livecoding, musique, python, em

FoxDot

FoxDot est un langage de livecoding qui permet d'écrire de la musique, de la transformer et de la jouer en temps réel.
FoxDot est développé en python par Ryan Kirkbride

Installation

Voir

- https://github.com/Qirky/FoxDots
- https://foxdot.org/installation/

Installation de supercollider et foxdot sur debian stetch 9.5

```
# installer supercollider
sudo apt-get install supercollider

# installer les sc3-plugins en suivant https://github.com/supercollider/sc3-plugins

# démarrer supercollider : démarrer / son et vidéo / supercollider IDE
# dans supercollider
# Quarks.install("https://github.com/Qirky/FoxDotQuark.git")
# Quarks.install("https://github.com/supercollider-quarks/BatLib.git")

pip3 install --user FoxDot --upgrade
sudo apt-get install python3-tk
```

Démarrage

```
# démarrer supercollider : démarrer / son et vidéo / supercollider IDE
# dans supercollider
# écrire dans la partie gauche de la fenêtre : FoxDot.start , pui faire CTRL+ENTER sur la ligne pour l'exécuter
# puis dans un terminal
python3 -m FoxDot &
```

Trucs et astuces

aide-mémoire

Un bloc de code est composé de lignes qui se suivent sans ligne vide, il est exécuté d'un coup.
On exécute une ligne ou un bloc avec Ctrl + ↵ Enter après avoir placé le curseur sur la ligne
Un «player» est toujours composé de deux caractères, pour jouer à plusieurs on recommande d'utiliser une lettre (initiale du pseudo) + un chiffre, exemples : d1, f4, etc.
Par défaut, Foxdot fonctionne avec les paramètres suivants :

- tempo à 120 BPM
- tonalité en Do majeur
- signature de temps en 4/4

Un rythme

```
Clock.bpm = 120

h1 >> play("x      xx x     ")
h2 >> play("    o       o   ")
h3 >> play("- - - - - - - - ")

h3 >> play("- - - - - = - = ")

h1.stop(); h2.stop(); h3.stop();
```

Deux instruments, le second suit le premier

```
y1 >> bass([0,2,3,4], dur=1/2, slide=0)
g1 >> varsaw(amp=0.3, oct=5).follow(y1)

y1.stop() ; g1.stop()
```

Stopper tous les players

```
Clock.clear()
```

Une suite d'accords (avec des patterns)

```
ac >> space(P[(0,3,5), (1,4,6), (2,5,7), (4,7,9)], dur=4)
```

Réinitialiser un player

```
ex.reset()
```

Ajouter un attribut

```
g1 >> varsaw(amp=0.6, oct=5, tremolo=3).follow(y1)
y1 >> bass([0,2,3,4], amp=1, dur=[4, 1/2, 1/2, 3], slide=0, chop=5)
```

Afficher le BPM

```
print(Clock.bpm)
```

Changer la gamme

```
Scale.default.set("minor")
# Identique à :
Scale.default = "minor"
```

Instruments, attributs, effets, samples, gammes

```
print(SynthDefs)               # afficher tous les instruments
print(Player.get_attributes()) # afficher tous les attributs
print(FxList)                  # afficher tous les effets
print(Samples)                 # afficher tous les échantillons
print(Scale.names())           # afficher les gammes
```

Ce qui donne avec la version 0.8.3 :

foxdot : tous les instruments

foxdot_0.8.3_tous_les_instruments.txt

```
['bass', 'orient', 'spark', 'keys', 'glass', 'arpy', 'dbass', 'zap', 'nylon', 'blip', 'saw', 'loop', 'play2', 'scratch', 'marimba',
'bell', 'soft', 'scatter', 'lazer', 'sitar', 'play1', 'fuzz', 'pasha', 'creep', 'swell', 'feel', 'quin', 'dirt', 'squish', 'bug',
'pads', 'sinepad', 'dab', 'jbass', 'donk', 'crunch', 'razz', 'prophet', 'charm', 'stretch', 'snick', 'space', 'gong', 'soprano',
'rave', 'ripple', 'sawbass', 'pulse', 'karp', 'star', 'noise', 'ambi', 'audioin', 'varsaw', 'klank', 'pluck', 'growl', 'dub',
'twang', 'viola']
```

foxdot : tous les attributs

foxdot_0.8.3_tous_les_attributs.txt

```
('degree', 'oct', 'freq', 'dur', 'delay', 'buf', 'blur', 'amplify', 'scale', 'bpm', 'sample', 'env', 'sus', 'fmod', 'pan', 'rate',
'amp', 'midinote', 'channel', 'vibdepth', 'vib', 'slide', 'slidedelay', 'sus', 'slidefrom', 'glidedelay', 'glide', 'bend',
'benddelay', 'coarse', 'rate', 'striate', 'buf', 'pshift', 'hpf', 'hpr', 'lpr', 'lpf', 'swell', 'bpnoise', 'bpr', 'bpf', 'chop',
'tremolo', 'beat_dur', 'echotime', 'echo', 'spin', 'cut', 'mix', 'room', 'formant', 'shape', 'drive')
```

foxdot : tous les effets

foxdot_0.8.3_tous_les_effets.txt

```
<'bandPassFilter': keyword='bpf', other args=['bpnoise', 'bpr', 'sus']>
<'chop': keyword='chop', other args=['sus']>
```

```
<'coarse': keyword='coarse', other args=['sus']>
<'combDelay': keyword='echo', other args=['echotime', 'beat_dur']>
<'filterSwell': keyword='swell', other args=['sus', 'hpr']>
<'formantFilter': keyword='formant'>
<'glissando': keyword='glide', other args=['glidedelay', 'sus']>
<'highPassFilter': keyword='hpf', other args=['hpr']>
<'lowPassFilter': keyword='lpf', other args=['lpr']>
<'overdriveDistortion': keyword='drive'>
<'pitchBend': keyword='bend', other args=['benddelay', 'sus']>
<'pitchShift': keyword='pshift'>
<'reverb': keyword='room', other args=['mix']>
<'slideFrom': keyword='slidefrom', other args=['slidedelay', 'sus']>
<'slideTo': keyword='slide', other args=['slidedelay', 'sus']>
<'spinPan': keyword='spin', other args=['sus']>
<'striate': keyword='striate', other args=['rate', 'buf', 'sus']>
<'tremolo': keyword='tremolo', other args=['beat_dur']>
<'trimLength': keyword='cut', other args=['sus']>
<'vibrato': keyword='vib', other args=['vibdepth']>
<'wavesShapeDistortion': keyword='shape'>
```

foxdot : tous les samples

foxdot_0.8.3_tous_les_samples.txt

```
'!': Yeah!
'#': Crash
'$': Beatbox
'%': Noise bursts
'&': Chime
'*': Clap
'+': Clicks
'-': Hi hat closed
'/': Reverse sounds
'1': Vocals (One)
'2': Vocals (Two)
'3': Vocals (Three)
'4': Vocals (Four)
':': Hi-hats
'=': Hi hat open
'@': Gameboy noise
'A': Gameboy kick drum
'B': Short saw
'C': Choral
'D': Dirty snare
'E': Ringing percussion
'F': Trumpet stabs
'G': Ambient stabs
'H': Clap
'I': Rock snare
'J': Ambient stabs
'K': Percussive hits
'L': Noisy percussive hits
'M': Acoustic toms
'N': Gameboy SFX
'O': Heavy snare
'P': Tabla long
'Q': Electronic stabs
'R': Metallic
'S': Tamborine
'T': Cowbell
'U': Misc. Fx
'V': Hard kick
'W': Distorted
'X': Heavy kick
'Y': High buzz
'Z': Loud stabs
'\\': Lazer
'^': 'Donk'
'a': Gameboy hihat
'b': Noisy beep
'c': Voice/string
'd': Woodblock
'e': Electronic Cowbell
'f': Pops
'g': Ominous
'h': Finger snaps
'i': Jungle snare
'j': Whines
'k': Wood shaker
'l': Robot noise
'm': 808 toms
'n': Noise
'o': Snare drum
'p': Tabla
'q': Ambient stabs
'r': Metal
's': Shaker
't': Rimshot
'u': Soft snare
'v': Soft kick
```

```
'w': Dub hits
'x': Bass drum
'y': Percussive hits
'z': Scratch
'|': Hangdrum
'~': Ride cymbal
```

foxdot : toutes les gammes

foxdot_0.8.3_toutes_les_gammes.txt

```
['aeolian', 'altered', 'bebopDom', 'bebopDorian', 'bebopMaj', 'bebopMelMin', 'blues', 'chinese', 'chromatic', 'custom', 'default',
'diminished', 'dorian', 'dorian2', 'egyptian', 'freq', 'halfDim', 'halfWhole', 'harmonicMajor', 'harmonicMinor', 'hungarianMinor',
'indian', 'justMajor', 'justMinor', 'locrian', 'locrianMajor', 'lydian', 'lydianAug', 'lydianDom', 'lydianMinor', 'major',
'majorPentatonic', 'melMin5th', 'melodicMajor', 'melodicMinor', 'minMaj', 'minor', 'minorPentatonic', 'mixolydian', 'phrygian',
'prometheus', 'romanianMinor', 'susb9', 'wholeHalf', 'wholeTone', 'yu', 'zhi']
```

Et une petite cheatsheet à télécharger :

foxdot cheatsheet

Divers

Ajouter des répertoires de samples
on peut modifier le fichier ci-dessous pour indiquer de nouveaux répertoires qui pourront être utilisés avec play (plutot que loop)

```
/home/emoc/.local/lib/python3.5/site-packages/FoxDot/lib/Buffers.py
```

Les échantillons sont rangés dans

```
/home/emoc/.local/lib/python3.5/site-packages/FoxDot/sndv
```

Ressources

FAQ : https://forum.toplap.org/t/frequently-asked-questions/504

Forum anglophone : https://forum.toplap.org/c/communities/foxdot/17

Tutoriels pour workshops : https://github.com/Qirky/FoxDot-Worksheet

Quelques exemples :

- exemples en moins de 10 lignes : https://github.com/Qirky/ten-lines-or-less
- snippets : https://forum.toplap.org/t/share-your-snippets/711

article de l'auteur sur Foxdot : http://users.sussex.ac.uk/~thm21/ICLI_proceedings/2016/Colloquium/68_FoxDot.pdf

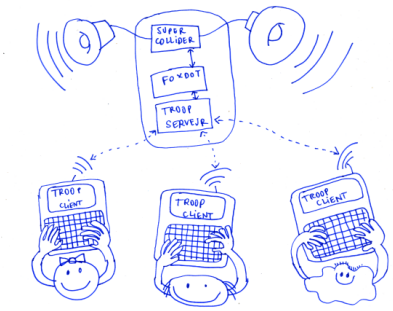
D'autres ressources sur : https://fablabo.net/wiki/Livecoding et https://ressources.labomedia.org/live_coding

Troop

Troop est un serveur qui permet de jouer à plusieurs, collaborativement. Il peut s'utiliser avec FoxDot ou d'autres langages de livecoding.

Installation de Troop : https://github.com/Qirky/Troop/

Il est recommandé d'utiliser la même version de Troop sur chaque machine qui se connecte au serveur et que le serveur et les clients soient de la même version.



Article extrait de : http://lesporteslogiques.net/wiki/ - **WIKI Les Portes Logiques**
Adresse : **http://lesporteslogiques.net/wiki/ressource/logiciel/foxdot_troop/start?rev=1632055600**
Article mis à jour: **2021/09/19 14:46**