

# Ajouter de nouveaux synthés à Sonic Pi

**Sonic Pi** est un logiciel parfait pour débuter la programmation tout en apprenant à composer.

Celui-ci est en fait lié à *SuperCollider* : il en reprend le principe mais le langage est beaucoup plus simple à appréhender.

**Sonic Pi** intègre de base de nombreux synthés que l'on peut utiliser pour faire de la musique, à l'image de l'émulation du célèbre TB303 :

```
synth :tb303
```

Malheureusement, on fait assez vite le tour des synthés proposés par défaut, et l'on peut souhaiter en créer nous même pour pouvoir les utiliser.

J'ai suivi [ce tutoriel](#) pour ce faire, et il est bluffant de voir à quel point c'est facile à faire !

**Voici une variante de l'exemple donné, corrigé** (un seul anti-slash dans le symbole de la *SynthDef*) et **amendé** pour être plus facile à utiliser. Il vous faudra **créer un dossier (par exemple nommé *SCDef*)**, et **enregistrer ce script à l'intérieur**, par exemple sous le titre *SCDefGeneration.scd*

```
(  
SynthDef(\piTest, { |freq = 200, amp = 1, out_bus = 0|  
  
  var son = SinOsc.ar([freq,freq], 0, 0.5) * Line.kr(1, 0, 5, amp, doneAction: 2);  
  
  Out.ar(out_bus, son);  
  
}).writeDefFile(thisProcess.nowExecutingPath.dirname ++ "/");  
)
```

Lorsque vous interprétez ce code dans **SuperCollider**, avec **ctrl + Entrée**, un fichier binaire est créé dans le dossier **SCDef**. Celui-ci contient le synthé à utiliser par **Sonic Pi**.

Pour enregistrer une autre **SynthDef**, il faudra la substituer à celle de l'exemple, autrement dit copier-coller la SynthDef, du mot SynthDef(, jusqu'à la parenthèse fermante, puis rajouter ".writeDefFile(thisProcess.nowExecutingPath.dirname ++ "/");" à la fin, avant d'interpréter le code pour créer le fichier binaire.

Pour charger ce synthé dans **Sonic Pi**, il faudra interpréter la commande suivante dans **Sonic Pi** après l'avoir lancé, et ce à chaque fois :

```
load_synthdefs "/chemin/vers/SCDef/"
```

Cette commande chargera dans **Sonic Pi** l'ensemble des **SynthDefs** présentes dans le dossier.

Après cela, on peut utiliser le synthé dans **Sonic Pi** comme d'habitude, mais celui-ci ne propose ni autocomplétion ni documentation. Attention, a priori il faut également mettre son nom entre guillemets pour qu'il soit reconnu :

```
synth 'piTest', freq: 330, amp: 0.5
```

Les différents paramètres accessibles depuis **Sonic Pi** pour moduler le synthé sont les arguments de la fonction associée à la **SynthDef**, c'est-à-dire ceux encadrés par des barres verticales, ou suivant le mot clef *arg*, tout en haut du code :

```
|freq = 200, amp = 1, out_bus = 0|
```

ou

```
arg freq = 200, amp = 1, out_bus = 0;
```

Attention, dans certains cas, certaines **SynthDef** peuvent utiliser des commandes qui ne fonctionnent pas avec **Sonic Pi**. Le résultat n'est donc pas garanti.

Si vous avez lu jusque là, je pense qu'une des prochaines étapes pourrait être de farfouiller parmi ces liens :

<https://github.com/SCLOrkHub/SCLOrkSynths/tree/master/SynthDefs>

[https://github.com/dakyri/synthdef\\_collection](https://github.com/dakyri/synthdef_collection)

<https://github.com/everythingwillbetakenaway/Synthdefs>

[https://github.com/theseanco/awesome-synthdefs/blob/master/co34pt/SynthDefs/co34pt\\_synthdefs.scd](https://github.com/theseanco/awesome-synthdefs/blob/master/co34pt/SynthDefs/co34pt_synthdefs.scd)

<https://github.com/supercollider-quarks/SynthDefPool/tree/master/pool>

<https://github.com/loopier/synthdefs>

...

---

<https://github.com/search?q=synthdef>

<https://sccode.org/>

<https://scsynth.org/>

---

Bon son !

---

Révision #1

Créé 3 août 2023 14:45:38 par Rachelle

Mis à jour 3 août 2023 14:48:45 par Rachelle