

Live coding with FoxDot and visualization with Processing

s1270115 Yoshiki Sato

Live Coding tools

- Live coding has several environments such as Sonic Pi, TidalCycles, and Foxdot.
- Sonic Pi may be the most elementary tool for code-based music creation and real-time performance.

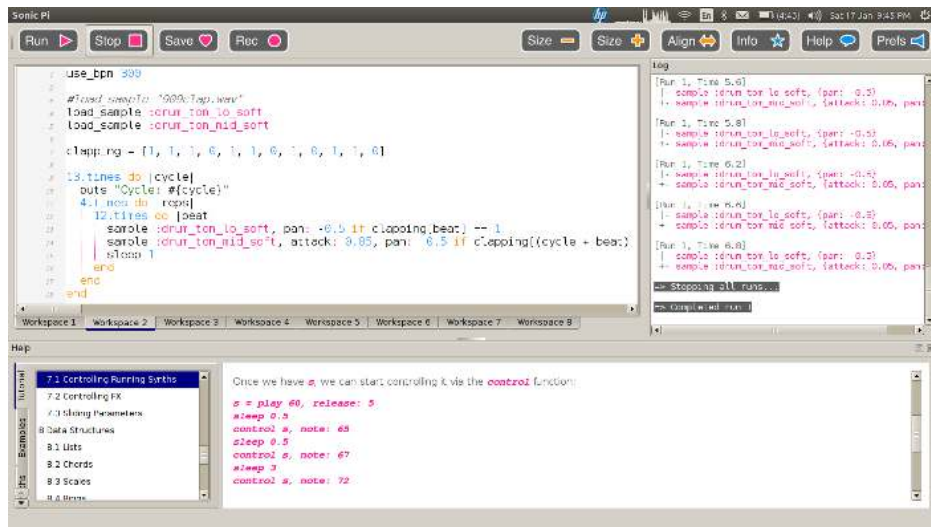


Fig 2. Sonic Pi environment

Live Coding tools

- TidalCycles is the most popular environment for live coding written in Haskell.
- It uses SuperCollider, which is a platform for audio synthesis, to make a sound [2].

```
1
2 d1
3 $every 4 (rev)
4 $sometimesBy 0.2 (slow 2)
5 $sometimesBy 0.8 (jux (iter 4))
6 $stack [
7   sound "ifdrums(3, 8, 0)",
8   sound "bd(3, 8, 3)",
9   sound "glitch(2, 8)"
10 ]
11 #n (irand 64)
12 #pan (rand)
13 #lpf (range 1000 18000 $slow 4 $sine) #resonance "0.2"
14 #delay "0.3" #delaytime "0.25" #delayfeedback "0.5"
15 #gain "1.0"
16
```

Fig 3. A code example of TidalCycles

Live Coding tools

- Foxdot is the library of Python for live coding.
- It also uses SuperCollider, thus Foxdot and TidalCycles are similar.
- The difference between Foxdot and TidalCycles is almost their syntax.

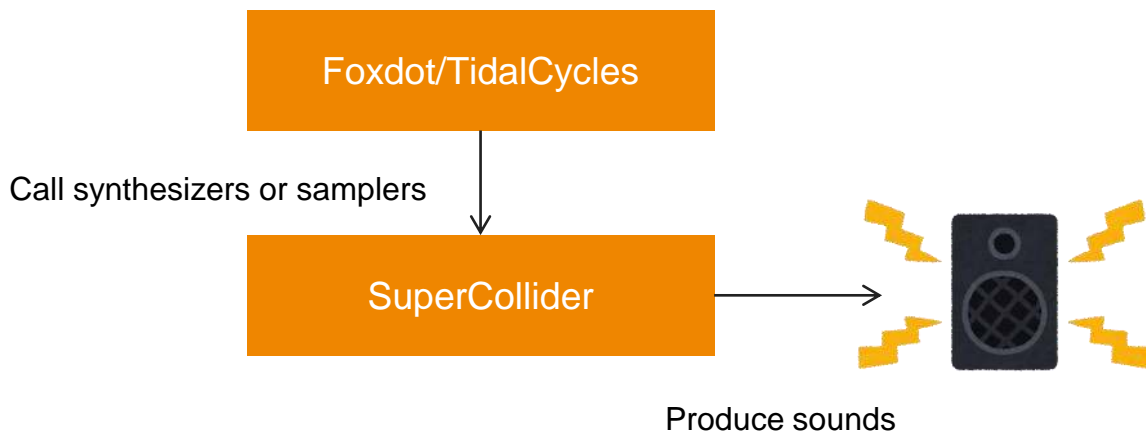


Fig 4. How Foxdot/Tidalcycles works

Visualization tools

- Visualization tools are also diverse.
- The most popular way for visualization is Processing. It contains an IDE, called “sketchbook” and we can develop code there.
- Other tools such as openFrameworks and touchdesigner exist.



Fig 5. Random lines with Processing

How to visualize sound

- OSC (Open Sound Control) is a communication protocol. OSC is used to send and receive music-related data.
- For instance, If synthesizers produce a sound in SuperCollider, Processing can get their information, what synthesizer is making sound, how parameters are set, etc.

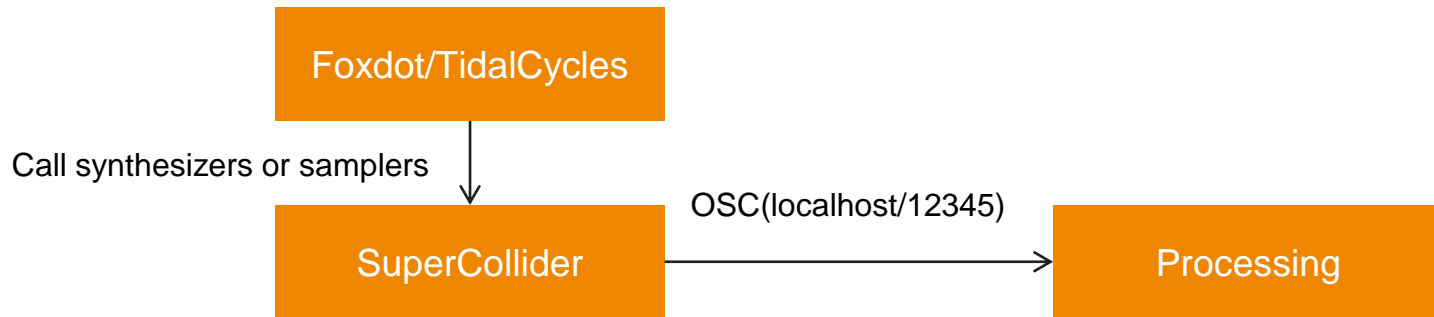


Fig 6. OSC usage in this situation

How to visualize sound

- FoxDot can send any OSC message to an address by OSCClient() [3].
- While TidalCycle also can use OSC using hosc (Haskell Open Sound Control) [4].
- I will use FoxDot since I am not familiar with Haskell.

How to visualize sound

- On FoxDot, I will use three instruments: dirt, pluck, play.
- When an instrument produces a sound, processing will describe a figure.
- Stereo sound intensity is sent to Processing and visualized without OSC.

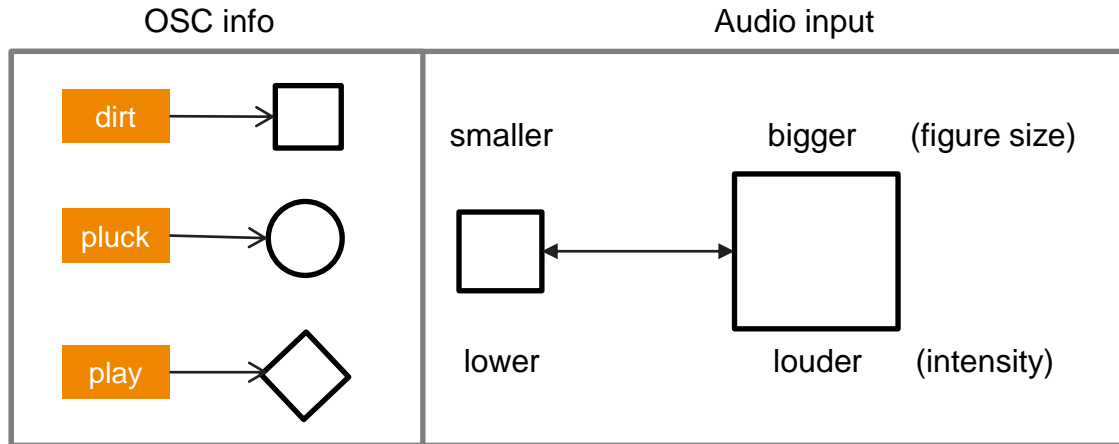


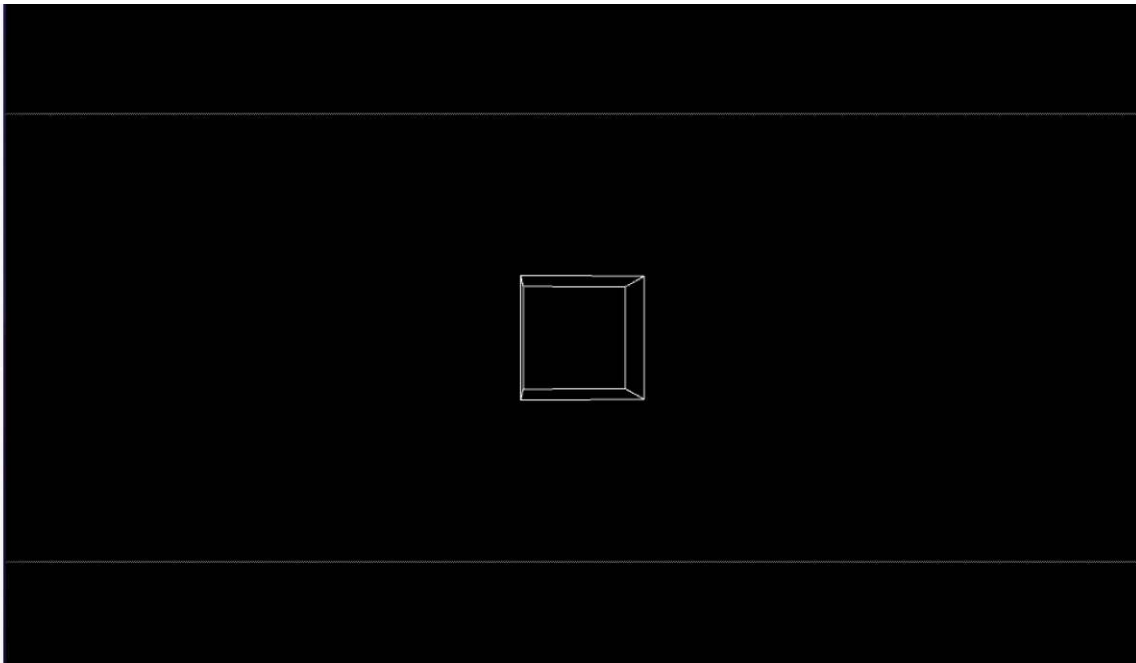
Fig 7. How sound parameters are visualized

Demo Movie

```
1 Clock.bpm=135
2
3 Server.add_forward("localhost",12345)
4
5 s1 >> dirt([-3,-2.8],dur=[0.5,1,0.25,0.75,0.25,0.25,0.75,0.25],amp=0.8,pan=[-.25,.25],lpf=250)
6
7 s1 >> dirt([1.2,1.0],dur=[0.5,1,0.25,0.75,0.25,0.25,0.75,0.25],amp=0.8,pan=[-.25,.25],lpf=250)
8
9
10 s2 >> pluck([23.5],dur=[.25,.75,.75,.5,.75,.75,.25],amp=[0.2],pan=[1,0,-1],hpf=10000)
11
12 s3 >> pluck([-3,-2.8,-4,-2.7],dur=[0.75,1/3,1/6,.5,.25],pan=[1,0,-1],lpf=4000,amp=[0.2,0.4,0.3,0.5])
13
14 s3 >> pluck([-3,-2.8,-4,-2.7],dur=[0.75,1/3,1/6,.5,.25],pan=[1,0,0,-1],lpf=1000,amp=[0.2,0.4])
15
16 d1 >> play('[-]',dur=1/2,delay=0.01,amp=[0,0.1],pan=0,sample=1)
17
18 d2 >> play('[X]',dur=1,amp=1.3,pan=0,sample=0,lpf=200)
19
20
21 Clock.stop()
```

Welcome to FoxDot! Press Cmd+K for help.

```
>>> Clock.bpm=135
>>> Server.add_forward("localhost",12345)
```



Demo Movie is available on



https://drive.google.com/file/d/1CO_nF8ZgVUNDw5-ucx6M0rozT86LnGog1/view?usp=sharing

Result / Discussion

- Live coding with real-time visualization was done.
- OSC messages that are produced by FoxDot contains much are complicated and unclear. If I understand them, visualization may be done by OSC only, without referring audio input.
- Processing may be used in other projects such as my current study, Data pattern sonification, since processing works on Browsers by p5.js.

References

- [1] NILSON, Click. Live coding practice. In: *Proceedings of the 7th international conference on New interfaces for musical expression*. 2007. p. 112-117.
- [2] SuperCollider. “SuperCollider » SuperCollider.” *SuperCollider » SuperCollider*, <https://supercollider.github.io/>. Accessed 11 Mar. 2022.
- [3] “Changelog — FoxDot 0.5.9 Documentation.” *Welcome to FoxDot’s Documentation! — FoxDot 0.5.9 Documentation*, <https://foxdotcode.readthedocs.io/en/latest/changelog.html?highlight=osc#v0-5-2-fixes-and-updates>. Accessed 11 Mar. 2022.
- [4] “Hosc: Haskell Open Sound Control.” *Hackage*, <https://hackage.haskell.org/package/hosc>. Accessed 11 Mar. 2022.