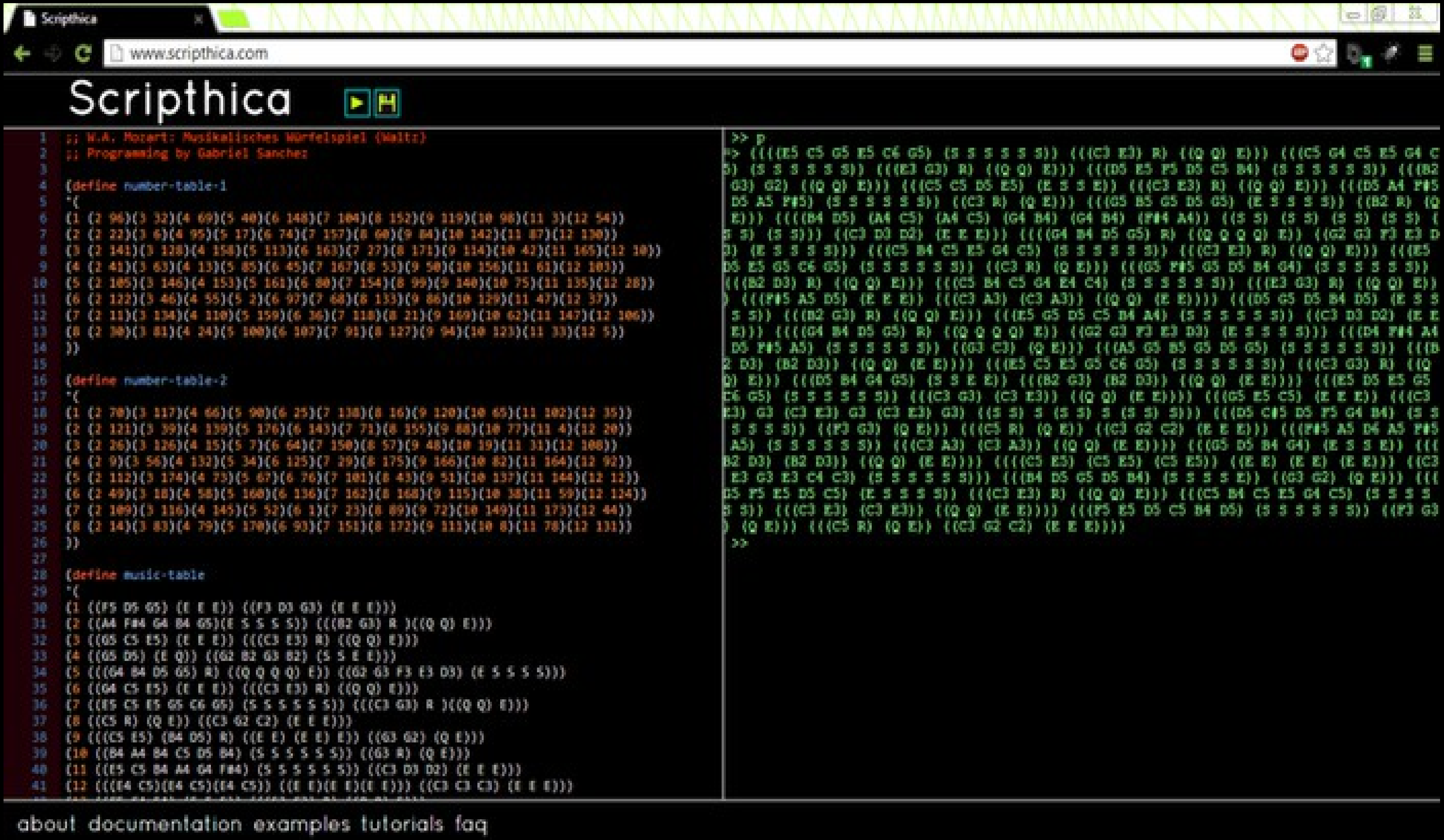
Scripthica

Gabriel
Sánchez

What is it?

- A web based environment for learning, listening, sharing and creating algorithmic computer music using Scheme and JavaScript.
- MIDI, Notation, Live Coding.
- Pedagogical tool.
- Collective Composition.
- FOSS.



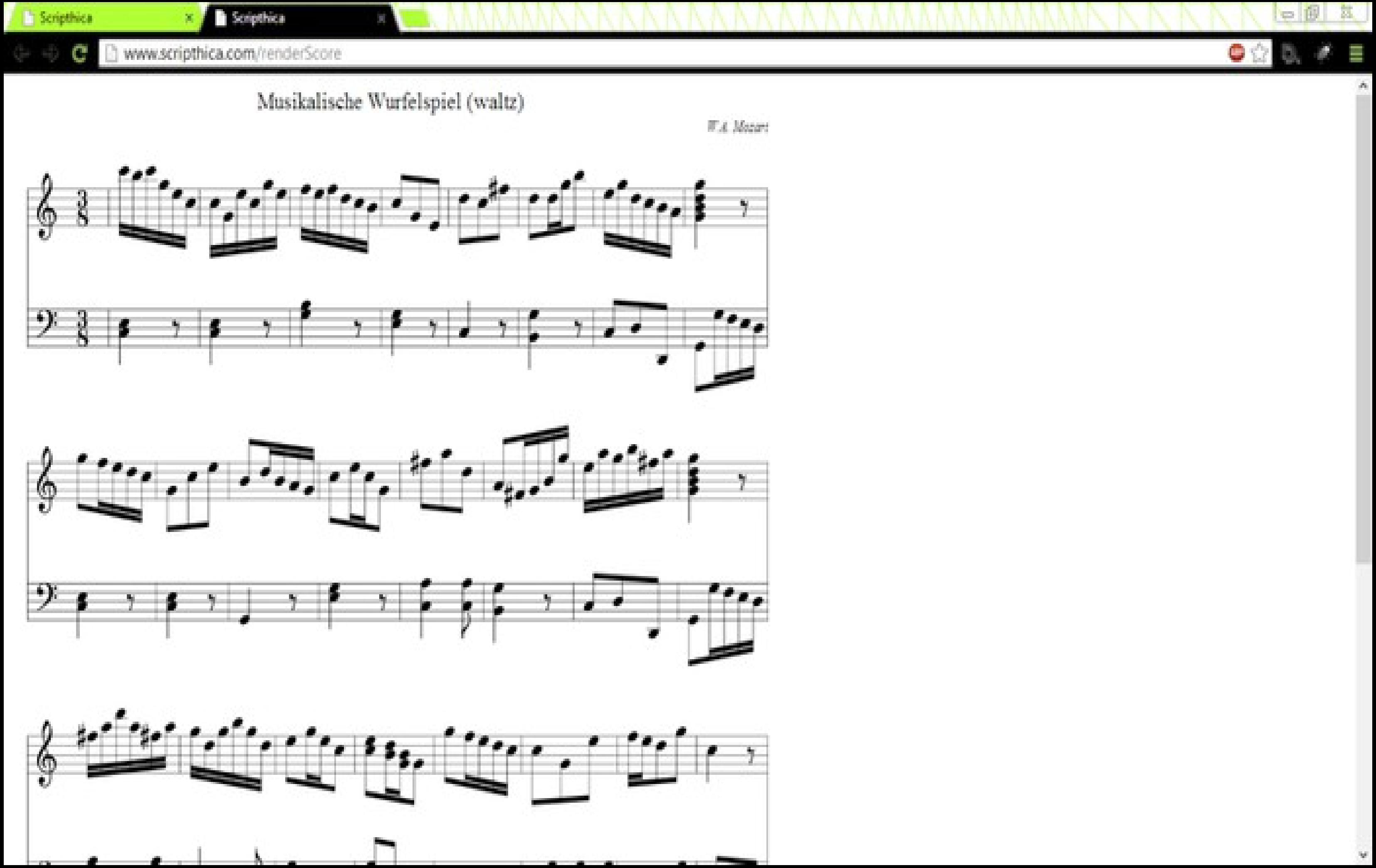
```
;; W.A. Mozart: Musikalisches Würfelspiel (Waltz)
;; Programming by Gabriel Sanchez

(define number-table-1
'(
(1 (2 96)(3 32)(4 69)(5 40)(6 148)(7 104)(8 152)(9 119)(10 98)(11 3)(12 54))
(2 (2 22)(3 6)(4 95)(5 17)(6 74)(7 157)(8 60)(9 84)(10 142)(11 87)(12 130))
(3 (2 141)(3 128)(4 158)(5 113)(6 163)(7 27)(8 171)(9 114)(10 42)(11 165)(12 10))
(4 (2 41)(3 63)(4 13)(5 85)(6 45)(7 167)(8 53)(9 50)(10 156)(11 61)(12 103))
(5 (2 105)(3 146)(4 153)(5 161)(6 80)(7 154)(8 99)(9 140)(10 75)(11 135)(12 28))
(6 (2 122)(3 46)(4 55)(5 2)(6 97)(7 68)(8 133)(9 86)(10 129)(11 47)(12 37))
(7 (2 11)(3 134)(4 110)(5 159)(6 36)(7 118)(8 21)(9 169)(10 62)(11 147)(12 106))
(8 (2 30)(3 81)(4 24)(5 100)(6 107)(7 91)(8 127)(9 94)(10 123)(11 33)(12 5))
))

(define number-table-2
'(
(1 (2 70)(3 117)(4 66)(5 90)(6 25)(7 138)(8 16)(9 120)(10 65)(11 102)(12 35))
(2 (2 121)(3 39)(4 139)(5 176)(6 143)(7 71)(8 155)(9 88)(10 77)(11 4)(12 20))
(3 (2 26)(3 126)(4 15)(5 7)(6 64)(7 150)(8 57)(9 48)(10 19)(11 31)(12 108))
(4 (2 9)(3 56)(4 132)(5 34)(6 125)(7 29)(8 175)(9 166)(10 82)(11 164)(12 92))
(5 (2 112)(3 174)(4 73)(5 67)(6 76)(7 101)(8 43)(9 51)(10 137)(11 144)(12 12))
(6 (2 49)(3 18)(4 58)(5 160)(6 136)(7 162)(8 168)(9 115)(10 38)(11 59)(12 124))
(7 (2 109)(3 116)(4 145)(5 52)(6 1)(7 23)(8 89)(9 72)(10 149)(11 173)(12 44))
(8 (2 14)(3 83)(4 79)(5 170)(6 93)(7 151)(8 172)(9 111)(10 8)(11 78)(12 131))
))

(define music-table
'(
(1 ((F5 D5 G5) (E E E)) ((F3 D3 G3) (E E E)))
(2 ((A4 F#4 G4 B4 G5)(E S S S S)) (((B2 G3) R )((Q Q) E)))
(3 ((G5 C5 E5) (E E E)) (((C3 E3) R) ((Q Q) E)))
(4 ((G5 D5) (E Q)) ((G2 B2 G3 B2) (S S E E)))
(5 (((G4 B4 D5 G5) R) ((Q Q Q Q) E)) ((G2 G3 F3 E3 D3) (E S S S S)))
(6 ((G4 C5 E5) (E E E)) (((C3 E3) R) ((Q Q) E)))
(7 ((E5 C5 E5 G5 C6 G5) (S S S S S S)) (((C3 G3) R )((Q Q) E)))
(8 ((C5 R) (Q E)) ((C3 G2 C2) (E E E)))
(9 (((C5 E5) (B4 D5) R) ((E E) (E E) E)) ((G3 G2) (Q E)))
(10 ((B4 A4 B4 C5 D5 B4) (S S S S S S)) ((G3 R) (Q E)))
(11 ((E5 C5 B4 A4 G4 F#4) (S S S S S S)) ((C3 D3 D2) (E E E)))
(12 (((E4 C5)(E4 C5)(E4 C5)) ((E E)(E E)(E E))) ((C3 C3 C3) (E E E)))
```

```
>> p
=> ((((E5 C5 G5 E5 C6 G5) (S S S S S S)) (((C3 E3) R) ((Q Q) E))) (((C5 G4 C5 E5 G4 C
5) (S S S S S S)) (((E3 G3) R) ((Q Q) E))) (((D5 E5 F5 D5 C5 B4) (S S S S S S)) (((B2
 G3) G2) ((Q Q) E))) (((C5 C5 D5 E5) (E S S E)) (((C3 E3) R) ((Q Q) E))) (((D5 A4 F#5
 D5 A5 F#5) (S S S S S S)) ((C3 R) (Q E))) (((G5 B5 G5 D5 G5) (E S S S S)) ((B2 R) (Q
 E))) ((((B4 D5) (A4 C5) (A4 C5) (G4 B4) (G4 B4) (F#4 A4)) ((S S) (S S) (S S) (S S) (
S S) (S S))) ((C3 D3 D2) (E E E))) ((((G4 B4 D5 G5) R) ((Q Q Q Q) E)) ((G2 G3 F3 E3 D
3) (E S S S S))) (((C5 B4 C5 E5 G4 C5) (S S S S S S)) (((C3 E3) R) ((Q Q) E))) (((E5
D5 E5 G5 C6 G5) (S S S S S S)) ((C3 R) (Q E))) (((G5 F#5 G5 D5 B4 G4) (S S S S S S))
 (((B2 D3) R) ((Q Q) E))) (((C5 B4 C5 G4 E4 C4) (S S S S S S)) (((E3 G3) R) ((Q Q) E))
) (((F#5 A5 D5) (E E E)) (((C3 A3) (C3 A3)) ((Q Q) (E E)))) (((D5 G5 D5 B4 D5) (E S S
 S S)) (((B2 G3) R) ((Q Q) E))) (((E5 G5 D5 C5 B4 A4) (S S S S S S)) ((C3 D3 D2) (E E
 E))) ((((G4 B4 D5 G5) R) ((Q Q Q Q) E)) ((G2 G3 F3 E3 D3) (E S S S S))) (((D4 F#4 A4
 D5 F#5 A5) (S S S S S S)) ((G3 C3) (Q E))) (((A5 G5 B5 G5 D5 G5) (S S S S S S)) (((B
2 D3) (B2 D3)) ((Q Q) (E E)))) (((E5 C5 E5 G5 C6 G5) (S S S S S S)) (((C3 G3) R) ((Q
Q) E))) (((D5 B4 G4 G5) (S S E E)) (((B2 G3) (B2 D3)) ((Q Q) (E E)))) (((E5 D5 E5 G5
C6 G5) (S S S S S S)) (((C3 G3) (C3 E3)) ((Q Q) (E E)))) (((G5 E5 C5) (E E E)) (((C3
E3) G3 (C3 E3) G3 (C3 E3) G3) ((S S) S (S S) S (S S) S))) (((D5 C#5 D5 F5 G4 B4) (S S
 S S S S)) ((F3 G3) (Q E))) (((C5 R) (Q E)) ((C3 G2 C2) (E E E))) (((F#5 A5 D6 A5 F#5
 A5) (S S S S S S)) (((C3 A3) (C3 A3)) ((Q Q) (E E)))) (((G5 D5 B4 G4) (E S S E)) (((
B2 D3) (B2 D3)) ((Q Q) (E E)))) ((((C5 E5) (C5 E5) (C5 E5)) ((E E) (E E) (E E))) ((C3
 E3 G3 E3 C4 C3) (S S S S S S))) (((B4 D5 G5 D5 B4) (S S S S E)) ((G3 G2) (Q E))) (((
D5 F5 E5 D5 C5) (E S S S S)) (((C3 E3) R) ((Q Q) E))) (((C5 B4 C5 E5 G4 C5) (S S S S
S S)) (((C3 E3) (C3 E3)) ((Q Q) (E E)))) (((F5 E5 D5 C5 B4 D5) (S S S S S S)) ((F3 G3
) (Q E))) (((C5 R) (Q E)) ((C3 G2 C2) (E E E))))
>>
```



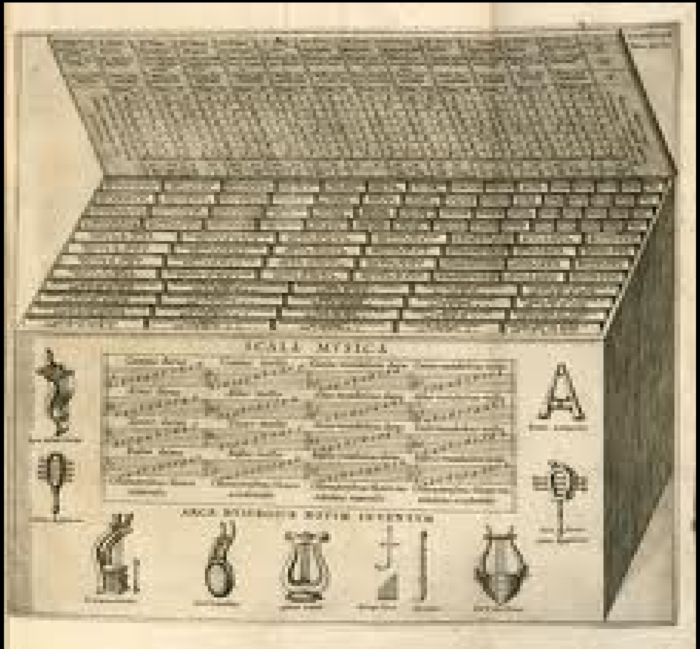
Musikalische Wurfelspiel (waltz)

W.A. Mozart

Arca Musarithmica

“Mechanical music-making is nothing other than a certain closely defined method I have invented, whereby anyone, even if he has no musical knowledge, may, by the varied application of music-making instruments, compose music according to a desired style.”

- Kircher



Why this tool?

- Listening to an algorithm's compositions in a convenient way.
- Composing in any computer with an internet connection.
- Sharing music with others and modifying others' code.
- Learning algorithmic composition.

Musical Operations

```
;; Creating chords:
(create-chord "C4" "min")   ;; => (C4 Eb4 G4)
(create-chord "C4" "maj")   ;; => (C4 E4 G4)
(create-chord "C4" "maj7") ;; => (C4 E4 G4 B4)
(create-chord "C4" "dom")   ;; => (C4 E4 G4 Bb4)
(create-chord 60 "maj")   ;; => (60 64 67)

;; A chord can be represented as a list inside
;; a list. e.g. (C3 (C3 E3 G3) A5)

;; We can transform the pitches to integers
;; for creating MIDI events:
(pitches->numbers '(c5 (c4 d2) e2))
;; => (72 (60 38) 40)
```

Musical Operations

```
;; shuffle
(shuffle '(1 2 3)) ;; => (3 2 1)

;; generate a sequence of integers
(generate-series 0 10 2) ;; => (0 2 4 6 8)

;; delta of a numerical sequence
(delta '(2 4 8 9)) ;; => (2 4 1)

;; retrograde / reverse
(retrograde '(c4 r e2 c2)) ;; => (c2 e2 r c4)

;; palindrome
(palindrome '(c4 r e2 c2)) ;; => (c4 r e2 c2 c2 e2 r
```

Musical Operations

```
;; constant value

(constant-value 3 96) ;; => (96 96 96)


;; inversions

(rotate '(c4 e4 g4) 1) ;; => (g4 c4 e4)


;; transpose

(transpose '(1 2 3) 10) ;; => (11 12 13)


; repeat a pattern

(repeat '(c4 d3 e2) 2) ;; => (c4 d3 e2 c4 d3 e2)
```

Rhythm

Name	Integer Value	Symbol	Value
Longa	16000	L	4
Double Whole	8000	DW	2
Whole	4000	W	1
Half	2000	H	1/2
Quarter	1000	Q	1/4
Eight	500	E	1/8
Sixteenth	250	S	1/16
Thirty-second	125	T	1/32
Sixty-fourth	63	SF	1/64
Hundred twenty-eigth	31	H	1/128

Sonification

```
;; transform data to an audio representation
(set-tempo '((0 0 220)))
(define data '(200 300 44 122 155 231 278))

(define pitches (normalize data 70 100))
;; pitches => (88 100 70 79 83 92 97)

(define lengths (normalize data 100 1000))
;; lengths => (648 1000 100 374 490 757 923)

(define vels (normalize data 40 127))
;; vels => (93 127 40 67 78 104 120)

(save-midi (create-events 0 pitches lengths vels 1))
```

Markov Chains

```
(markov '(c4 d2 e3 c4 a2 d2 g4 a2 a3) 4)
;; => (c4 d2 e3 c4 a2 d2 g4 a2 d2 g4 a2 a3)

(markov '(c4 d2 e3 c4 a2 d2 g4 a2 a3) 4)
;; => (c4 d2 e3 c4 a2 d2 e3 c4 a2 d2 g4 a2 d2 g4 a2 d2
e3 c4 a2 a3)
```

Sieves

```
;create sieves
(define s1 (sieve 2 1 0 20))
;; s1 => (1 3 5 7 9 11 13 15 17 19)
(define s2 (sieve 3 1 0 20))
;; s2 => (1 4 7 10 13 16 19)

;sieve union
(define su (sieve-union s1 s2))
;; su => (1 3 4 5 7 9 10 11 13 15 16 17 19)

;intervallic succession
(define is (delta su))
```

Artificial Neural Networks

- Brain.js

```
;Neural Networks XOR
example

(define neural-net
(nn))

(nn-train neural-net
  '((0 0)(0 1)(1 0)(1
1))
  '((0)(1)(1)(0)))

(nn-predict neural-
net
                 '((0
0)))
(nn-predict neural-
net
```

Scripthica as an educational tool

- 4 day workshop at the National Center for the Arts (Mexico City)
- Results: Students from backgrounds in the Arts, Music, Math, Computer Science, etc. created musical compositions and shared them.

Some of the pieces...

- Unidimensional Cellular Automata by Eduardo Espinoza

  http://www.scripthica.com/6291262726471680

- Fibonacci-Markov by Emiliano Duque

  http://www.scripthica.com/5473896090304512

- N-factorial jazzero by Alex Hernández

  http://www.scripthica.com/6216581835128832

- A piece I wrote while riding the bus by Elihu Garrett

  http://www.scripthica.com/4919909733629952

- Other compositions and code:

  https://www.facebook.com/groups/554811184631785/

JavaScript (OOP) vs Scheme (FP)

Round 1: Radiohead – No Surprises

JavaScript

```
var s = createScore();
var piano =
createTrack("",1,0,86);
var lead = createTrack("",2,0,86);
var bass = createTrack("",3,0,86);

var ppp = ["A5 C5 F5 C5 A5 C5 F5
C5", "A5 C5 F5 C5 Bb4 Db5 F5 G5",
"A5 Bb4 F5 Bb4 A5 Bb4 F5 Bb4",
"Bb5 D5 F5 D5 Bb5 D5 F5 D5", "F5
E5 C5 G4 G4 A4 C5 Bb4", "D5 C5 C5
Bb4 Bb4 A4 C5 Bb4", "G4 G4 G4 A4
Bb4 C5 Bb4 F4", "G4 G4 G4 G4 F5 G4
E5 G4", "E4 G4 E5 G4 F5 E5 C5 G4
C5", "Bb4 Bb4 Db5 F5 C6 Bb5 F5
Db5", "G4 G4 Bb4 G4 F5 D5 Bb4 C5",
"Db5 F4 Eb5 F4 F5 F4 G5 F5",
"A5"];

var ppl =["E E E E E E E E", "E E
E E S S Q E", "Q E S S E E E E",
"E E E E S S E E E","W"];
```

Scheme

```
(set-tempo '((0 0 86)))

(define p-pitches
'(
     (0 A5 C5 F5 C5 A5 C5 F5
C5)
     (1 A5 C5 F5 C5 Bb4 Db5 F5
G5)
     (2 A5 Bb4 F5 Bb4 A5 Bb4
F5 Bb4)
     (3 Bb5 D5 F5 D5 Bb5 D5 F5
D5)
     (4 F5 E5 C5 G4 G4 A4 C5
Bb4)
     (5 D5 C5 C5 Bb4 Bb4 A4 C5
Bb4)
     (6 G4 G4 G4 A4 Bb4 C5 Bb4
F4)
     (7 G4 G4 G4 G4 F5 G4 E5
G4)
     (8 E4 G4 E5 G4 F5 E5 C5
G4 C5)
     (9 Bb4 Bb4 Db5 F5 C6 Bb5
```

JavaScript

```
var bpp = ["F2 F2 F2","F2 Bb2 Bb2
A2","Bb2 Bb2 Bb2","G2 G2 G2","C2
C3 C3","F3"];

var bpl = ["H E Q.","H E. E.
E","W"];

var lpp = ["R","A3 C3 A3 A3 G3 F3
A3 Bb2 R F3 A3 A3 G3 F3 Bb3 D3 E3
E3 F3 G3 A3 R","A3 C3 A3 A3 G3 F3
A3 Bb2 Bb3 A3 A3 A3 G3 F3","Bb3 A3
G3 F3 E3 E3 F3 G3","A3 F3 A3
F3","G3 C4 G3 C3 G3 G3 G3 F3 R Eb3
F3 G3 Ab3 G3 F3 D3 G3 C4 G3 A2 G3
E4 C4 R Bb3 F3 G3 Ab3 G3 F3 R D4
D3 Bb3 A3 Bb3 C4 Db4 Bb3 A3 Db4"];

var lpl = ["W","H H E Q Q. E E+H Q
E. S E Q Q Q E+H H E Q Q Q E+H
H","H H E Q Q. E E+H Q. E E E E Q
Q.","E Q Q. E Q. E Q Q.","W W W
W","E E E E E E E E E E E E E E E
E E E E E Q E E E E E E E E Q Q E
E E E E E Q Q Q Q"];
```

Scheme

```
(define b-pitches
'((0 F2 F2 F2)(1 F2 Bb2 Bb2 A2)(2 Bb2 Bb2
Bb2) (3 G2 G2 G2)(4 C2 C3 C3)(5 F3)))

(define b-lengths
'((0 H E Q.)(1 H E. E. E)(2 W)))

(define l-pitches
'((0 R)(1 A3 C3 A3 A3 G3 F3 A3 Bb2 R F3 A3
A3 G3 F3 Bb3 D3 E3 E3 F3 G3 A3 R)(2 A3 C3
A3 A3 G3 F3 A3 Bb2 Bb3 A3 A3 A3 G3 F3) (3
Bb3 A3 G3 F3 E3 E3 F3 G3)(4 A3 F3 A3 F3)(5
G3 C4 G3 C3 G3 G3 G3 F3 R Eb3 F3 G3 Ab3 G3
F3 D3 G3 C4 G3 A2 G3 E4 C4 R Bb3 F3 G3 Ab3
G3 F3 R D4 D3 Bb3 A3 Bb3 C4 Db4 Bb3 A3 Db4)
))

(define l-lengths
'((0 W)(1 H H E Q Q. E E+H Q E. S E Q Q Q
E+H H E Q Q Q E+H H)(2 H H E Q Q. E E+H Q.
E E E E Q Q.)(3 E Q Q. E Q. E Q Q.)(4 W W W
W)(5 E E E E E E E E E E E E E E E E E E E
E Q E E E E E E E E Q Q E E E E E E Q Q Q
Q)
))
```

JavaScript

```
var piano_intro_p =
mergePatterns(ppp[0], ppp[1]);
var piano_intro_l =
mergePatterns(ppl[0], ppl[0]);

var bass_intro_p =
mergePatterns(bpp[0], bpp[1]);
var bass_intro_l =
mergePatterns(bpl[0], bpl[1]);

var lead_verse_p = lpp[1];
var lead_verse_l = lpl[1];

var piano_verse_p =
mergePatterns(ppp[0], ppp[0],
ppp[2], ppp[2], ppp[3], ppp[4]);
var piano_verse_l =
mergePatterns(ppl[0], ppl[0],
ppl[0], ppl[0], ppl[0], ppl[1]);

var bass_verse_p =
mergePatterns(bpp[0], bpp[0],
bpp[2], bpp[2], bpp[3], bpp[4]);
var bass_verse_l =
mergePatterns(bpl[0], bpl[0],
bpl[0], bpl[0], bpl[0], bpl[0]);
```

Scheme

```
(define pp-intro '((0
1) (0 0)))
(define pp-verse '((0
0 2 2 3 4)
                  (0
0 0 0 0 1)))
(define pp-verse2
'((0 0 2 2 5 6 5 7 5
6)
                  (0
0 0 0 0 2 0 0 0 2)))
(define pp-chorus
'((8 9 8 9 10 11)

(3 0 3 0 0 0)))
(define pp-end '((12)
(4)))
```

JavaScript

```
//intro piano
var sec1 = createSection();
sec1.addEvents(piano_intro_p,
piano_intro_l);
piano.addSection(sec1,4);

//intro bass
var bass_sec1 = createSection();
bass_sec1.addEvents(bass_intro_p,
bass_intro_l);
bass.addSection(bass_sec1,4);

var lead_sec1 = createSection();
lead_sec1.addEvents(lpp[0],
lpl[0]);
lead.addSection(lead_sec1,8);

//verse
var sec2 = createSection();
sec2.addEvents(piano_verse_p,
piano_verse_l);
piano.addSection(sec2,1);
piano.addSection(sec1,1);
piano.addSection(sec2,1);
piano.addSection(sec1,1);

var bass_sec2 = createSection();
```

Scheme

```
(define (l-create-
form f)
(append (repeat (f l-
silence) 8)(f l-
phrase1)(f l-silence)
(f l-phrase1)(f l-
silence)(f l-phrase2)
(repeat (f l-phrase3)
3)(f l-phrase4)(f l-
phrase2)(repeat (f l-
phrase3) 3)(repeat (f
l-silence) 2)(f l-
phrase5)(f l-phrase2)
(repeat (f l-phrase3)
3)))

(define (p-create-
```

JavaScript

```
var sec3 = createSection();
sec3.addEvents(piano_verse2_p,
piano_verse2_l);
piano.addSection(sec3,1);
piano.addSection(sec1,2);
piano.addSection(sec3,1);
piano.addSection(sec1,1);

var bass_sec3 = createSection();
bass_sec3.addEvents(bass_verse2_p,
bass_verse2_l);
bass.addSection(bass_sec3,1);
bass.addSection(bass_sec1,2);
bass.addSection(bass_sec3,1);
bass.addSection(bass_sec1,1);

var lead_sec3 = createSection();
lead_sec3.addEvents(lpp[2],
lpl[2]);
lead.addSection(lead_sec3,1);

var lead_sec4 = createSection();
lead_sec4.addEvents(lpp[3],
lpl[3]);
lead.addSection(lead_sec4,3);

var lead_sec5 = createSection();
```

Scheme

```
(define pp-form (p-
create-form (lambda
(x) (car x))))
(define pl-form (p-
create-form (lambda
(x) (cadr x))))

(define bp-form (b-
create-form (lambda
(x) (car x))))
(define bl-form (b-
create-form (lambda
(x) (cadr x))))

(define lp-form (l-
create-form (lambda
(x) (car x))))
```

JavaScript

```
//chorus
var sec4 = createSection();
sec4.addEvents(piano_chorus_p, piano_chorus_l);
piano.addSection(sec4,1);
piano.addSection(sec3,1);
piano.addSection(sec1,1);

var bass_sec4 = createSection();
bass_sec4.addEvents(bass_chorus_p,bass_chorus_l);
bass.addSection(bass_sec4,1);
bass.addSection(bass_sec3,1);
bass.addSection(bass_sec1,1);

var sec5 = createSection();
sec5.addEvents(ppp[12], ppl[4]);
piano.addSection(sec5,1);

var bass_sec6 = createSection();
bass_sec6.addEvents(bpp[5], bpl[2]);
bass.addSection(bass_sec6,1);

s.addTrack(piano);
s.addTrack(bass);
s.addTrack(lead);
s.saveMidi();
```

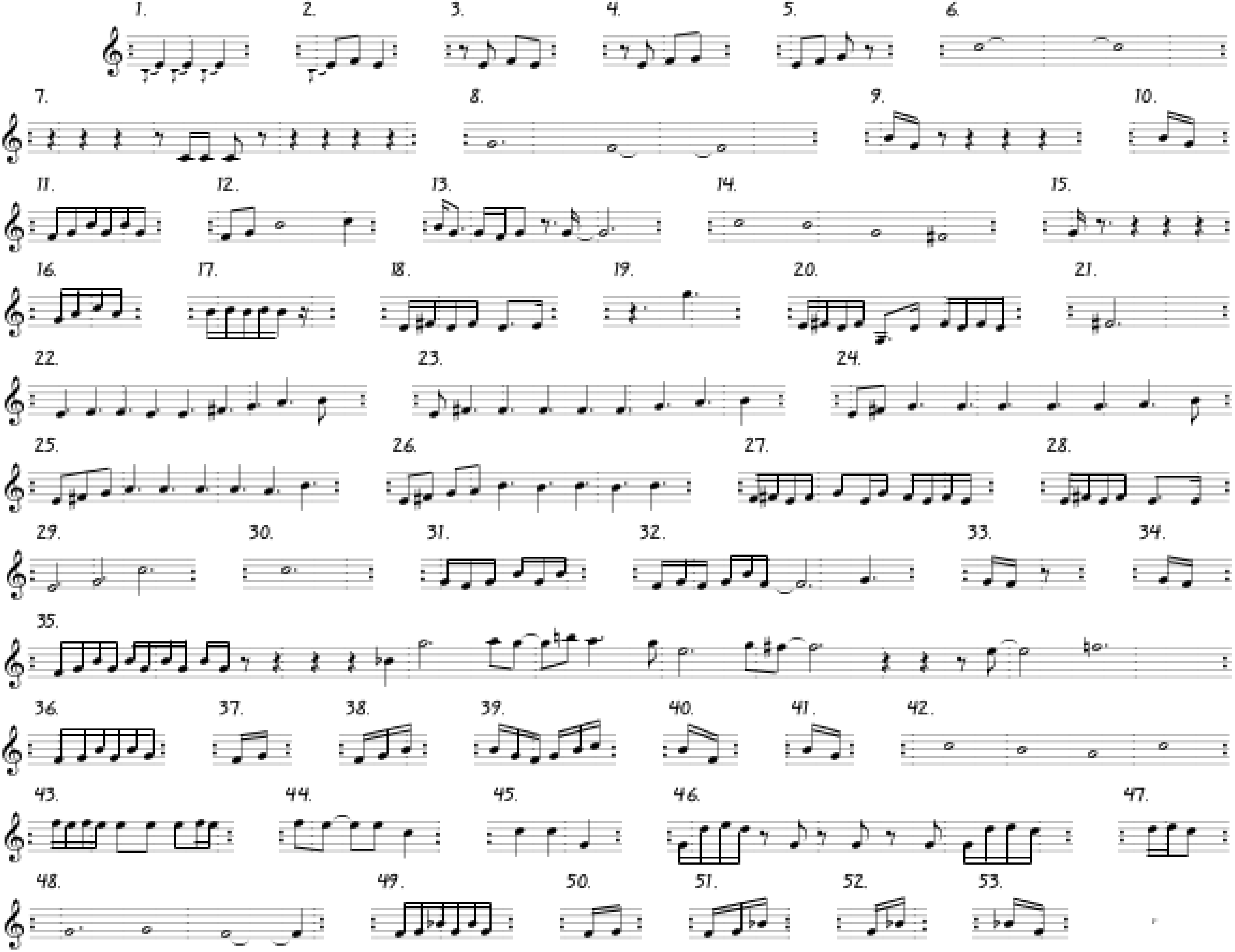
Scheme

```
(define pp (pitches->numbers (merge-patterns pp-form p-pitches)))
(define pl (lengths->numbers (merge-patterns pl-form p-lengths)))
(define pv (create-constant-velocities pp 96))
(define piano (create-events 0 pp pl pv 0))

(define lp (pitches->numbers (merge-
```

JavaScript (OOP) vs Scheme (FP)

Round 2: Terry Riley – IN C


1.
2.
3.
4.
5.
6.
7.
8.
9.
10.
11.
12.
13.
14.
15.
16.
17.
18.
19.
20.
21.
22.
23.
24.
25.
26.
27.
28.
29.
30.
31.
32.
33.
34.
35.
36.
37.
38.
39.
40.
41.
42.
43.
44.
45.
46.
47.
48.
49.
50.
51.
52.
53.

JavaScript

```
var s = createScore();
var t1 = createTrack("",1,0,160);
var t2 = createTrack("",2,4,160);
var t3 = createTrack("",3,46,160);
var t4 = createTrack("",4,13,160);
var t5 = createTrack("",5,24,160);
var t6 = createTrack("",6,32,160);
```

Scheme

```
(set-tempo '((0 0 160)))
(set-instruments '((0 0 2) (1
0 6) (2 0 46) (3 0 8) (4 0 1)
(5 0 0)))
```

JavaScript

```
var x = [["C4 E4 C4
E4 C4 E4","E E E E E
E"],["C4 E4 F4 E4","E
E E Q"],["R E4 F4
E4","E E E E"],["R E4
F4 G4","E E E E"],
["E4 F4 G4 R","E E E
E"],["C5","DW"],["R R
R R C4 C4 C4 R R R R
R","Q Q Q E S S E E Q
Q Q Q"],["G4 F4","W.
DW"],["B4 G4 R R R
R","S S E Q Q Q"],
["B4 G4","S S"],["F4
G4 B4 G4 B4 G4","S S
S S S S"],["F4 G4 B4
C5","E E W Q"],["B4
```

Scheme

```
(define cells
'((0 (C4 E4 C4
E4 C4 E4) (E E E
E E E))(1 (C4 E4
F4 E4)(E E E Q))
(2 (R E4 F4 E4)
(E E E E))(3 (R
E4 F4 G4)(E E E
E))(4 (E4 F4 G4
R)(E E E E))(5
(C5)(DW))(6 (R R
R R C4 C4 C4 R R
R R R)(Q Q Q E S
```

JavaScript

```
for(var i = 0; i<x.length; i++){

var section = createSection();
section.addEvents(x[i][0],x[i][1]);

t1.addSection(section,random(3,13));
t2.addSection(clone(section),random(3,13));
t3.addSection(clone(section),random(3,13));
t4.addSection(clone(section),random(3,13));
t5.addSection(clone(section),random(3,13));
t6.addSection(clone(section),random(3,13));

}
```

Scheme

```
(define (iter l f x)
(let loop ((n 0))
 (if (> n l) '()
  (append
   (repeat
    (f (list-ref cells n)) x)
 (loop (+ n 1)))))))

(define (create-voice)
(let ((k (+ (random-integer
4) 5)))(list (iter 52 (lambda
(x) (cadr x)) k)(iter 52
(lambda (x) (caddr x)) k))))
```

JavaScript

```
t1.transpose(-
24);
t2.transpose(12)
;
t3.transpose(24)
;
t5.transpose(-
12);
t6.transpose(-
24);

s.addTrack(t1);
s.addTrack(t2);
```

Scheme

```
(define (make-evs l)
 (let loop ((n 0))
  (let ((v (create-voice)))
    (if (> n l) '()
      (append
       (create-events 0
(transpose (pitches->numbers
(car v))(list-ref '(0 12 -12
24 36) (random-integer 4)))
(lengths->numbers (cadr v))
(create-constant-velocities
(pitches->numbers (car v))
96) n) (loop (+ n 1)))))))

(define evs (make-evs 5))
(save-midi evs)
```

JS vs Scheme

- Programming languages and paradigms are tools that help us sketch our artistic ideas.
- Each work of art is different, hence, we must use different tools (programming languages, techniques, paradigms) that will help us reach our aesthetic goals.

Future Work

- Live Coding.
- Finish tutorials.
- Language support.
- Improve the libraries.

Special thanks to...

- FOSS Community
- Kickstarter Supporters
- Eduardo Espinosa (National University of Mexico)
- Hernani Villaseñor & Alberto Cerro (Taller de Audio – National Center for the Arts)
- ACM SIGPLAN & FARM'14 Organizers
- Girlfriend, friends & family