

ATELIER : INTRODUCTION AU LANGAGE DE PROGRAMMATION FAUST

Yann ORLAREY

GRAME, Centre national de création musicale
9 rue du Garet
69202 Lyon,
France,
orlarey@grame.fr

RÉSUMÉ

FAUST est un langage de programmation fonctionnel synchrone spécialement conçu pour le traitement du signal et la synthèse de sons en temps réel. L'objectif de l'atelier est de proposer une introduction simple au langage et à ses possibilités, notamment en relation avec les autres environnements musicaux. Les participants qui le souhaitent peuvent venir avec leur machine.

1. PRÉSENTATION

FAUST [1] est un langage de programmation spécialisé, conçu pour décrire de manière concise des algorithmes de synthèse et de traitement du son. Un programme FAUST décrit un processeur de signaux, c'est à dire une fonction, au sens mathématique du terme, qui prend des signaux en entrée et produit des signaux en sortie.

L'une des caractéristiques de FAUST, contrairement aux autres langages musicaux, est d'être entièrement compilé. On peut donc tout à fait utiliser FAUST à la place de C pour écrire par exemple des plugins audio. Les techniques de compilation mise en oeuvre sont très optimisées et permettent de générer du code de qualité, dont l'efficacité est généralement comparable à du code C écrit à la main.

Le système d'architecture de FAUST [2] facilite le déploiement des programmes et permet de générer, à partir d'un même source, du code pour les principales plateformes audio : MaxMSP, VST, Puradata, Csound, SuperCollider, etc.

FAUST est un langage textuel. Sa syntaxe est basée sur l'idée de *composition* de processeurs de signaux. Ainsi par exemple si A et B sont deux processeurs de signaux, (A:B) représente le processeur de signaux obtenu en branchant les sorties de A sur les entrées correspondantes de B. Tandis que (A,B) représente la mise en parallèle de A et B.

Un programme FAUST est constitué d'un ensemble de définitions dont celle du mot clef `process`, l'équivalent de `main()` en C.

Par exemple le programme suivant comporte 3 définitions et décrit un générateur de bruit blanc dont le niveau est contrôlé par un réglage de volume dont la valeur par

défaut est 0, qui prend des valeurs entre 0 et 1 avec un pas de variation de 0.01.

```
noise = random / 2147483647.0;  
random = +(12345) ~ *(1103515245);  
process = noise * hslider("volume", 0, 0,  
1, 0.01);
```

Il est possible d'associer, en parallèle de l'interface graphique, d'autres interfaces de contrôles comme OSC ou HTTP. L'exemple qui suit est une variante du précédent où l'on indique que le slider de volume est contrôlé également par la réception du message OSC `/accxyz` ce qui permet de piloter ce slider à distance via un smartphone et l'application TouchOSC.

```
noise = random / 2147483647.0;  
random = +(12345) ~ *(1103515245);  
process = noise * hslider("volume[osc:/  
accxyz/0_-10_10]", 0, 0, 1, 0.01);
```

La façon la plus simple de tester FAUST est d'utiliser le compilateur en ligne <http://Faust.grame.fr>. Depuis la page d'accueil cliquez sur *Online Examples*. Après avoir fermé la fenêtre de bienvenue choisissez dans la rubrique *Effects* le programme *freeverb*. Cliquez ensuite sur l'onglet *C++ code* pour choisir l'architecture souhaitée, par exemple *VST*. Enfin allez sur l'onglet *Exec File* pour déclencher la compilation et récupérer le plugin VST prêt à l'emploi.

Le site contient également toutes les informations nécessaires pour télécharger FAUST et l'installer sur une machine Linux ou MacOSX.

2. REFERENCES

- [1] Orlarey, Y., Fober, D. et Letz, S. "An algebra for block diagram languages", *Proceedings of the International Computer Music Conference (ICMA)*, Gothenburg, Suède, 2002.
- [2] Fober, D., Orlarey, Y. et Letz, S. "FAUST architecture design and OSC support", *Proceedings of the Conference on Digital Audio Effects (DAFx-11)*, IRCAM, Paris, France, 2011.