

Initiation a la recherche

**Gédoux Architecte multimédia   
Gautier  
 N2**

La synthèse additive avec Clojure

Table des matières

Introduction4

Etat de l’art5

Contexte et environnement de travail1

Clojure2

Overtone2

Harmonikit2

Présentation du projet1

Analyse du programme2

L’OSC et ses outils2

Problèmes rencontré2

Amélioration2

Ouverture2

Conclusion1

Introduction

Dans le cadre de mon module d’initiation a la recherche il m’a été demander de travailler sur un projet bien particulier : Harmonikit, le synthétiseur additif que Rich Hickey à présenté lors de sa dernière conférence a l aClojureCon.

Ce prjet a été un défi de taille pour moi, en effet ayant a la base une formation électronique, je me suis pris d passion pour l’informatique après mon stage orienté dans ce domaine. Plus encore j’ai du apprendre un language très original qui a une structure totalement différente des autres anguage que j’ai pu utiliser auparavant.

Je me suis vraiment attéché ce projet car il concerne un domaine qui me passionne depuis mon enfance : la musique électronique.

Dans ce rapport je vais vous présenter les envirronement sur lequels j’ai travailler et parler un peu de leurs spécificité puis je vous parlerais de la réalisation de mon projet et de l’evolution des mes objectifs en fonctions de mes avancés, de mes découvertes et des difficultès rencontrées tout au long de ces six mois.

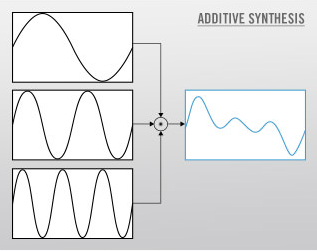
Etat de l’art

Lorsque l’on veut aborder un projet avec une démarche nouvelle, notamment dans un projet de recherche, il est important de réaliser un etat de l’art de ce qui a été fait avant celui-ci. Un état de l’art rassemble nos recherche sur otut ce qui a été fait avant dans le thème abordé par notre projet.

## La synthèse additive

Pour réaliser mon état de l’art je me suis otut d’abord intérréssé à la synthèse additive, le synthétiseur sur lequel je devais travailler etant basé sur ce principe.

La synthèse additive consiste a créer un signal périodique en superposant plusieures sinusoides harmoniques. Elle est bas’ sur le principe de Fourier qui nous dit que tout signal périodique peut être théoriquement décomposé en une somme de sinus et cosinus de fréquences multiples a la fréquence fondamentale du signal.



A l’époque la synthèse additive etait peu utilisée car très énergivore en terme d’utilisation des processeur, on lui préférait donc la synthèse soustractive , qui pars du principe inverse, filtrer un signal périodique très riche a l’aide de filtre fréquentiels pour obtenir le son voulu.

Mais déhormais grace aux progrès de l'informatique en matière de puissance de calcul,

la Synthèse additive commence a ètre de plus en plus utilisé sur les synthétiseurs numérique.

L’inconvénient pricipale de la synthèse additive est l’importance d’avoir un nombre d’oscillateur conséquent de manière a créé des sons sintéréssant qui se détache des sons pures donné par les sinusoides.

Ce type de synthèse requiert donc une innovation dans la manière de controler ces sinusoide, la première tentative a été réalisé par Iannis Xenakis , compositeur de musique concrete réputé (qui est pour la petite anecdote le grand père de ma copine) avec l’UPIC, une talbe graphique permettant d’actionner les oscilatteur en dessiannt de lignes et des points.

Différentes méthodes ont ensuite vu le jour et sont implémenté dans certains synthétiseur que je vais vous présenter ci dessous.

## Quelques Synthétiseurs additifs

### Le Parsec



Le Parsec comporte un nombre impressionnant de 1024 Partiels (Les sinusoides harmoniques), qui sont reparties sur deux oscillateurs principaux.

Si le parsec est innovant avec son nombre de sinusoide utilisé, il ne l’est pas dans son fonctionnement : en effet il utilise exactement les même contrôle qu’un synthétiseur soustractif lambda.

### Le Phosphor



Le Phosphor est un synthetiseur un peu différent : il utilise 26 partiels, réparti sur deux oscillateur, on peut a l’instar du synthétiseur sur lequel j’ai travaillé modifier l’amplitude de chaque partiels séparément. Pour le reste, l’envollope et les effets s’effectuent sur les seuls oscillateur.

### Le Razor



Le Razor est, en terme de contrôle le synthétiseur additif le plus inovant qu’il m’est été donné de voir.

Il utilise jusqu’à 320 partiels, mais ce qui fait sa force c’est que l’intégralité des effets et modulation qu’il propose on aussi été crées sur le principe de la synthèse additive. Ce qui donne des effets completement nouveaux comme le Formant qui permet de faire rentrer en résonnance tel ou tel partiels, et bien d’autres effets mais il serait un peu fastidieux de tous les décrie ici.

Contexte et environnement de travail

## Clojure

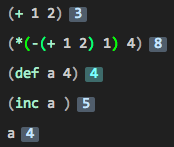
Clojure est un language Lisp (pour list processing) : c’est un langage informatique de traitement de list. Il a été inventé par Rich Hickey,,fondateur de Conitech, car ce dernier voulait avoir un langage Lisp qui soit totalement compatible avec la plateforme Java.

Le développement via Clojure se fait à l’aide d’un REPL (pour Read-Eval-Print-Loop)

Qui se comporte exactement comme un terminal : une fos l’instruction executé, elle est évalué puis compilé, et la boucle revient au point de départ et redemande une nouvelle instruction.

Ce Type de compilation est pour ma part très efficase. Contrairement a d’autres langages comme le le langage C où il faut compiler l’ensemble du programme pour le faire fonctionner,.si il y a une erreur, il faut tout recompiler apres correction.

Ici si par exemple vous avez compilé l’ensemble du programme, et que vous voulez modifier une fonction ou corriger une erreur, vous n’avez besoin de recompiler que la partie du programme qui pose problème. Le gain de temps est consequent.

Nous avons ici un exemple de différentes commandes simple.

Comme on peu le voir avec la variable « a », appliquer une action sur cette variable ne modifie que sa valeur au moment de celle-ci.

C’est une des particularité fondamentale de Clojure et c’est une des caractéristiques qui en font un langage très efficase.

Personnellement j’ai eu beaucoup de mal au départ a comprendre ce type de fonctionnement contrairement au java par exemple ou chaque variable est stocké et est modifié dès que nous le voulons.

## Overtone

Overtone est une bibliothèque de Clojure qui utilise le moteur audio de Supercollider pour creér une multitude d’objet sonore

## Harmonikit

Présentation du projet

## Analyse du programme

## L’OSC et ses outils

## Problèmes rencontré

## Amélioration

## Ouverture

Conclusion

Bibliographie