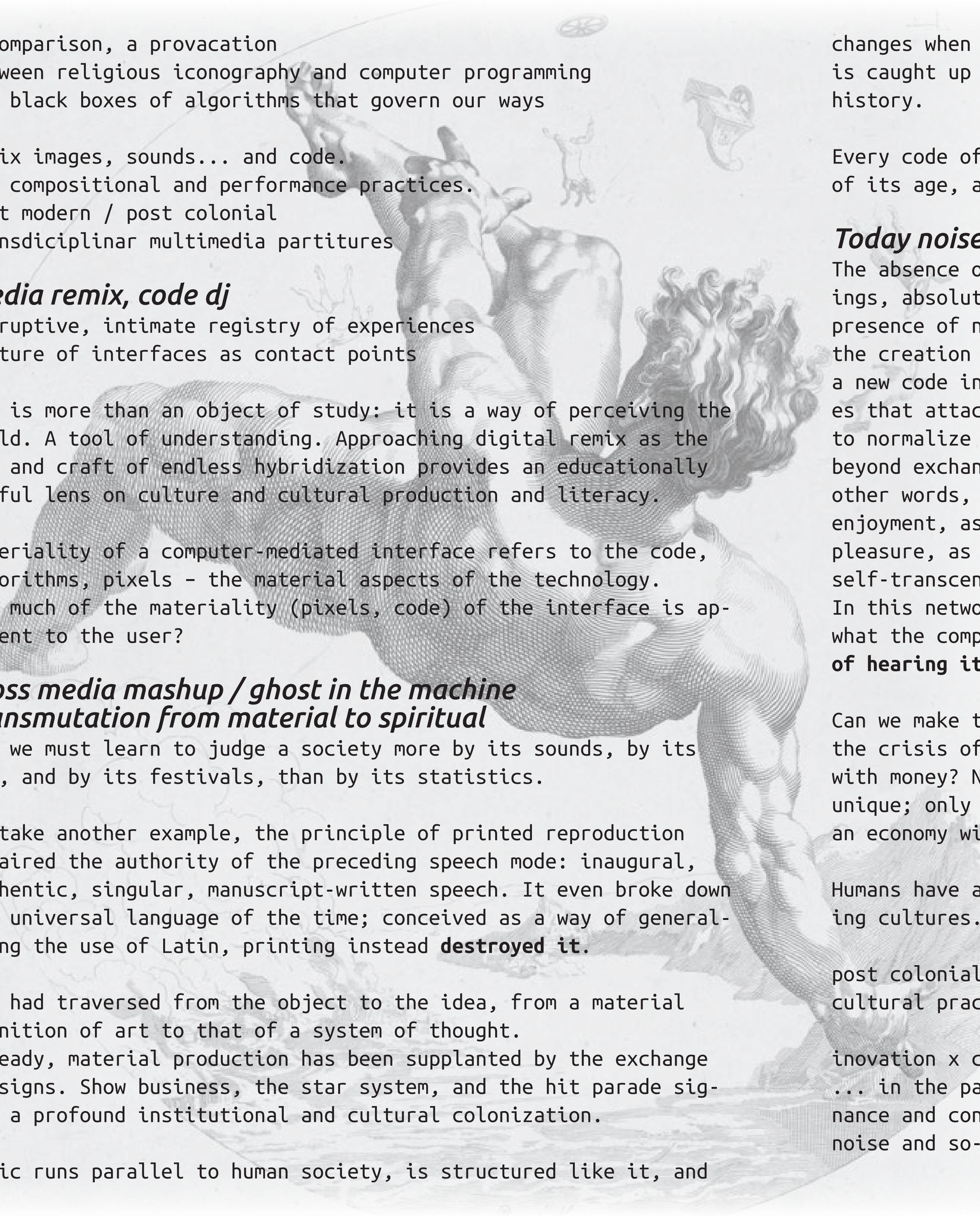
rANGEL

machine
prayer
for
a new
world



a comparison, a provacation
between religious iconography and computer programming
two black boxes of algorithms that govern our ways

Remix images, sounds... and code.
New compositional and performance practices.
post modern / post colonial
transdiciplinar multimedia partitures

media remix, code dj

disruptive, intimate registry of experiences
rupture of interfaces as contact points

Art is more than an object of study: it is a way of perceiving the world. A tool of understanding. Approaching digital remix as the art and craft of endless hybridization provides an educationally useful lens on culture and cultural production and literacy.

Materiality of a computer-mediated interface refers to the code, algorithms, pixels – the material aspects of the technology.
How much of the materiality (pixels, code) of the interface is apparent to the user?

cross media mashup / ghost in the machine transmutation from material to spiritual

Now we must learn to judge a society more by its sounds, by its art, and by its festivals, than by its statistics.

To take another example, the principle of printed reproduction impaired the authority of the preceding speech mode: inaugural, authentic, singular, manuscript-written speech. It even broke down the universal language of the time; conceived as a way of generalizing the use of Latin, printing instead **destroyed it**.

Art had traversed from the object to the idea, from a material definition of art to that of a system of thought.
Already, material production has been supplanted by the exchange of signs. Show business, the star system, and the hit parade signal a profound institutional and cultural colonization.

Music runs parallel to human society, is structured like it, and changes when it does. It does not evolve in a linear fashion, but is caught up in the complexity and circularity of the movements of history.

Every code of music is rooted in the ideologies and technologies of its age, and at the same time produces them.

Today noise reigns supreme over human sensibility.

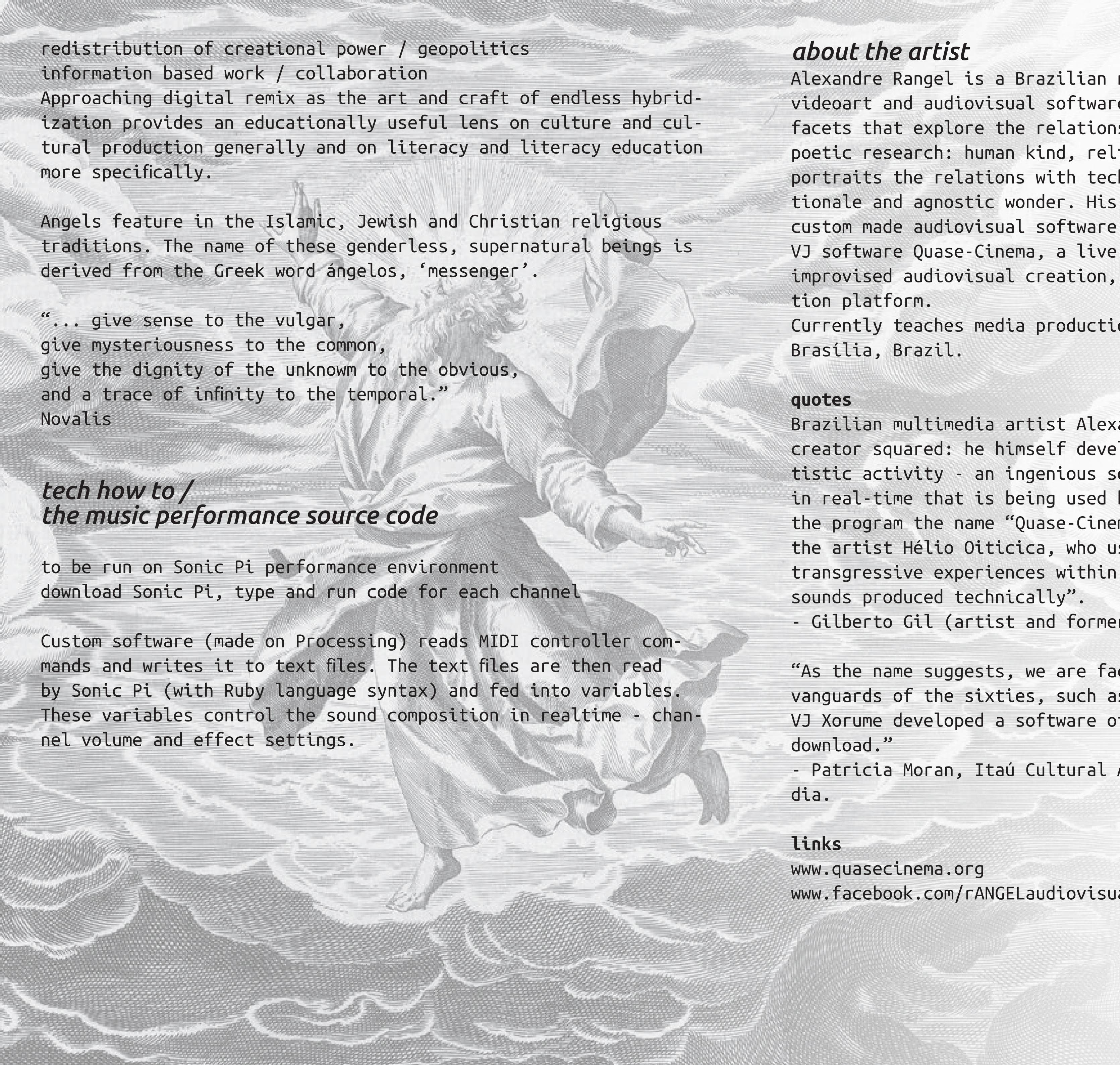
The absence of meaning is in this case the presence of all meanings, absolute ambiguity, a construction outside meaning. The presence of noise makes sense, makes meaning. It makes possible the creation of a new order on another level of organization, of a new code in another network. A network can be destroyed by noises that attack and transform it, if the codes in place are unable to normalize and repress them. we can envision one last network, beyond exchange, in which music could be lived as composition, in other words, in which it would be performed for the musician’s own enjoyment, as self-communication, with no other goal than his own pleasure, as something fundamentally outside all communication, as self-transcendence, a solitary, egotistical, noncommercial act.
In this network, what is heard by others would be a by-product of what the composer or interpreter wrote or **performed for the sake of hearing it**,

Can we make the connections? Can we hear the crisis of society in the crisis of music? Can we understand music through its relations with money? Notwithstanding, the political economy of music is unique; only lately commodified, it soars in the immaterial. It is an economy without quantity.

Humans have always made new culture by taking and remixing existing cultures. -Leissig

post colonial
cultural practices, organization, politics

inovation x concretization
... in the past, the point of disagreement has been between dissonance and consonance, it will be, in the immediate future, between noise and so-called musical sounds. - Cage



redistribution of creational power / geopolitics
information based work / collaboration
Approaching digital remix as the art and craft of endless hybridization provides an educationally useful lens on culture and cultural production generally and on literacy and literacy education more specifically.

Angels feature in the Islamic, Jewish and Christian religious traditions. The name of these genderless, supernatural beings is derived from the Greek word ángelos, ‘messenger’.

“... give sense to the vulgar,
give mysteriousness to the common,
give the dignity of the unknowm to the obvious,
and a trace of infinity to the temporal.”
Novalis

tech how to / the music performance source code

to be run on Sonic Pi performance environment
download Sonic Pi, type and run code for each channel

Custom software (made on Processing) reads MIDI controller commands and writes it to text files. The text files are then read by Sonic Pi (with Ruby language syntax) and fed into variables. These variables control the sound composition in realtime - channel volume and effect settings.

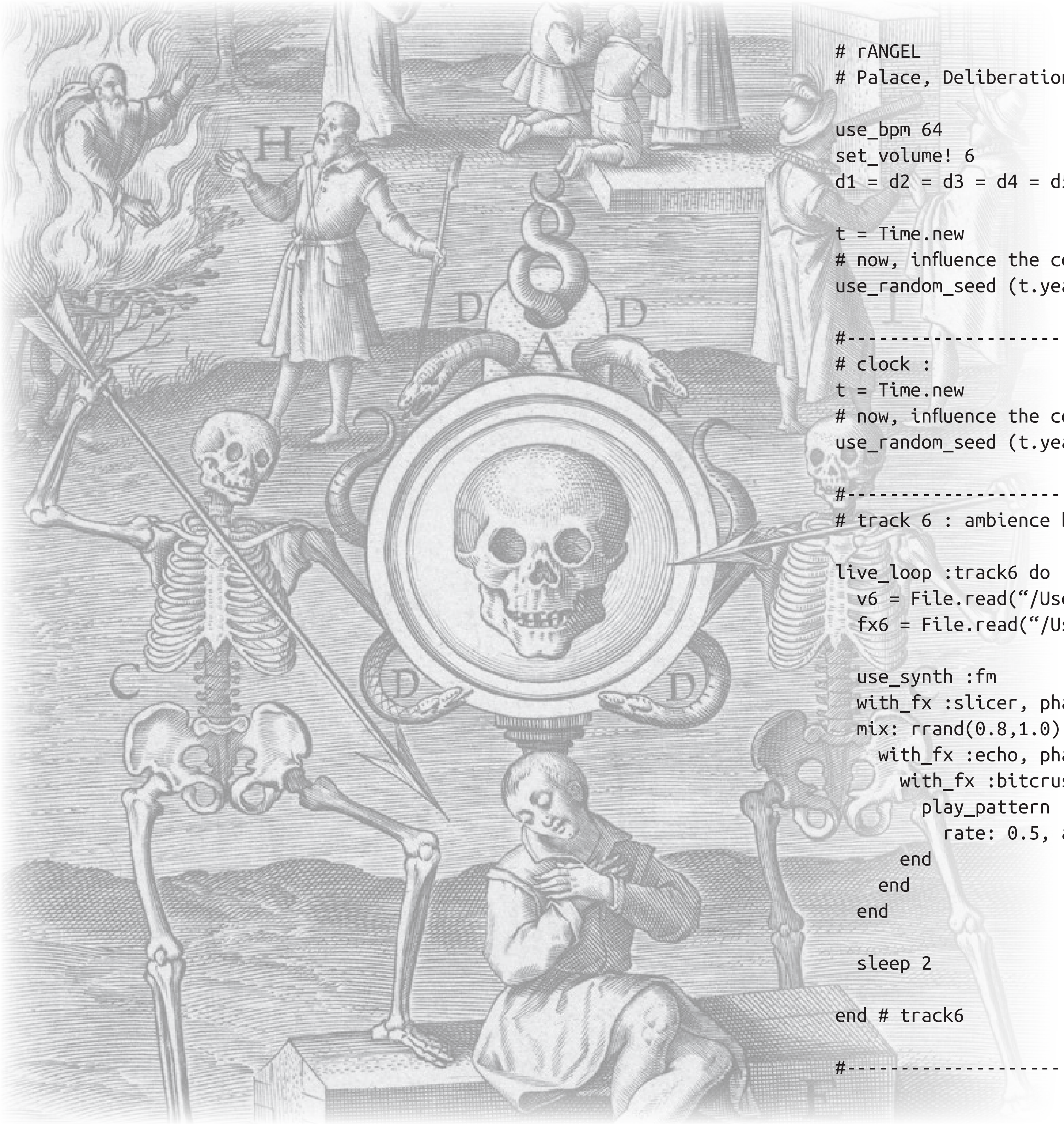
about the artist

Alexandre Rangel is a Brazilian new media artist working with videoart and audiovisual software development. His works have facets that explore the relations between the main themes of his poetic research: human kind, religiosity and urban society. He portraits the relations with technology as motif of higher rationale and agnostic wonder. His main tools are the camera and custom made audiovisual software. Rangel develops the open source VJ software Quase-Cinema, a live video editing environment for improvised audiovisual creation, experimentation and art education platform.
Currently teaches media production at the IESB University in Brasília, Brazil.

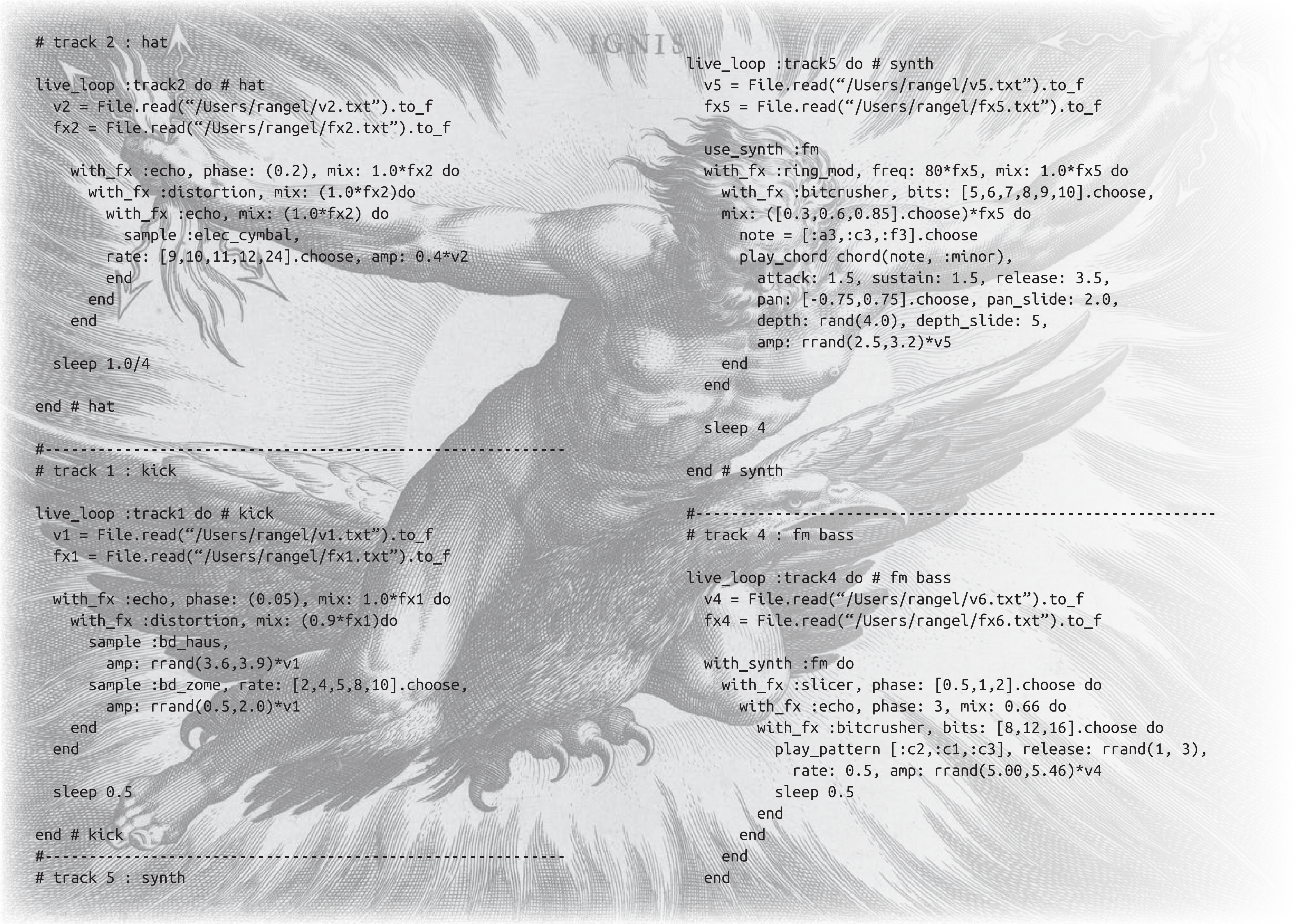
quotes
Brazilian multimedia artist Alexandre Rangel (VJ Xorume) is a creator squared: he himself developed the main tool of his artistic activity - an ingenious software for manipulating video in real-time that is being used by VJs worldwide. Alexandre gave the program the name “Quase-Cinema” (almost cinema), in homage to the artist Hélio Oiticica, who used the term to mean “a field of transgressive experiences within the realm of media or images and sounds produced technically”.
- Gilberto Gil (artist and former Brazilian minister of culture)

“As the name suggests, we are facing an heir of the Brazilian art vanguards of the sixties, such as Hélio Oiticica and Lygia Clark. VJ Xorume developed a software of the same name available for download.”
- Patricia Moran, Itaú Cultural Arts and Technology Encyclopaedia.

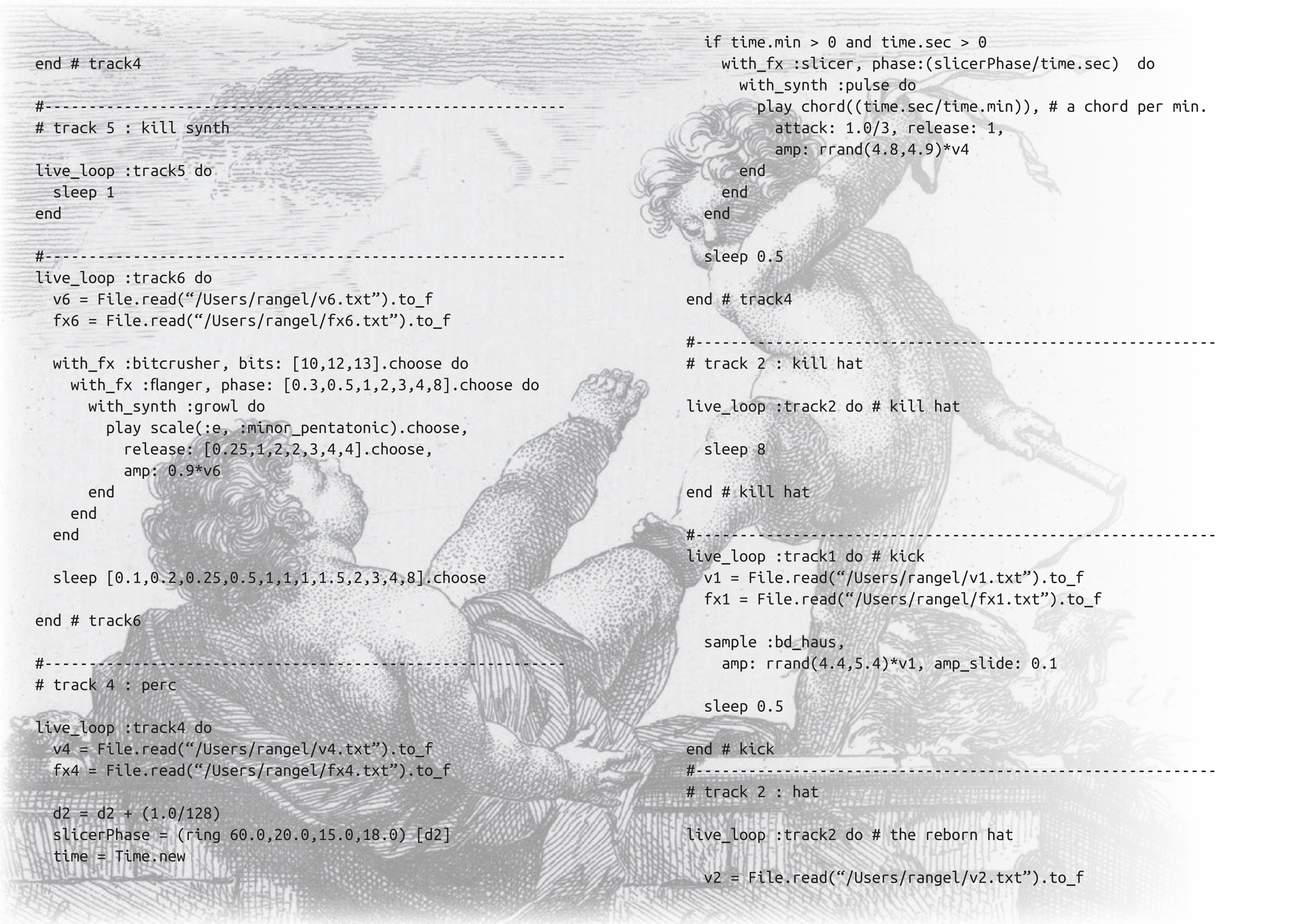
links
www.quasecinema.org
www.facebook.com/rANGELaudiovisual



```
# rANGEL
# Palace, Deliberation, Horn, Manifestation, Wings

use_bpm 64
set_volume! 6
d1 = d2 = d3 = d4 = d5 = d6 = 0

t = Time.new
# now, influence the composition with the current time
use_random_seed (t.year + t.month + t.day + t.hour + t.min)

#------------------------------------------------------------
# clock :
t = Time.new
# now, influence the composition with the current time
use_random_seed (t.year + t.month + t.day + t.hour + t.sec)

#------------------------------------------------------------
# track 6 : ambience bass

live_loop :track6 do
  v6 = File.read(“/Users/rangel/v6.txt”).to_f
  fx6 = File.read(“/Users/rangel/fx6.txt”).to_f

  use_synth :fm
  with_fx :slicer, phase: [0.25,0.5,0.5,1,1].choose,
  mix: rrand(0.8,1.0) do
    with_fx :echo, phase: 3, mix: 0.66 do
      with_fx :bitcrusher, bits: [8,12,16].choose do
        play_pattern [:c2,:c1,:c3], release: rrand(1, 3),
          rate: 0.5, amp: rrand(4.45,4.8)*v6
      end
    end
  end

  sleep 2

end # track6

#------------------------------------------------------------
```

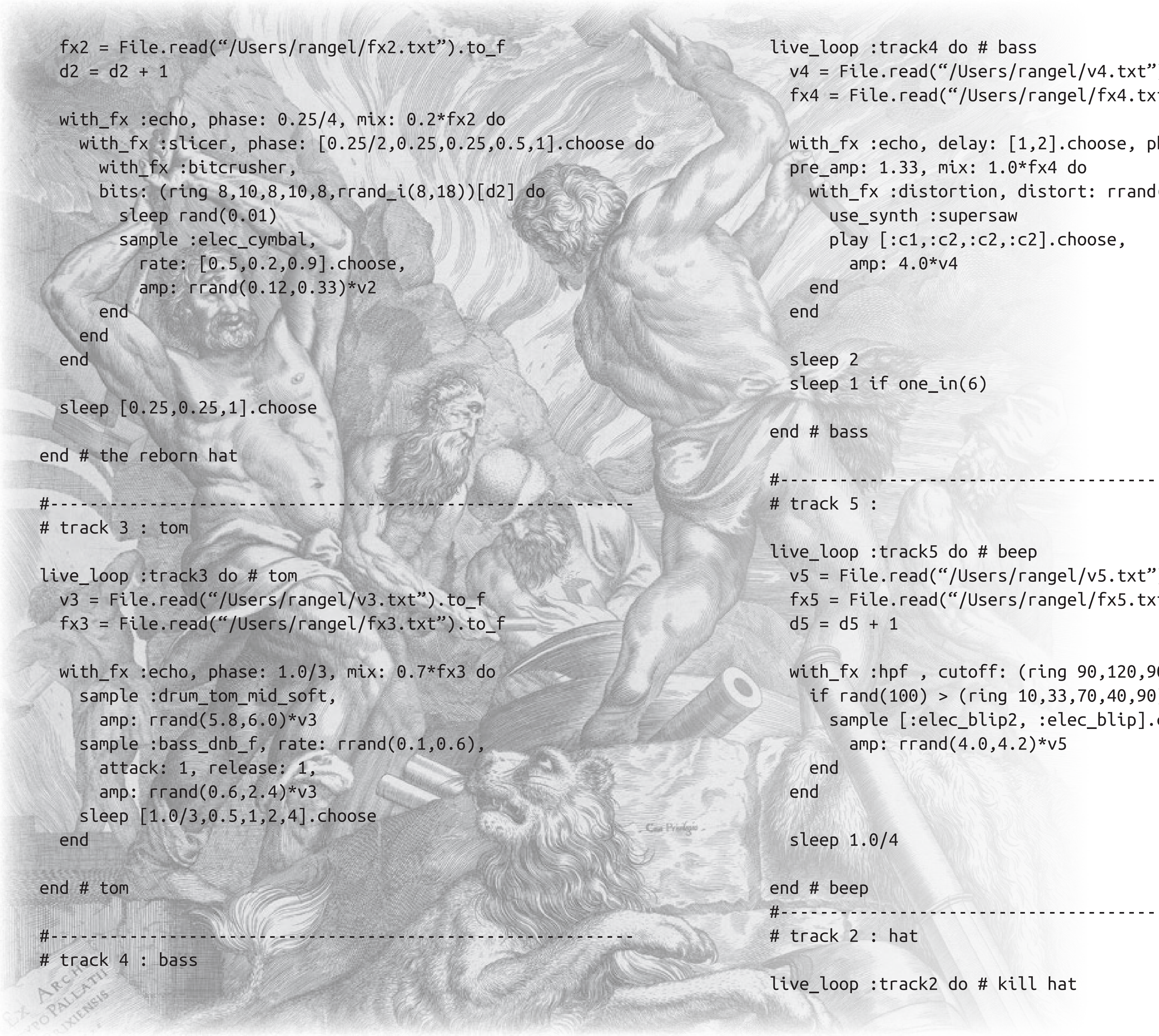


```
# track 2 : hat

live_loop :track2 do # hat
  v2 = File.read(“/Users/rangel/v2.txt”).to_f
  fx2 = File.read(“/Users/rangel/fx2.txt”).to_f

    with_fx :echo, phase: (0.2), mix: 1.0*fx2 do
      with_fx :distortion, mix: (1.0*fx2)do
        with_fx :echo, mix: (1.0*fx2) do
          sample :elec_cymbal,
        rate: [9,10,11,12,24].choose, amp: 0.4*v2
        end
      end
    end

  sleep 1.0/4

end # hat

#------------------------------------------------------------
# track 1 : kick

live_loop :track1 do # kick
  v1 = File.read(“/Users/rangel/v1.txt”).to_f
  fx1 = File.read(“/Users/rangel/fx1.txt”).to_f

  with_fx :echo, phase: (0.05), mix: 1.0*fx1 do
    with_fx :distortion, mix: (0.9*fx1)do
      sample :bd_haus,
        amp: rrand(3.6,3.9)*v1
      sample :bd_zome, rate: [2,4,5,8,10].choose,
        amp: rrand(0.5,2.0)*v1
    end
  end

  sleep 0.5

end # kick
#------------------------------------------------------------
# track 5 : synth

live_loop :track5 do # synth
  v5 = File.read(“/Users/rangel/v5.txt”).to_f
  fx5 = File.read(“/Users/rangel/fx5.txt”).to_f

  use_synth :fm
  with_fx :ring_mod, freq: 80*fx5, mix: 1.0*fx5 do
    with_fx :bitcrusher, bits: [5,6,7,8,9,10].choose,
    mix: ([0.3,0.6,0.85].choose)*fx5 do
      note = [:a3,:c3,:f3].choose
      play_chord chord(note, :minor),
        attack: 1.5, sustain: 1.5, release: 3.5,
        pan: [-0.75,0.75].choose, pan_slide: 2.0,
        depth: rand(4.0), depth_slide: 5,
        amp: rrand(2.5,3.2)*v5
    end
  end

  sleep 4

end # synth

#------------------------------------------------------------
# track 4 : fm bass

live_loop :track4 do # fm bass
  v4 = File.read(“/Users/rangel/v6.txt”).to_f
  fx4 = File.read(“/Users/rangel/fx6.txt”).to_f

  with_synth :fm do
    with_fx :slicer, phase: [0.5,1,2].choose do
      with_fx :echo, phase: 3, mix: 0.66 do
        with_fx :bitcrusher, bits: [8,12,16].choose do
          play_pattern [:c2,:c1,:c3], release: rrand(1, 3),
            rate: 0.5, amp: rrand(5.00,5.46)*v4
          sleep 0.5
        end
      end
    end
  end
```

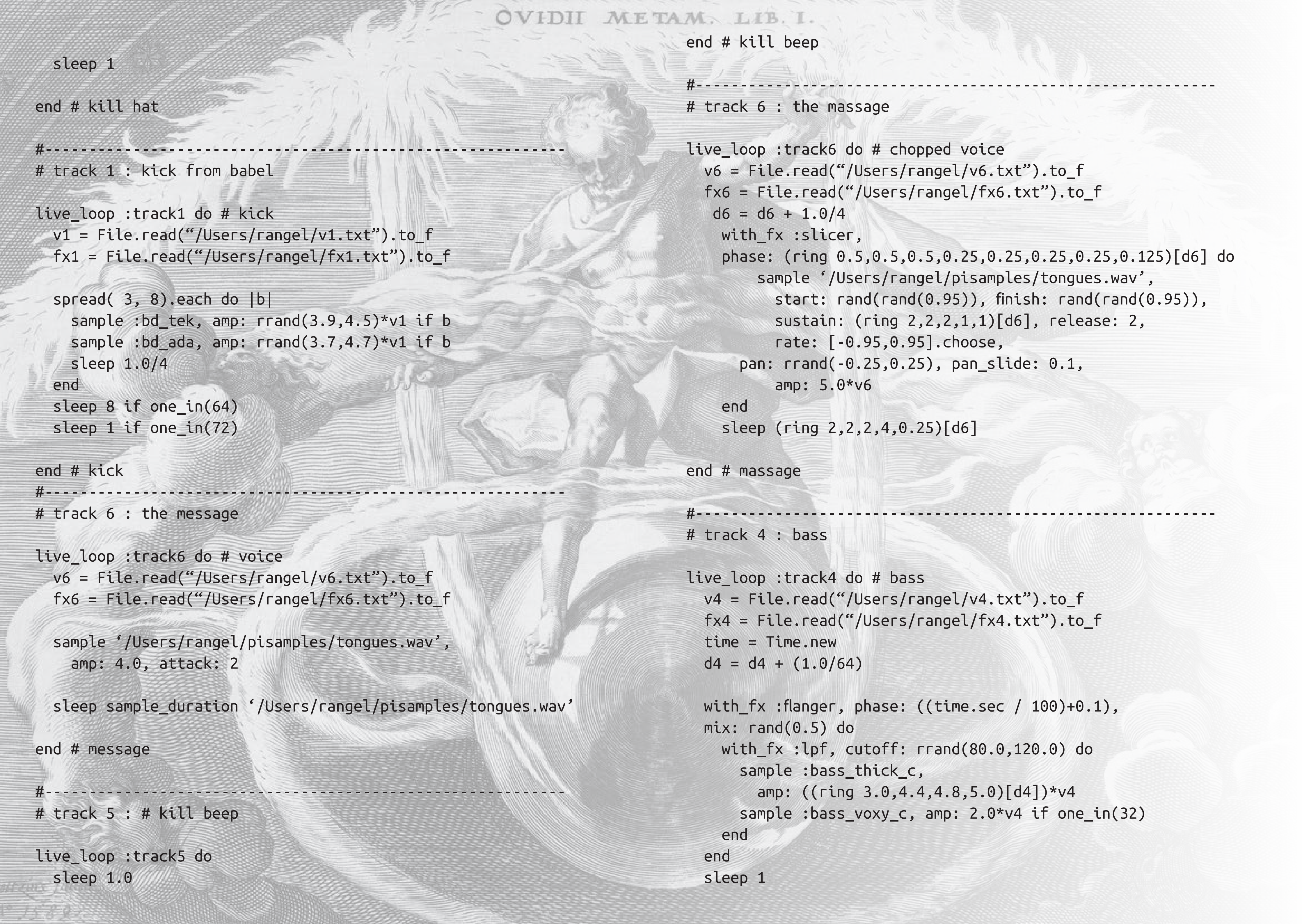


```
end # track4

#-----------------------------------------------------------
# track 5 : kill synth

live_loop :track5 do
  sleep 1
end

#-----------------------------------------------------------
live_loop :track6 do
  v6 = File.read(“/Users/rangel/v6.txt”).to_f
  fx6 = File.read(“/Users/rangel/fx6.txt”).to_f

  with_fx :bitcrusher, bits: [10,12,13].choose do
    with_fx :flanger, phase: [0.3,0.5,1,2,3,4,8].choose do
      with_synth :growl do
        play scale(:e, :minor_pentatonic).choose,
          release: [0.25,1,2,2,3,4,4].choose,
          amp: 0.9*v6
      end
    end
  end

  sleep [0.1,0.2,0.25,0.5,1,1,1,1.5,2,3,4,8].choose

end # track6

#-----------------------------------------------------------
# track 4 : perc

live_loop :track4 do
  v4 = File.read(“/Users/rangel/v4.txt”).to_f
  fx4 = File.read(“/Users/rangel/fx4.txt”).to_f

  d2 = d2 + (1.0/128)
  slicerPhase = (ring 60.0,20.0,15.0,18.0) [d2]
  time = Time.new
```

```
  if time.min > 0 and time.sec > 0
    with_fx :slicer, phase:(slicerPhase/time.sec)  do
      with_synth :pulse do
        play chord((time.sec/time.min)), # a chord per min.
          attack: 1.0/3, release: 1,
          amp: rrand(4.8,4.9)*v4
      end
    end
  end

  sleep 0.5

end # track4

#-----------------------------------------------------------
# track 2 : kill hat

live_loop :track2 do # kill hat

  sleep 8

end # kill hat

#-----------------------------------------------------------
live_loop :track1 do # kick
  v1 = File.read(“/Users/rangel/v1.txt”).to_f
  fx1 = File.read(“/Users/rangel/fx1.txt”).to_f

  sample :bd_haus,
    amp: rrand(4.4,5.4)*v1, amp_slide: 0.1

  sleep 0.5

end # kick
#-----------------------------------------------------------
# track 2 : hat

live_loop :track2 do # the reborn hat

  v2 = File.read(“/Users/rangel/v2.txt”).to_f
```

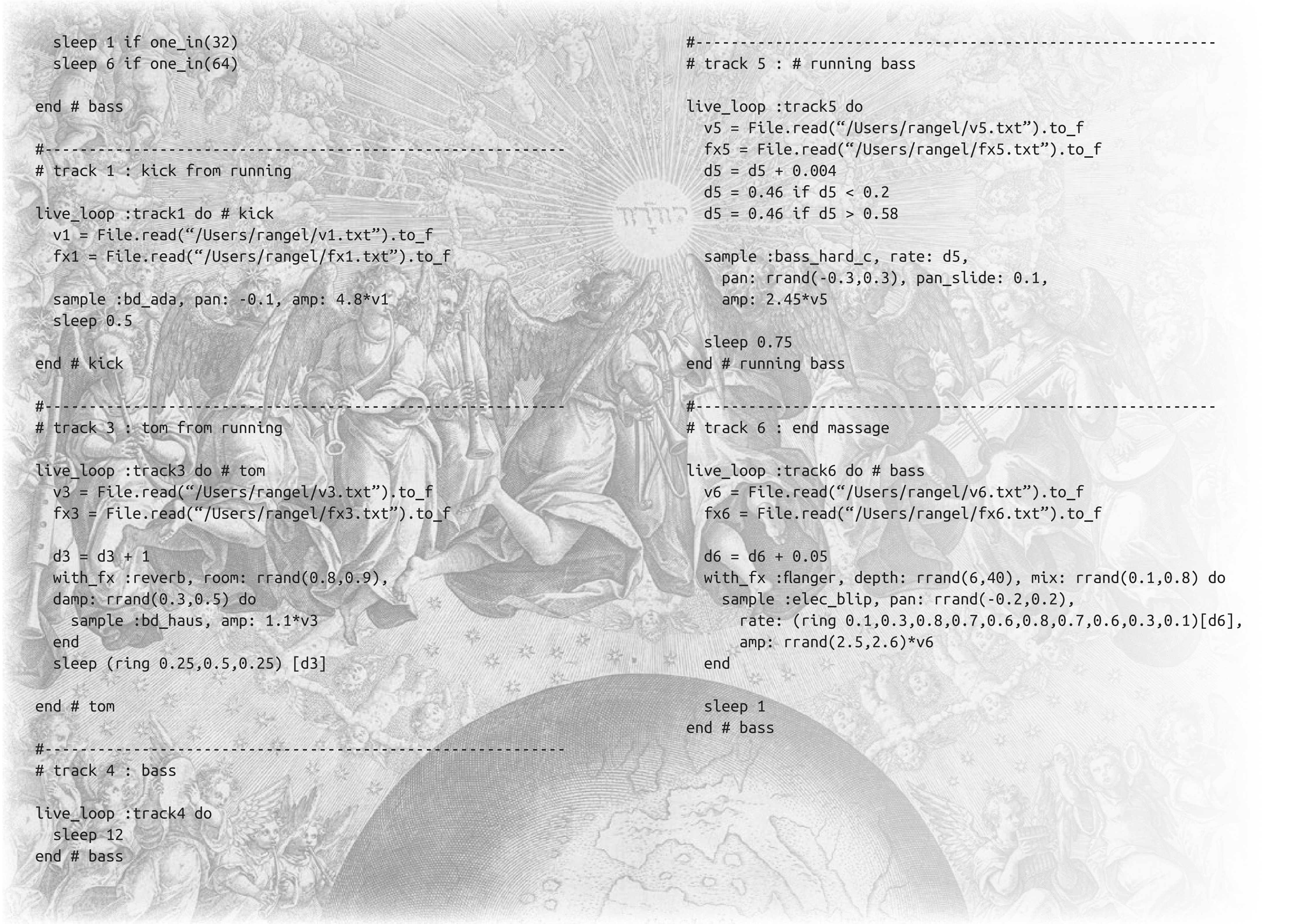


```
  fx2 = File.read(“/Users/rangel/fx2.txt”).to_f
  d2 = d2 + 1

  with_fx :echo, phase: 0.25/4, mix: 0.2*fx2 do
    with_fx :slicer, phase: [0.25/2,0.25,0.25,0.5,1].choose do
      with_fx :bitcrusher,
      bits: (ring 8,10,8,10,8,rrand_i(8,18))[d2] do
        sleep rand(0.01)
        sample :elec_cymbal,
          rate: [0.5,0.2,0.9].choose,
          amp: rrand(0.12,0.33)*v2
      end
    end
  end

  sleep [0.25,0.25,1].choose

end # the reborn hat

#------------------------------------------------------------
# track 3 : tom

live_loop :track3 do # tom
  v3 = File.read(“/Users/rangel/v3.txt”).to_f
  fx3 = File.read(“/Users/rangel/fx3.txt”).to_f

  with_fx :echo, phase: 1.0/3, mix: 0.7*fx3 do
    sample :drum_tom_mid_soft,
      amp: rrand(5.8,6.0)*v3
    sample :bass_dnb_f, rate: rrand(0.1,0.6),
      attack: 1, release: 1,
      amp: rrand(0.6,2.4)*v3
    sleep [1.0/3,0.5,1,2,4].choose
  end

end # tom

#------------------------------------------------------------
# track 4 : bass

live_loop :track4 do # bass
  v4 = File.read(“/Users/rangel/v4.txt”).to_f
  fx4 = File.read(“/Users/rangel/fx4.txt”).to_f

  with_fx :echo, delay: [1,2].choose, phase: rrand(0.6,0.8),
  pre_amp: 1.33, mix: 1.0*fx4 do
    with_fx :distortion, distort: rrand(0.4,0.6) do
      use_synth :supersaw
      play [:c1,:c2,:c2,:c2].choose,
        amp: 4.0*v4
    end
  end

  sleep 2
  sleep 1 if one_in(6)

end # bass

#------------------------------------------------------------
# track 5 :

live_loop :track5 do # beep
  v5 = File.read(“/Users/rangel/v5.txt”).to_f
  fx5 = File.read(“/Users/rangel/fx5.txt”).to_f
  d5 = d5 + 1

  with_fx :hpf , cutoff: (ring 90,120,90,60)[d5] do
    if rand(100) > (ring 10,33,70,40,90)[d5]
      sample [:elec_blip2, :elec_blip].choose,
        amp: rrand(4.0,4.2)*v5
    end
  end

  sleep 1.0/4

end # beep
#------------------------------------------------------------
# track 2 : hat

live_loop :track2 do # kill hat
```



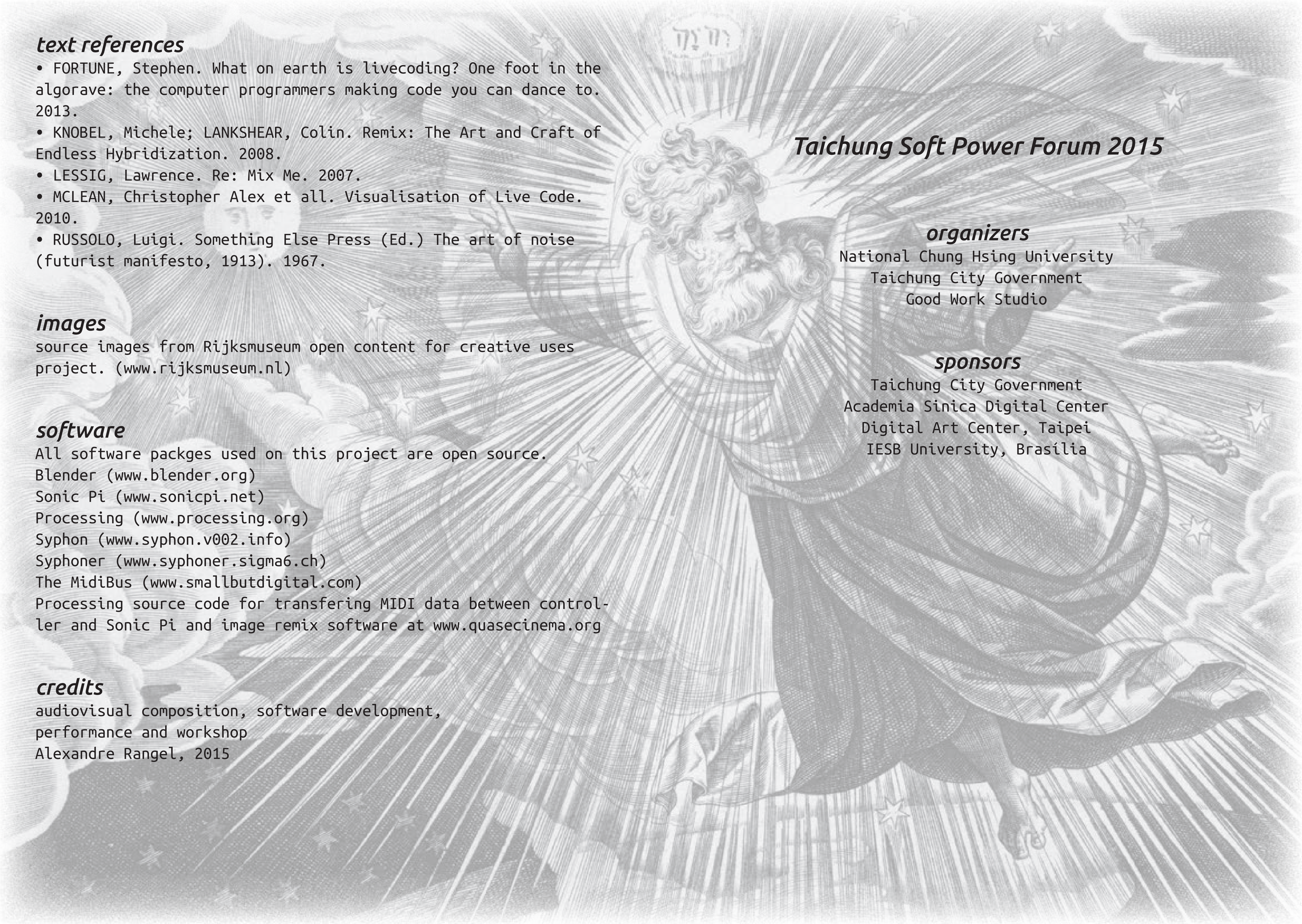
```
  sleep 1

end # kill hat

#------------------------------------------------------------
# track 1 : kick from babel

live_loop :track1 do # kick
  v1 = File.read("/Users/rangel/v1.txt").to_f
  fx1 = File.read("/Users/rangel/fx1.txt").to_f

  spread( 3, 8).each do |b|
    sample :bd_tek, amp: rrand(3.9,4.5)*v1 if b
    sample :bd_ada, amp: rrand(3.7,4.7)*v1 if b
    sleep 1.0/4
  end
  sleep 8 if one_in(64)
  sleep 1 if one_in(72)

end # kick
#------------------------------------------------------------
# track 6 : the message

live_loop :track6 do # voice
  v6 = File.read("/Users/rangel/v6.txt").to_f
  fx6 = File.read("/Users/rangel/fx6.txt").to_f

  sample '/Users/rangel/pisamples/tongues.wav',
    amp: 4.0, attack: 2

  sleep sample_duration '/Users/rangel/pisamples/tongues.wav'

end # message

#------------------------------------------------------------
# track 5 : # kill beep

live_loop :track5 do
  sleep 1.0
```

```
end # kill beep

#------------------------------------------------------------
# track 6 : the massage

live_loop :track6 do # chopped voice
  v6 = File.read("/Users/rangel/v6.txt").to_f
  fx6 = File.read("/Users/rangel/fx6.txt").to_f
   d6 = d6 + 1.0/4
    with_fx :slicer,
    phase: (ring 0.5,0.5,0.5,0.25,0.25,0.25,0.25,0.125)[d6] do
        sample '/Users/rangel/pisamples/tongues.wav',
          start: rand(rand(0.95)), finish: rand(rand(0.95)),
          sustain: (ring 2,2,2,1,1)[d6], release: 2,
          rate: [-0.95,0.95].choose,
      pan: rrand(-0.25,0.25), pan_slide: 0.1,
          amp: 5.0*v6
    end
    sleep (ring 2,2,2,4,0.25)[d6]

end # massage

#------------------------------------------------------------
# track 4 : bass

live_loop :track4 do # bass
  v4 = File.read("/Users/rangel/v4.txt").to_f
  fx4 = File.read("/Users/rangel/fx4.txt").to_f
  time = Time.new
  d4 = d4 + (1.0/64)

  with_fx :flanger, phase: ((time.sec / 100)+0.1),
  mix: rand(0.5) do
    with_fx :lpf, cutoff: rrand(80.0,120.0) do
      sample :bass_thick_c,
        amp: ((ring 3.0,4.4,4.8,5.0)[d4])*v4
      sample :bass_voxy_c, amp: 2.0*v4 if one_in(32)
    end
  end
  sleep 1
```



```
  sleep 1 if one_in(32)
  sleep 6 if one_in(64)

end # bass

#------------------------------------------------------------
# track 1 : kick from running

live_loop :track1 do # kick
  v1 = File.read("/Users/rangel/v1.txt").to_f
  fx1 = File.read("/Users/rangel/fx1.txt").to_f

  sample :bd_ada, pan: -0.1, amp: 4.8*v1
  sleep 0.5

end # kick

#------------------------------------------------------------
# track 3 : tom from running

live_loop :track3 do # tom
  v3 = File.read("/Users/rangel/v3.txt").to_f
  fx3 = File.read("/Users/rangel/fx3.txt").to_f

  d3 = d3 + 1
  with_fx :reverb, room: rrand(0.8,0.9),
  damp: rrand(0.3,0.5) do
    sample :bd_haus, amp: 1.1*v3
  end
  sleep (ring 0.25,0.5,0.25) [d3]

end # tom

#------------------------------------------------------------
# track 4 : bass

live_loop :track4 do
  sleep 12
end # bass
```

```
#------------------------------------------------------------
# track 5 : # running bass

live_loop :track5 do
  v5 = File.read("/Users/rangel/v5.txt").to_f
  fx5 = File.read("/Users/rangel/fx5.txt").to_f
  d5 = d5 + 0.004
  d5 = 0.46 if d5 < 0.2
  d5 = 0.46 if d5 > 0.58

  sample :bass_hard_c, rate: d5,
    pan: rrand(-0.3,0.3), pan_slide: 0.1,
    amp: 2.45*v5

  sleep 0.75
end # running bass

#------------------------------------------------------------
# track 6 : end massage

live_loop :track6 do # bass
  v6 = File.read("/Users/rangel/v6.txt").to_f
  fx6 = File.read("/Users/rangel/fx6.txt").to_f

  d6 = d6 + 0.05
  with_fx :flanger, depth: rrand(6,40), mix: rrand(0.1,0.8) do
    sample :elec_blip, pan: rrand(-0.2,0.2),
      rate: (ring 0.1,0.3,0.8,0.7,0.6,0.8,0.7,0.6,0.3,0.1)[d6],
      amp: rrand(2.5,2.6)*v6
  end

  sleep 1
end # bass
```



text references

• FORTUNE, Stephen. What on earth is livecoding? One foot in the algorave: the computer programmers making code you can dance to. 2013.
• KNOBEL, Michele; LANKSHEAR, Colin. Remix: The Art and Craft of Endless Hybridization. 2008.
• LESSIG, Lawrence. Re: Mix Me. 2007.
• MCLEAN, Christopher Alex et all. Visualisation of Live Code. 2010.
• RUSSOLO, Luigi. Something Else Press (Ed.) The art of noise (futurist manifesto, 1913). 1967.

images

source images from Rijksmuseum open content for creative uses project. (www.rijksmuseum.nl)

software

All software packges used on this project are open source.
Blender (www.blender.org)
Sonic Pi (www.sonicpi.net)
Processing (www.processing.org)
Syphon (www.syphon.v002.info)
Syphoner (www.syphoner.sigma6.ch)
The MidiBus (www.smallbutdigital.com)
Processing source code for transfering MIDI data between controller and Sonic Pi and image remix software at www.quasecinema.org

credits

audiovisual composition, software development,
performance and workshop
Alexandre Rangel, 2015

Taichung Soft Power Forum 2015

organizers

National Chung Hsing University
Taichung City Government
Good Work Studio

sponsors

Taichung City Government
Academia Sinica Digital Center
Digital Art Center, Taipei
IESB University, Brasília

code, perform,
share, transform