

Capire il Multithreading con il Live Coding

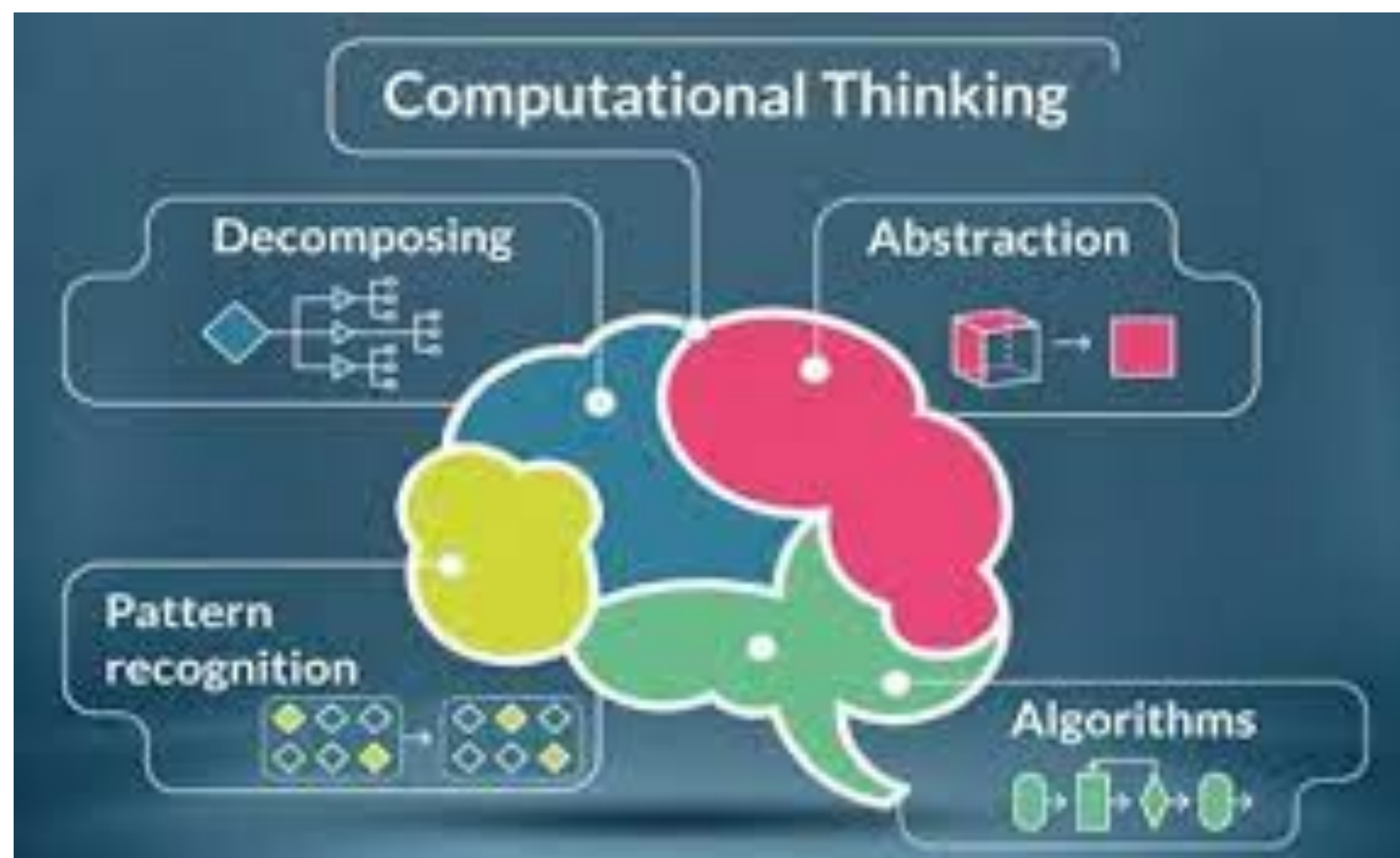
Giorgio Delzanno & Daniele Traversaro



Pensiero Computazionale, Linguaggi, Suoni

Computational thinking is a fundamental skill for everyone, not just for computer scientists. To reading, writing, and arithmetic, we should add computational thinking to every child's analytical ability
[J. Wing 2006]

La programmazione è anzitutto un linguaggio e come tale ha grandi potenzialità anche nell'ambito espressivo ed artistico-estetico inclusa quindi la suoni e musica



“Imagine if the only allowed use of reading and writing was to make legal documents. Would that be a nice world? Now, today’s programmers are all like that” .

“[Programming] is not just a tool for solving complicated equations. It’s also a tool for humans to express ourselves, just like writing and reading is not just a tool for lawyers and legal documents. It’s also a tool for poems, for planes, for rap lyrics, for shopping lists, for lot of different things allow us to be the people that we are”

Sam Aaron, musicista e programmatore,
ideatore di Sonic Pi



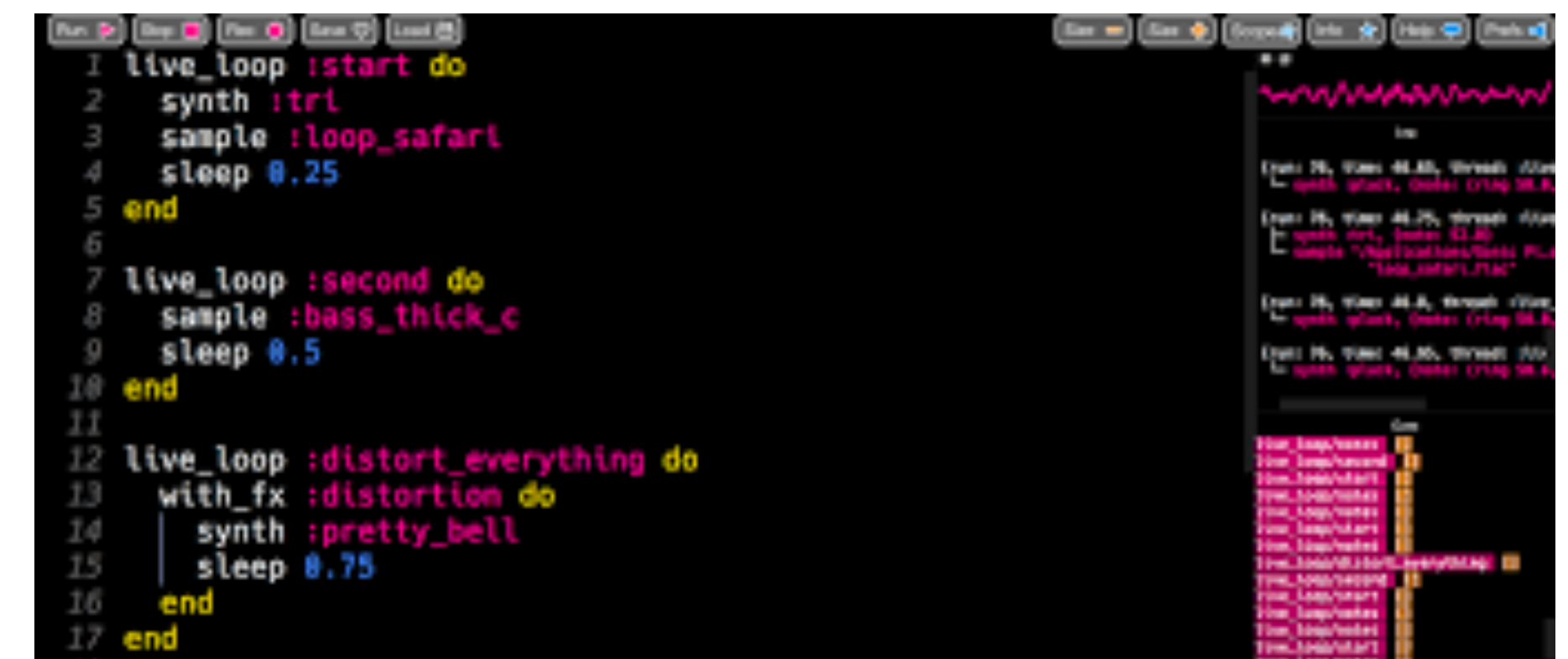
Sonic Pi

L'idea di creare un software per produrre musica attraverso un linguaggio di programmazione nasce nel 2013, dalle ricerche di Sam Aaron presso

l'Università di Cambridge in collaborazione con la Raspberry Pi Foundation

Sonic Pi nasce dunque in primo luogo come strumento educativo parte della suite didattica di Raspberry Pi.

Linguaggio di programmazione come alternativa alla notazione tradizionale!



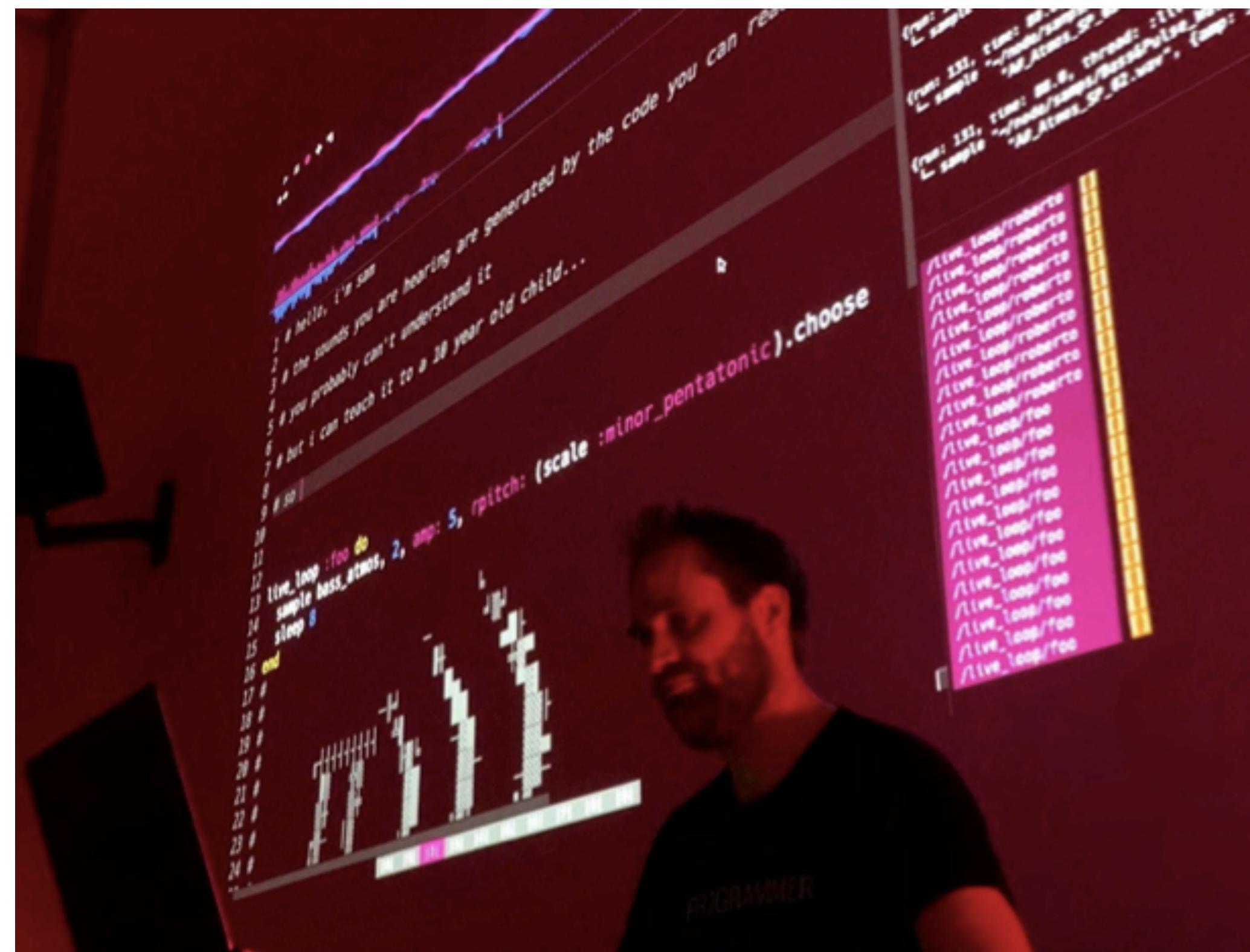
The Live Coding Music Synth for Everyone

<https://sonic-pi.net/>

Oggi Sonic Pi è sviluppato come progetto open source e viene usato in eventi di live coding. Il *live coder* programma in diretta di fronte al pubblico mentre il codice stesso sta girando.

Il live coding è dunque un'attività performativa che prevede estemporaneità, improvvisazione, estro, rischio.

Per rendere evidente questa dimensione i *live coders* proiettano in uno schermo il codice sorgente



Sonic Pi e Multithreading

Il linguaggio musicale per definizione è nato per dare ordine al caos (coordinare un'insieme di strumenti attraverso temporizzazione e sincronizzazione attraverso ad es un direttore d'orchestra)

Sonic Pi, basato su Ruby, fornisce primitive per creare brani da suonare simultaneamente e per definire punti di sincronizzazione per evitare che il pezzo composto non vada out of synch

```
    loop do
      sample :loop_amen
      sleep sample_duration :loop_amen
    end
```

```
3.times do
  play 50
  sleep 0.5
  sample :elec_blup
  sleep 0.5
  play 62
  sleep 0.25
end
```

```
    loop do

      if one_in(2)
        sample :drum_heavy_kick
      else
        sample :drum_cymbal_closed
      end

      sleep 0.5
    end
```

Live Loop e Thread

I live loop sono realizzati con threading e i punti di sincronizzazione sono realizzati con eventi di comunicazione interthread (lock, variabili condition)
Sonic Pi si può anche usare come server (es in locale) inviando comandi per riprodurre suoni da programmi con API come python-sonic


```
in_thread do
  loop do
    sample :drum_heavy_kick
    sleep 1
  end
end

loop do
  use_synth :fm
  play 40, release: 0.2
  sleep 0.5
end
```

```
in_thread do
  loop do
    cue :tick
    sleep 1
  end
end
```

```
in_thread do
  loop do
    sync :tick
    sample :drum_heavy_kick
  end
end
```

|| | | | | | | | ||

Sonic Pi come server

Sonic Pi si può anche usare come server (es in locale) inviando comandi per riprodurre suoni da programmi con API come python-sonic


```
from psonic import *
set_server_parameter('127.0.0.1',4557,4559)
from threading import Thread
```

```
def bass_sound():
    c = chord(E3, MAJOR7)
    while True:
        use_synth(PROPHET)
        play(random.choice(c), release=0.6)
        sleep(0.5)

def snare_sound():
    while True:
        sample(ELEC_SNARE)
        sleep(1)
```

```
bass_thread = Thread(target=bass_sound)
snare_thread = Thread(target=snare_sound)

bass_thread.start()
snare_thread.start()

while True:
    pass
```

Come on, let's play!

