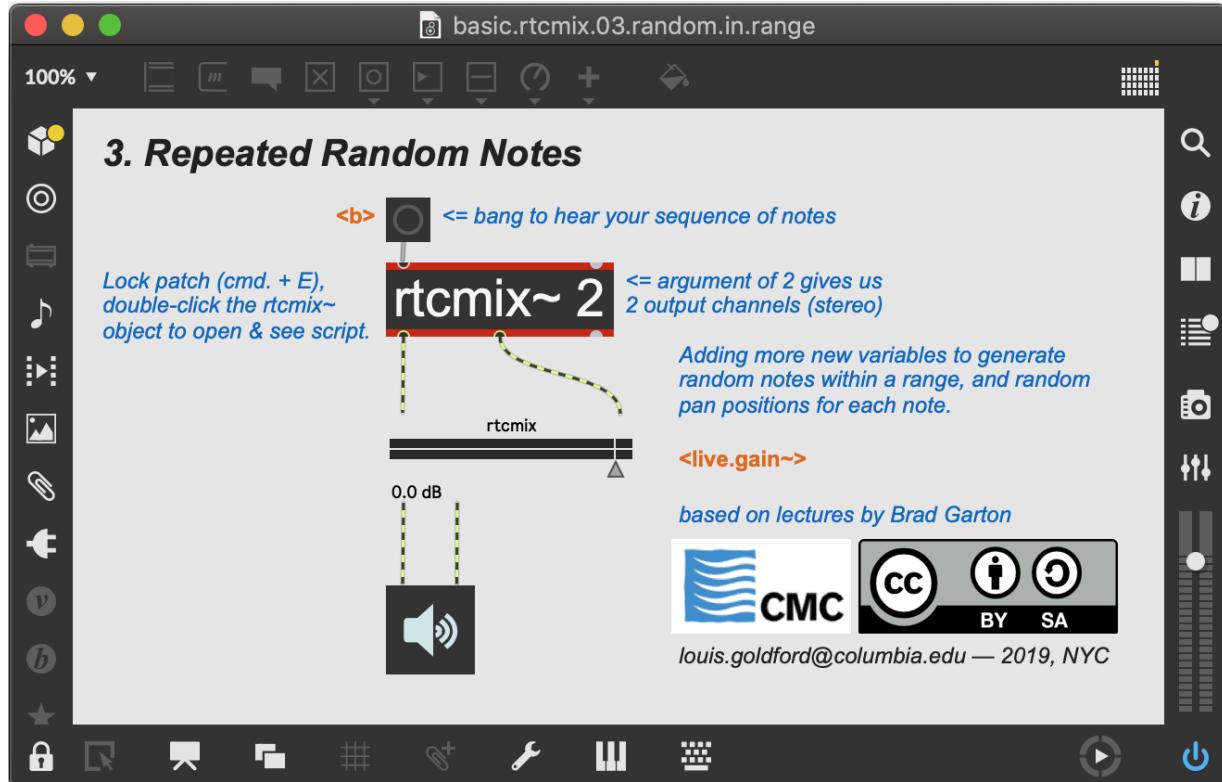
At the bottom of the script editor, the status bar displays "Cursor Line: 5 Insertion Point Line: 1".

## 2.2 Repetition with a FOR LOOP



Cursor Line: 7 Insertion Point Line: 1

## 2.3 Repeated Random Notes

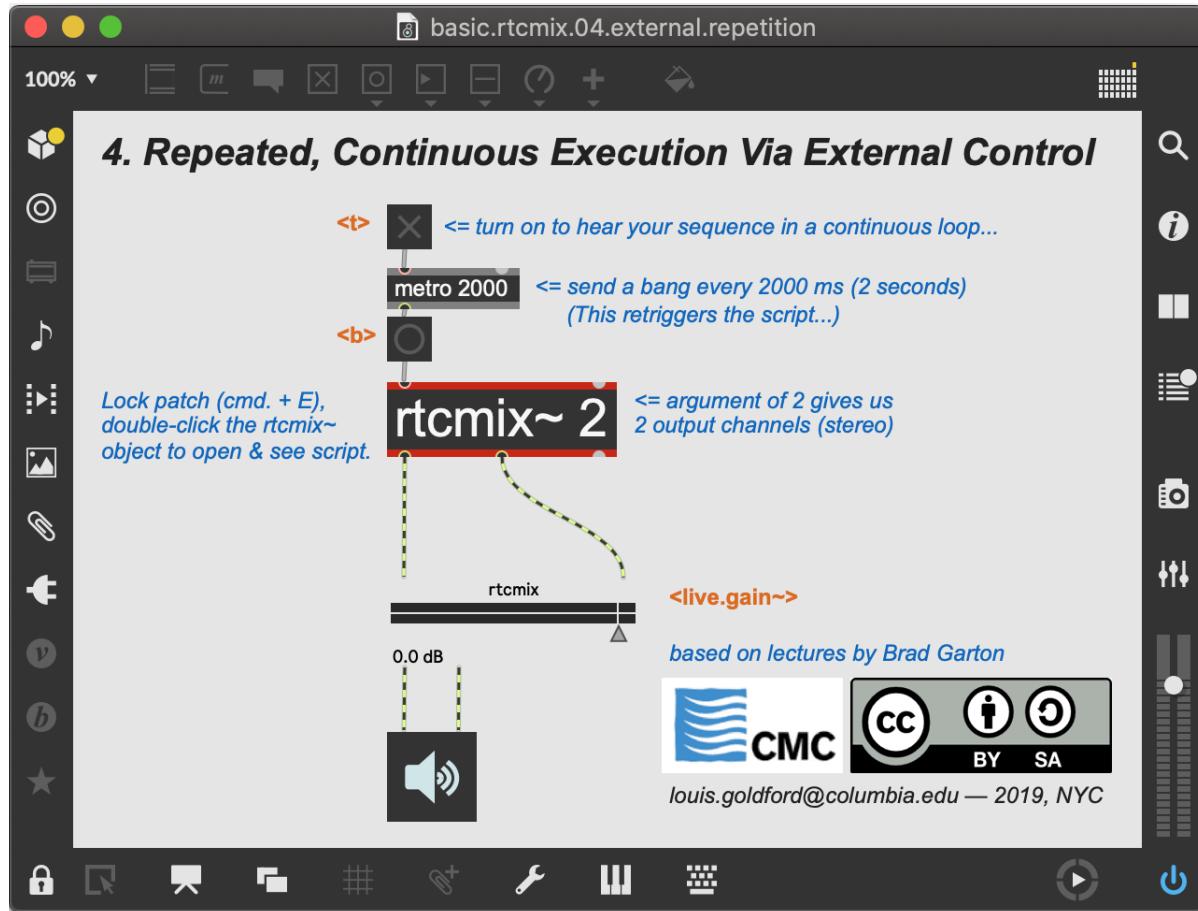


```
1 // STRUM2 Instrument Syntax:  
2 // STRUM2(outsk, dur, AMP, PITCH, squish, decay_time[, PAN])  
3 // source: http://rtcmix.org/reference/instruments/STRUM2.php  
4  
5 starttime = 0  
6  
7 for (i = 0; i < 20; i = i+1) {  
8     dillon = irand(300, 700) // random number within range of 300-700 hz  
9     STRUM2(starttime, 3.5, 20000, dillon, 2, 3.5, random())  
10    starttime = starttime + 0.5  
11 }  
12  
13 // Notice we're also choosing random pan positions (in range 0.0-1.0)
```

script\_0

Cursor Line: 10 Insertion Point Line: 1

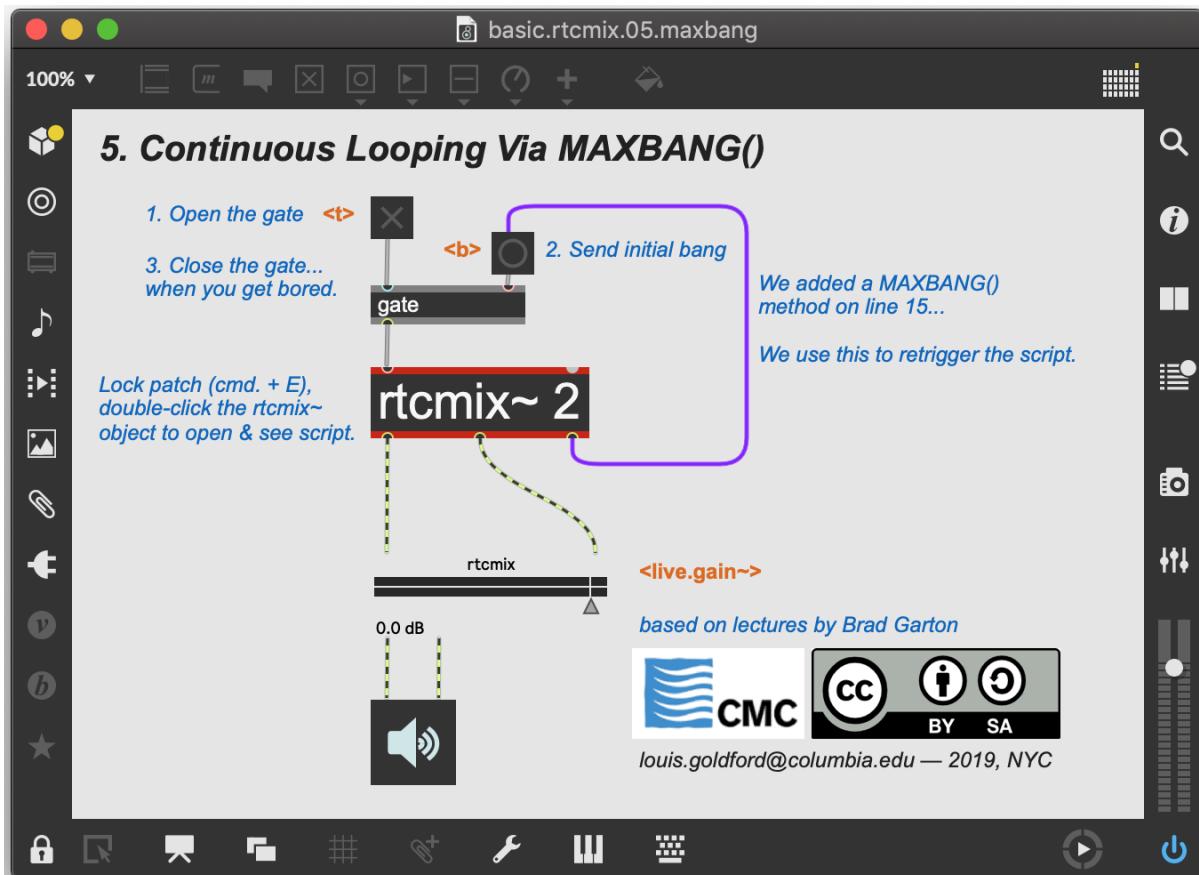
## 2.4 Continuous External Control



```
1 // STRUM2 Instrument Syntax:  
2 // STRUM2(outsk, dur, AMP, PITCH, squish, decay_time[, PAN])  
3 // source: http://rtcmix.org/reference/instruments/STRUM2.php  
4  
5 starttime = 0  
6  
7 for (i = 0; i < 20; i = i+1) {  
8     dillon = irand(300, 700) // random number within range of 300-700 hz  
9     STRUM2(starttime, 3.5, 20000, dillon, 2, 3.5, random())  
10    starttime = starttime + 0.1 // speed up the sequence!  
11 }  
12  
13 // Notice we're also choosing random pan positions (in range 0.0-1.0)
```

Cursor Line: 9 Insertion Point Line: 1

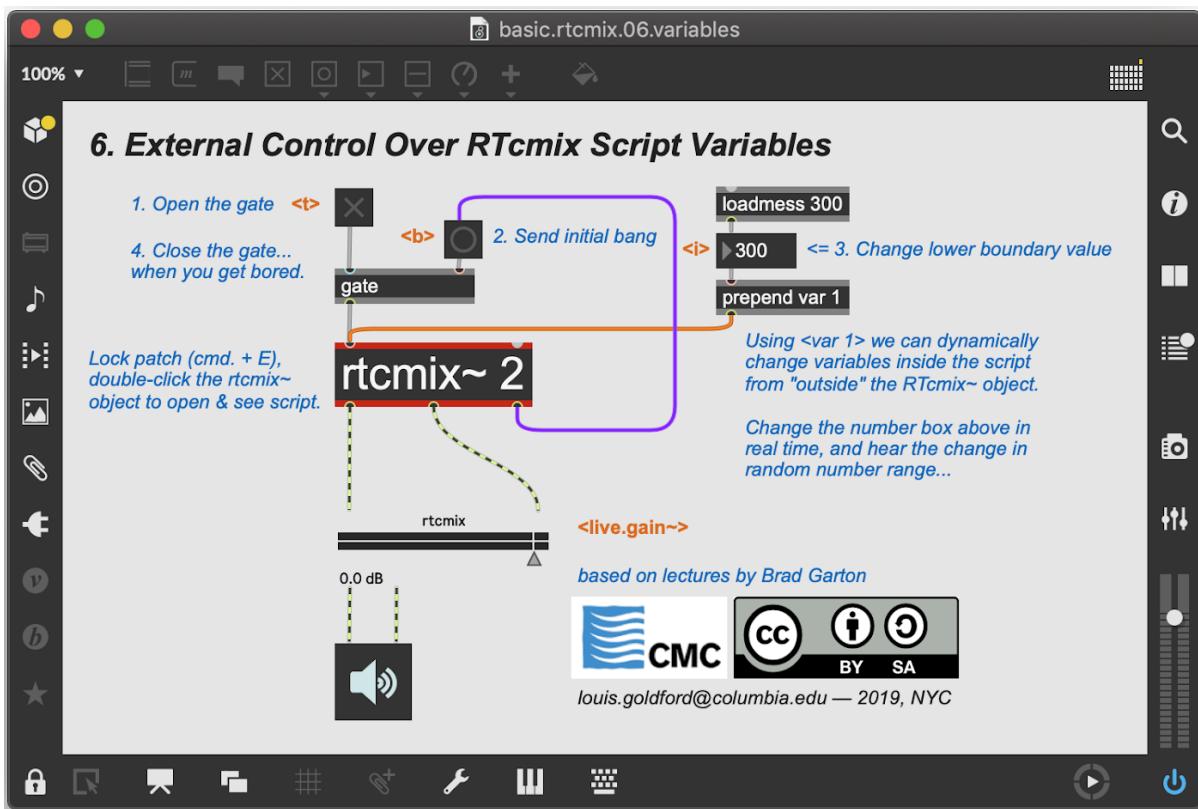
## 2.5 Continuous Looping Via MAXBANG()



```
// STRUM2 Instrument Syntax:  
// STRUM2(outsk, dur, AMP, PITCH, squish, decay_time[, PAN])  
// source: http://rtcmix.org/reference/instruments/STRUM2.php  
  
starttime = 0  
  
for (i = 0; i < 20; i = i+1) {  
    dillon = irand(300, 700) // random number within range of 300-700 hz  
    STRUM2(starttime, 3.5, 20000, dillon, 2, 3.5, random())  
    starttime = starttime + 0.1 // speed up the sequence!  
}  
  
// Notice we're also choosing random pan positions (in range 0.0-1.0)  
  
MAXBANG(starttime)
```

Cursor Line: 10 Insertion Point Line: 1

## 2.6 Controlling *RTcmix~* Variables



```
script_0
1 // STRUM2 Instrument Syntax:
2 // STRUM2(outsk, dur, AMP, PITCH, squish, decay_time[, PAN])
3 // source: http://rtcmix.org/reference/instruments/STRUM2.php
4
5 starttime = 0
6 lowedge = $1 // <var 1> in parent patch
7
8 for (i = 0; i < 20; i = i+1) {
9     dillon = irand(lowedge, 700) // replaced lower bound with new variable
10    STRUM2(starttime, 3.5, 20000, dillon, 2, 3.5, random())
11    starttime = starttime + 0.1 // speed up the sequence!
12 }
13
14 // Notice we're also choosing random pan positions (in range 0.0-1.0)
15
16 MAXBANG(starttime)
```

Cursor Line: 11 Insertion Point Line: 1

# 3. The WAVETABLE Instrument

## 3.1 A Single Sine Tone

The screenshot shows a Max/MSP patch window titled "basic.rtcmix.07.single.sine". The patch itself is titled "7. A Single Sine Wave using the WAVETABLE Instrument". It features a central "rtcmix~ 2" object. A "bang" message (indicated by a small circle with an arrow) is connected to its left inlet. A slider labeled "rtcmix" is connected to its right outlet. A "live.gain~" object is connected to the "rtcmix~" object. A speaker icon is connected to the output of the "live.gain~" object. A text box on the left says "Lock patch (cmd + E), double-click the rtcmix~ object to open & see script.". Another text box on the right contains the following text:

We're making a basic sine wave tone using the WAVETABLE instrument, whose documentation can be found here:

```
max launchbrowser  
http://rtcmix.org/reference/instruments/WAVETABLE.php
```

WAVETABLE is one of the most basic synthesis instruments available in RTcmix. It works by using a 'wavetable' template (usually constructed using the maketable("wave" ...) scorefile command) to determine the waveform used to create the sound. If no template is given (p5), then an internal sine-wave template is used.

based on lectures by Brad Garton

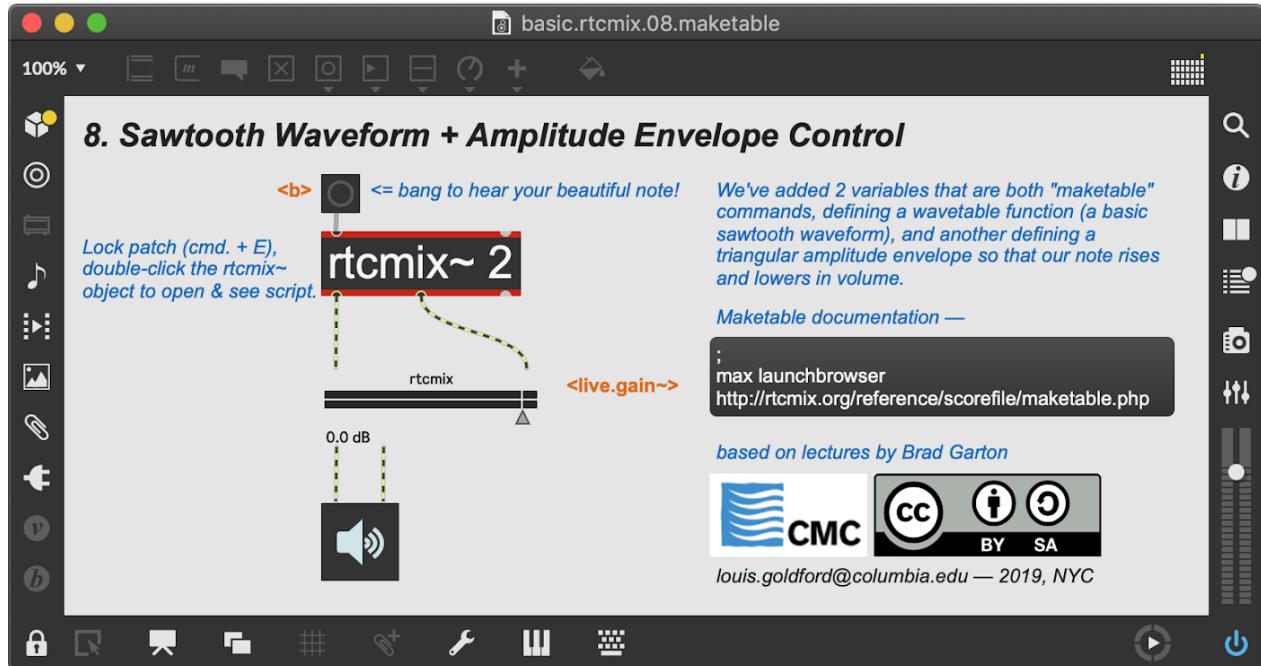
CMC CC BY SA

louis.goldford@columbia.edu — 2019, NYC

```
script_0
1 // WAVETABLE Instrument Syntax:
2 // WAVETABLE(outsk, dur, AMP, PITCH[, PAN, WAVETABLE])
3 // source: http://rtcmix.org/reference/instruments/WAVETABLE.php
4
5
6 WAVETABLE(0, 4.3, 20000, 8.02, 0.5)
7
8
9 // We didn't include the optional "wavetable" pfield,
10 // but we DID include the PAN pfield; a value of 0.5.
11 // This means we split equally between L/R channels.
12
13 // Also notice that our pitch syntax is not in hertz.
14 // We're now using oct.pc (octave - pitch-class).
15 // So, 8.02 means 8th octave (middle C) + plus 2 half-steps.
16 // In other words, we've defined the D in the 8th octave.
```

Cursor Line: 13 Insertion Point Line: 1

## 3.2 Using Maketable

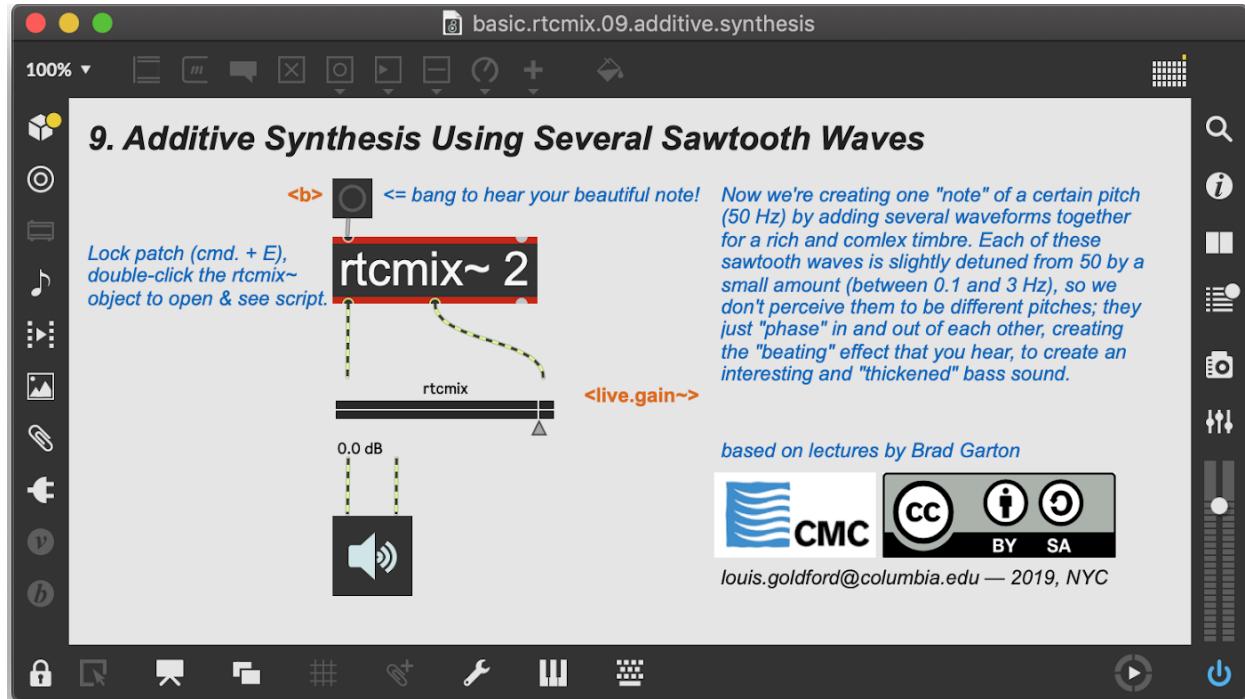


The screenshot shows the Max/MSP script editor with a dark background. The title bar says "script\_0". The code area contains the following Max/MSP script:

```
1 // WAVETABLE Instrument Syntax:  
2 // WAVETABLE(outsk, dur, AMP, PITCH[, PAN, WAVETABLE])  
3 // source: http://rtcmix.org/reference/instruments/WAVETABLE.php  
4  
5 // Define a wavetable as a sawtooth waveform:  
6 keon = maketable("wave", 1000, "saw")  
7  
8 // Define a triangular amplitude envelope:  
9 ampenv = maketable("line", 1000, 0, 0, 2.5, 1, 4.3, 0)  
10  
11  
12 WAVETABLE(0, 4.3, 20000*ampenv, 8.02, 0.5, keon)  
13  
14 // Multiply ampenv by constant 20000 to get real amplitude!  
15
```

At the bottom, it says "Cursor Line: 9 Insertion Point Line: 1".

### 3.3 Additive Synthesis



```
script_0
1 // WAVETABLE Instrument Syntax:
2 // WAVETABLE(outsk, dur, AMP, PITCH[, PAN, WAVEFILE])
3 // source: http://rtcmix.org/reference/instruments/WAVETABLE.php
4
5 // Define a wavetable as a sawtooth waveform:
6 keon = maketable("wave", 1000, "saw")
7
8 // Define a triangular amplitude envelope:
9 ampenv = maketable("line", 1000, 0, 0, 50, 1, 100, 0)
10
11
12 WAVETABLE(0, 7, 10000*ampenv, 50, 0.5, keon)
13
14 WAVETABLE(0, 7, 10000*ampenv, 50+irand(0.1, 3), 0.5, keon)
15
16 WAVETABLE(0, 7, 10000*ampenv, 50+irand(0.1, 3), 0.5, keon)
17
18 WAVETABLE(0, 7, 10000*ampenv, 50+irand(0.1, 3), 0.5, keon)
19
```

Cursor Line: 11 Insertion Point Line: 1

# 4. Connecting to *bach.roll* and *poly~*

## 4.1 Dynamic WAVETABLE Messages

basic.rtc当地.10.bach

### 10. Monophonic Playback Using Bach.roll and Dynamic WAVETABLE Messages

Every time audio is turned on or off, the RTcmix queue and data get reinitialized. This means that you need to resend any RTcmix data (such as wavetable information, amplitude envelopes, etc.) if they are not contained in the message or score being sent. The [delay] is to ensure that the RTcmix queue is working.

rtcmix makegen 1 24 1000 0 0 1 1 2 0,  
rtcmix makegen 2 10 1000 1 0.1 0.,  
rtcmix reset 2000

Watch the Max console: print dynamic

bach.playkeys cents velocity duration @out t  
<midicents> pitch      <velocity> volume      <duration> (ms)  
bach.mc2f @out t      scale 0 127 0 32768      \* 0.001  
to frequency (Hz)      to 16-bit wordlength      to seconds

pack f i f

rtcmix WAVETABLE 0 \$3 \$2 \$1

the max/msp message "rtcmix" can be used to send script information to rtcmix-

rtcmix~ <= RTcmix~ with no hard-coded script inside! (double-click to see NOTHINGNESS...)

live.gain~

0.0 dB

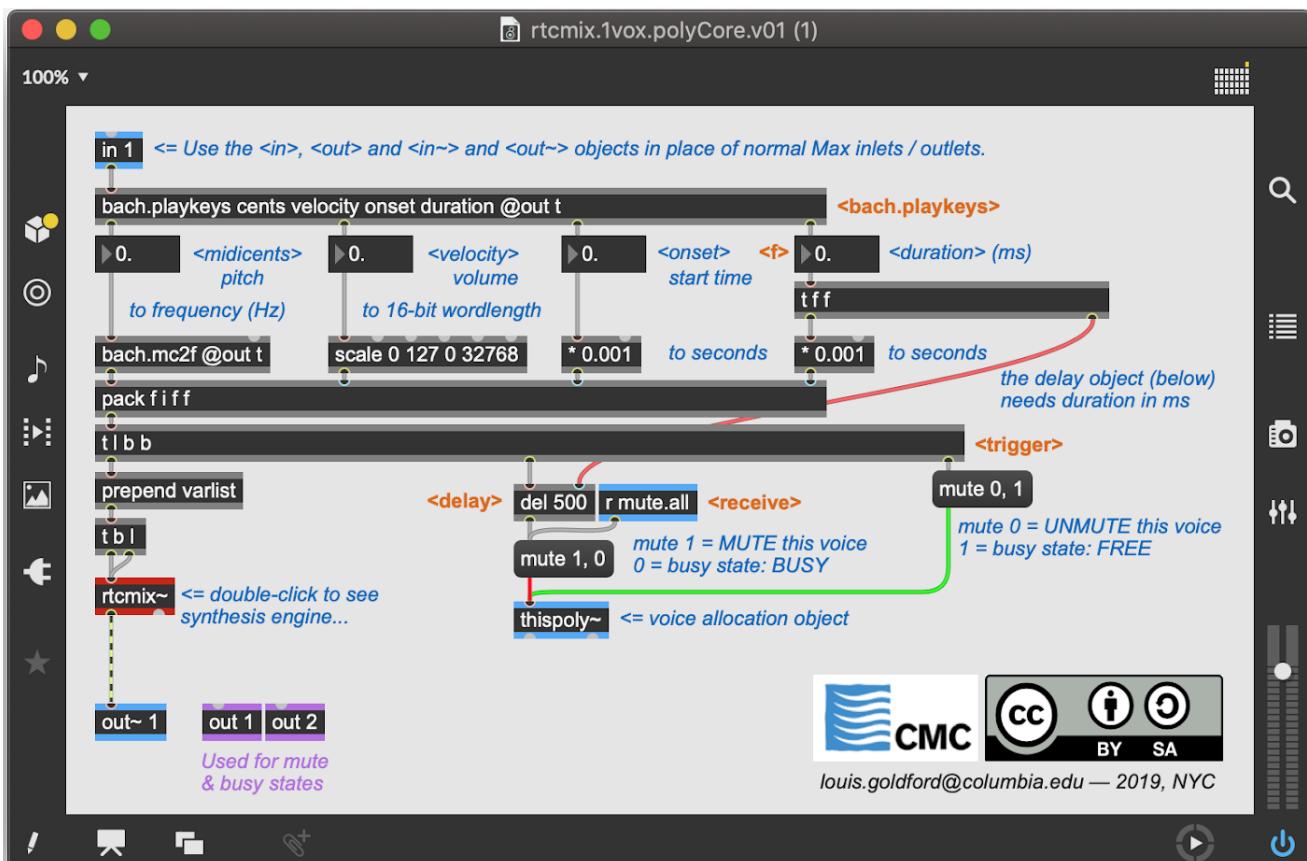
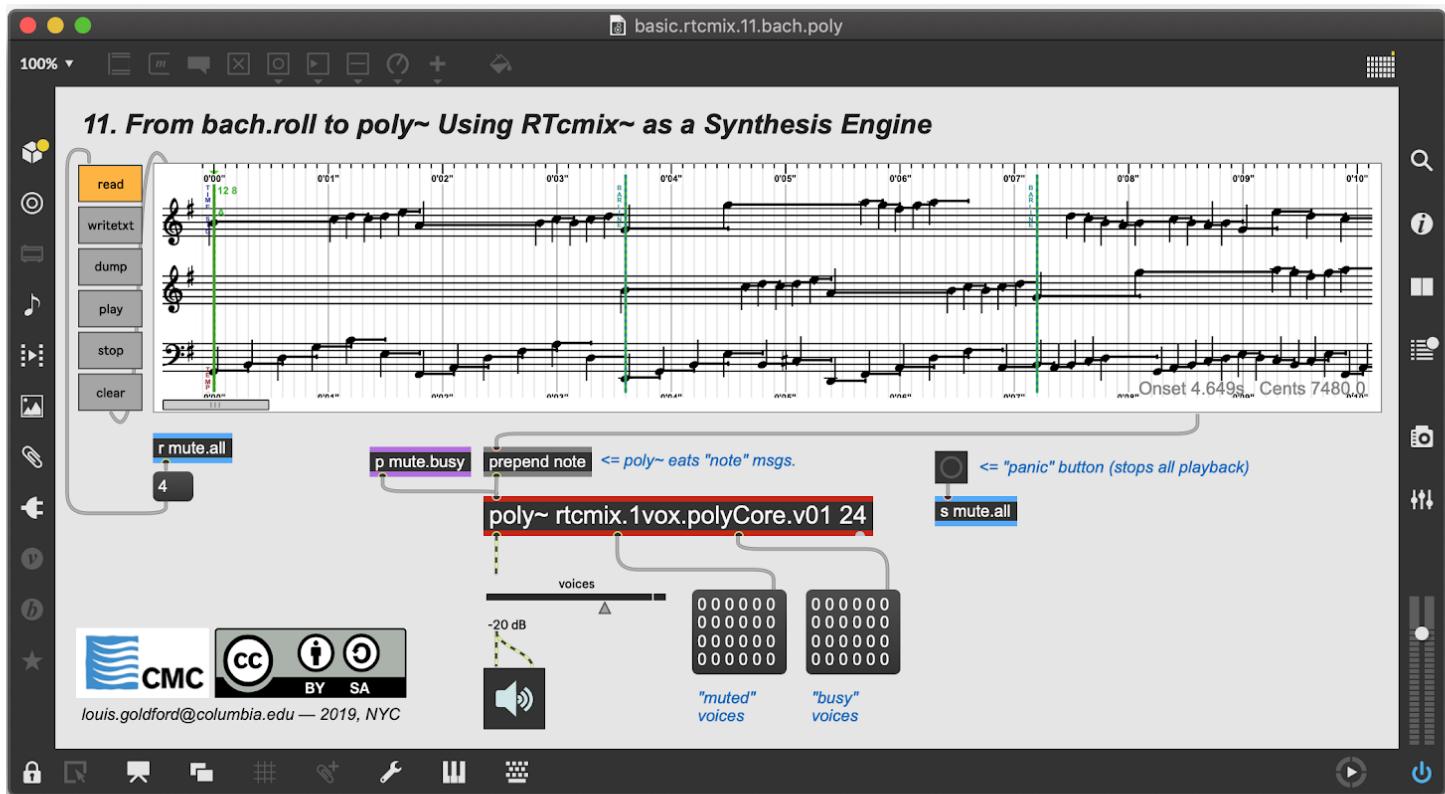
WAVETABLE(outsk, dur, AMP, PITCH[, PAN, WAVE...]  
\$3 \$2 \$1

source: <http://rtcmix.org/reference/instruments/WAVETABLE.php>

Remember: you can always find the syntax you need by browsing the RTcmix website:  
max launchbrowser <http://rtcmix.org/reference/>

CMC CC BY SA  
louis.goldford@columbia.edu — 2019, NYC

## 4.2 poly~ with RTcmix~ Synthesis Engine



## 4.3 Writing an RTcmix~ Script

basic.rtcmix.12.write.script

12. Write Your Sequence to an RTcmix~ Script

Onset 5.045s Cents 4380.0

<= poly~ eats "note" msgs.

<= bang to format RTcmix script

<= double-click to see how!

<= after it finished playing, write your sequence to a .txt file that can be played outside of Max...

<= (double-click!) to see collection of all WAVETABLE messages representing each note in the bach.roll; as indiv. lines in an RTcmix script to be run outside of Max...

To get our <bach.roll> sequence and our <RTcmix~> sounds outside of Max, and eventually into Unity, we need to write a complete RTcmix score as a text file.

See the "initialize.script" subpatch and the poly~ to see how this RTcmix score is formatted. Watch the <text> object collect each line of code, and finally, write the contents to a text file.

If you have the Standalone version of RTcmix installed on your computer (rtcmix.org/standalone/) you can run the text file at the command prompt in your terminal by typing the following:

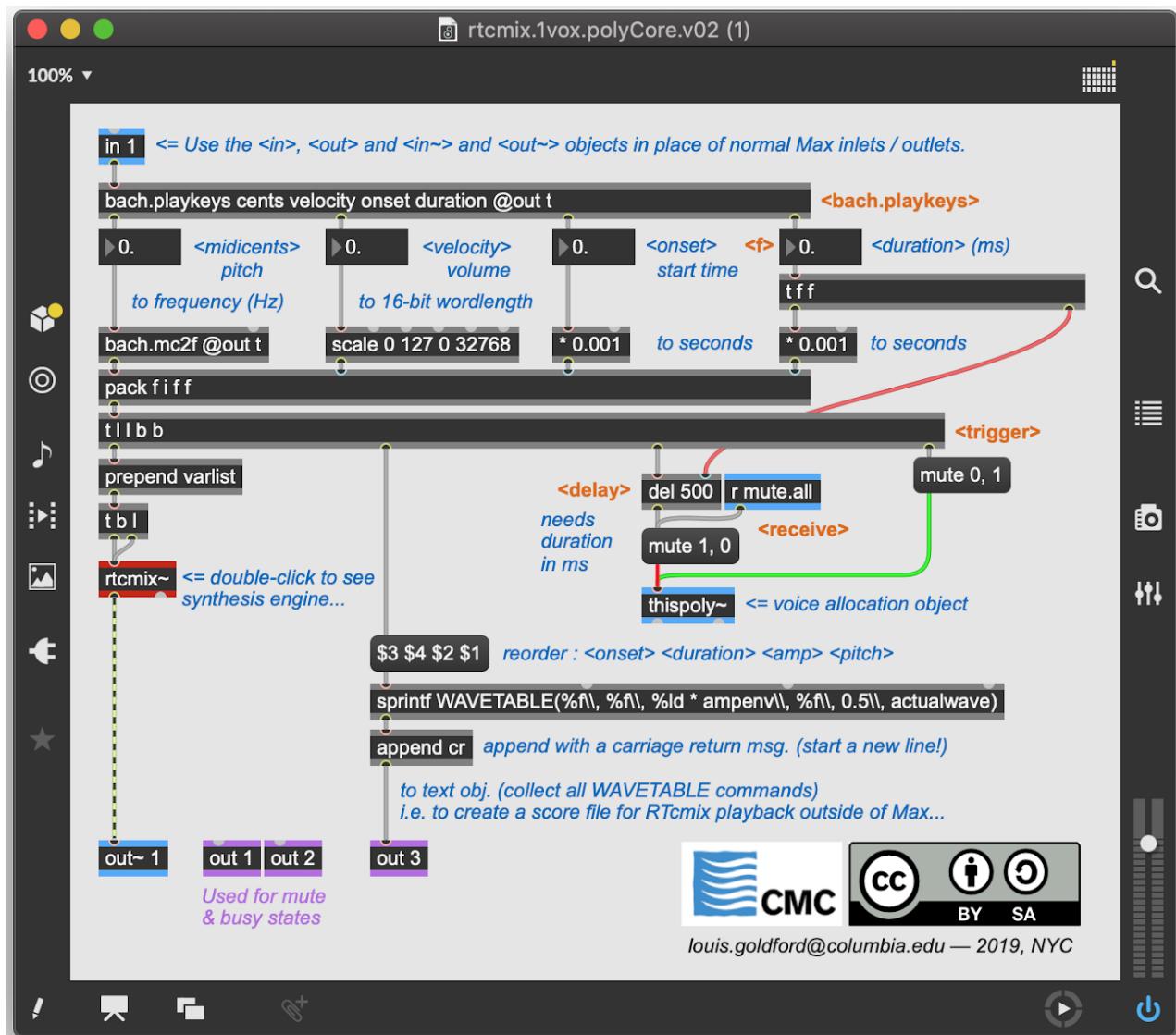
cmix < [path to text file]

louis.goldford@columbia.edu — 2019, NYC

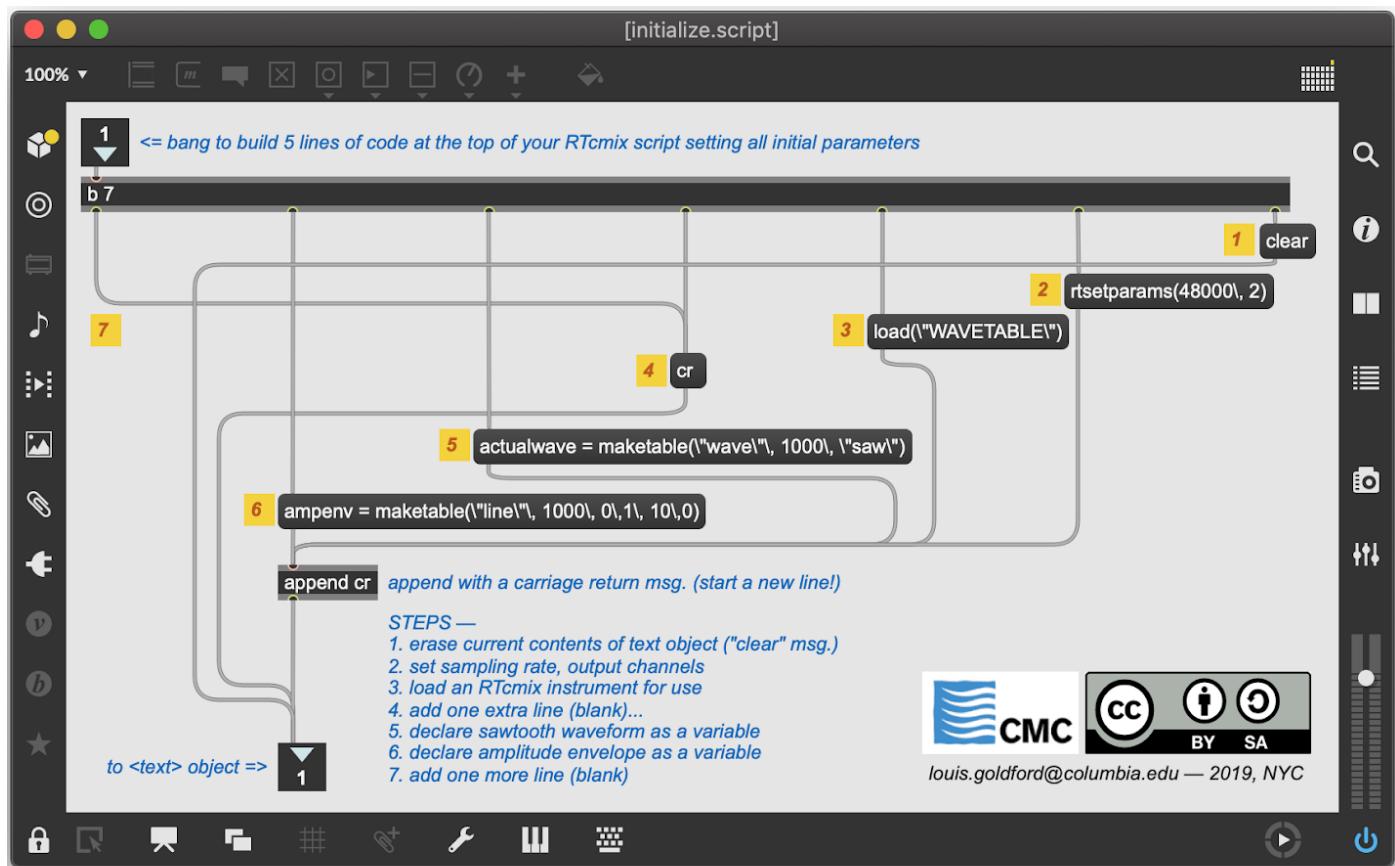
CMC

CC BY SA

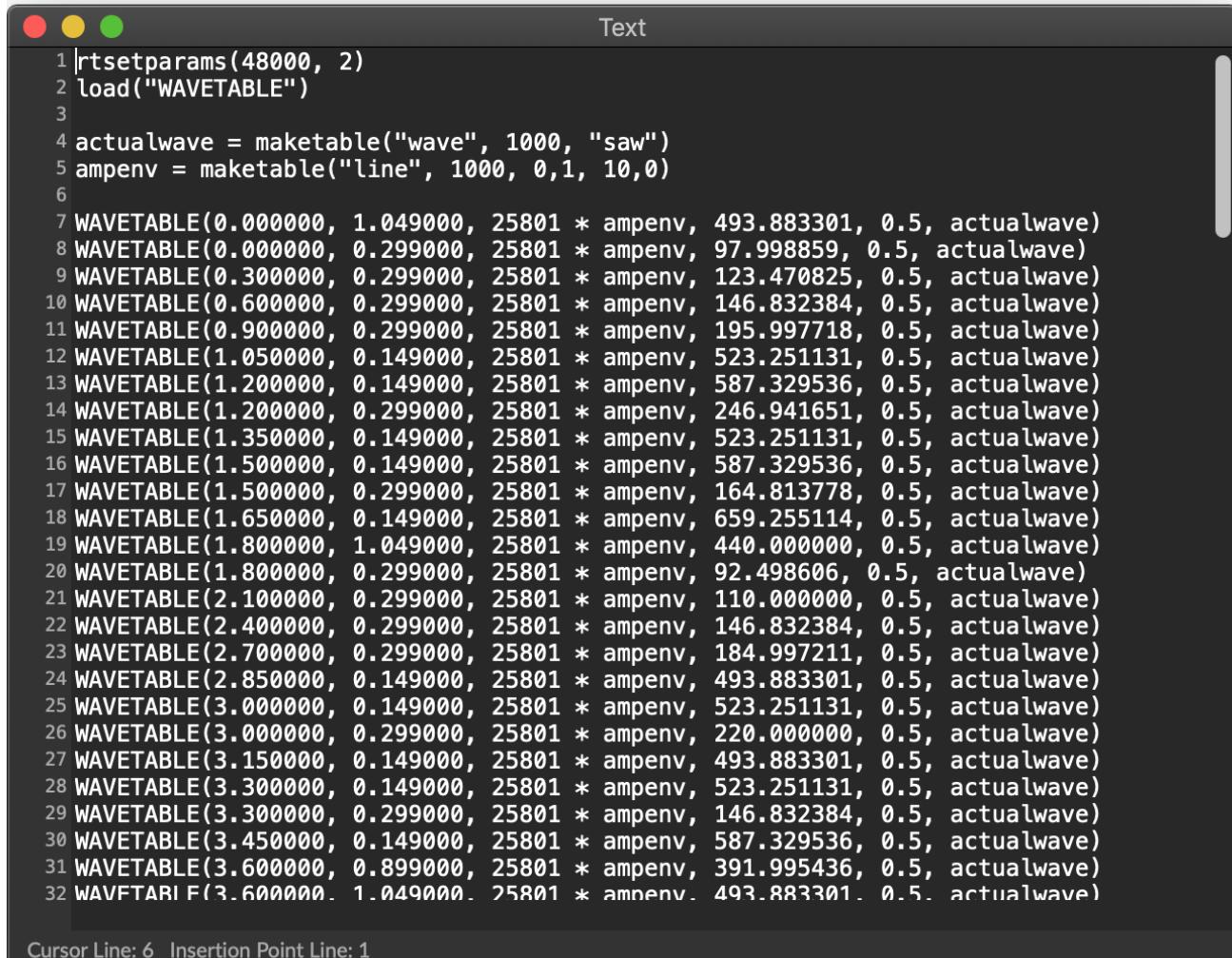
## 4.3.1 poly~ Subpatch



## 4.3.2 Initialize.script Subpatch



### 4.3.3 Contents of <Text> Object



```
1 |rtsetparams(48000, 2)
2 load("WAVETABLE")
3
4 actualwave = maketable("wave", 1000, "saw")
5 ampenv = maketable("line", 1000, 0,1, 10,0)
6
7 WAVEFILE(0.000000, 1.049000, 25801 * ampenv, 493.883301, 0.5, actualwave)
8 WAVEFILE(0.000000, 0.299000, 25801 * ampenv, 97.998859, 0.5, actualwave)
9 WAVEFILE(0.300000, 0.299000, 25801 * ampenv, 123.470825, 0.5, actualwave)
10 WAVEFILE(0.600000, 0.299000, 25801 * ampenv, 146.832384, 0.5, actualwave)
11 WAVEFILE(0.900000, 0.299000, 25801 * ampenv, 195.997718, 0.5, actualwave)
12 WAVEFILE(1.050000, 0.149000, 25801 * ampenv, 523.251131, 0.5, actualwave)
13 WAVEFILE(1.200000, 0.149000, 25801 * ampenv, 587.329536, 0.5, actualwave)
14 WAVEFILE(1.200000, 0.299000, 25801 * ampenv, 246.941651, 0.5, actualwave)
15 WAVEFILE(1.350000, 0.149000, 25801 * ampenv, 523.251131, 0.5, actualwave)
16 WAVEFILE(1.500000, 0.149000, 25801 * ampenv, 587.329536, 0.5, actualwave)
17 WAVEFILE(1.500000, 0.299000, 25801 * ampenv, 164.813778, 0.5, actualwave)
18 WAVEFILE(1.650000, 0.149000, 25801 * ampenv, 659.255114, 0.5, actualwave)
19 WAVEFILE(1.800000, 1.049000, 25801 * ampenv, 440.000000, 0.5, actualwave)
20 WAVEFILE(1.800000, 0.299000, 25801 * ampenv, 92.498606, 0.5, actualwave)
21 WAVEFILE(2.100000, 0.299000, 25801 * ampenv, 110.000000, 0.5, actualwave)
22 WAVEFILE(2.400000, 0.299000, 25801 * ampenv, 146.832384, 0.5, actualwave)
23 WAVEFILE(2.700000, 0.299000, 25801 * ampenv, 184.997211, 0.5, actualwave)
24 WAVEFILE(2.850000, 0.149000, 25801 * ampenv, 493.883301, 0.5, actualwave)
25 WAVEFILE(3.000000, 0.149000, 25801 * ampenv, 523.251131, 0.5, actualwave)
26 WAVEFILE(3.000000, 0.299000, 25801 * ampenv, 220.000000, 0.5, actualwave)
27 WAVEFILE(3.150000, 0.149000, 25801 * ampenv, 493.883301, 0.5, actualwave)
28 WAVEFILE(3.300000, 0.149000, 25801 * ampenv, 523.251131, 0.5, actualwave)
29 WAVEFILE(3.300000, 0.299000, 25801 * ampenv, 146.832384, 0.5, actualwave)
30 WAVEFILE(3.450000, 0.149000, 25801 * ampenv, 587.329536, 0.5, actualwave)
31 WAVEFILE(3.600000, 0.899000, 25801 * ampenv, 391.995436, 0.5, actualwave)
32 WAVEFILE(3.600000, 1.049000, 25801 * ampenv, 493.883301, 0.5, actualwave)
```

Cursor Line: 6 Insertion Point Line: 1

#### **4.3.4 Running an RTcmix Script in Terminal**

*If you have the Standalone version of RTcmix installed on your computer (<http://rtcmix.org/standalone/>), you can run the text file at the command prompt in your terminal by typing the following:*

**cmix < [path to text file]**



```
(base) Louiss-MBP:~ louisgoldford$ cmix < /Users/louisgoldford/Desktop/ENT3320-Louis/tutorial.02.basic.rtcmix/my.incredible.composition.v01.txt
```

### 4.3.5 Example Terminal Output

```
Louiss-MBP:~ louisgoldford$ cmix < /Users/louisgoldford/Desktop/ENT3320-Louis/tutorial.02.basic.rtcmix/my.incredible.composition.v01.txt

-----> RTcmix 4.2.1 (cmix) <-----
=====
rtsetparams: 48000 2
Audio set: 48000 sampling rate, 2 channels

=====
load: "WAVETABLE"
load: Loaded RT functions from shared library:
      '/Users/Shared/RTcmix-4.1/shlib/libWAVETABLE.so'.
=====

maketable: "wave" 1000 "saw"
=====
maketable: "line" 1000 0 1 10 0
=====<rt-queueing>=====
WAVETABLE: 0 1.049 PF:[25801,...,0] 493.883 0.5 PF:[1,...,-0.998]
No bus_config defined, setting default (in/out).
default: () => WAVETABLE => (out 0-1)

=====<rt-queueing>=====
WAVETABLE: 0 0.299 PF:[25801,...,0] 97.9989 0.5 PF:[1,...,-0.998]
=====<rt-queueing>=====
WAVETABLE: 0.3 0.299 PF:[25801,...,0] 123.471 0.5 PF:[1,...,-0.998]
=====<rt-queueing>=====
WAVETABLE: 0.6 0.299 PF:[25801,...,0] 146.832 0.5 PF:[1,...,-0.998]
=====<rt-queueing>=====
WAVETABLE: 0.9 0.299 PF:[25801,...,0] 195.998 0.5 PF:[1,...,-0.998]
=====<rt-queueing>=====
WAVETABLE: 1.05 0.149 PF:[25801,...,0] 523.251 0.5 PF:[1,...,-0.998]
=====<rt-queueing>=====
WAVETABLE: 1.2 0.149 PF:[25801,...,0] 587.33 0.5 PF:[1,...,-0.998]
=====<rt-queueing>=====
WAVETABLE: 1.2 0.299 PF:[25801,...,0] 246.942 0.5 PF:[1,...,-0.998]
=====<rt-queueing>=====
WAVETABLE: 1.35 0.149 PF:[25801,...,0] 523.251 0.5 PF:[1,...,-0.998]
closing...
Output duration: 2.87 seconds

Peak amplitudes of output:
  channel 0: 25801.000000 ( -2.08 dBFS) at frame 0 (0 seconds)
  channel 1: 25801.000000 ( -2.08 dBFS) at frame 0 (0 seconds)

(base) Louiss-MBP:~ louisgoldford$
```