

## Numéros / n° 1 - automne 2011

# « QSox : un éditeur audio libre, multiplateforme et adapté au traitement par lots »

**Julien Bréval et Benoît Courribet**

Résumé

Dans cet article, nous présentons le projet de développement du logiciel QSox. Ce logiciel est un éditeur audio libre, multiplateforme et destiné à être utilisé dans le cadre de traitements par lots de fichiers audionumériques. Il propose une interface graphique pour SoX.

## 1. Introduction

QSox est un éditeur audionumérique libre, gratuit et multiplateforme possédant une interface graphique et orienté vers le traitement par lots de fichiers audionumériques (ou *batch processing*) <sup>(1)</sup>. Il est téléchargeable à l'adresse suivante <http://qsox.sourceforge.net/>

Dans cet article, nous présentons les fonctionnalités de QSox, puis la généalogie du projet, ainsi que l'architecture informatique retenue. Enfin, nous proposons quelques évolutions à considérer pour ses futurs développements, en espérant que ce logiciel puisse intéresser la communauté des utilisateurs de l'informatique musicale et celle des développeurs de logiciels libres.

Un éditeur audionumérique est un logiciel permettant d'effectuer des traitements numériques du signal sur des fichiers audio. QSox se démarque des éditeurs audio les plus connus <sup>(2)</sup> par les points suivants :

- QSox a été développé dans l'optique de fournir un logiciel très simple et rapide d'utilisation, sans visualisation des données audionumériques (pas de représentation graphique de la forme d'onde ni du spectre fréquentiel)
- QSox a été prévu pour être utilisé dans le cas de traitements par lots de fichiers.

Dans le cas de QSox, le traitement par lots consiste à appliquer un même traitement ou une même chaîne de traitements à plusieurs fichiers audionumériques. Cette fonction est surtout utilisée lorsque l'on doit traiter un grand nombre de fichiers audio (il s'agit souvent de plusieurs dizaines, voire centaines de fichiers).

Le choix du nom "QSox" est justifié par les deux points suivants :

- QSox utilise le logiciel SoX <sup>(3)</sup> comme moteur de traitements audionumériques (en ce sens, QSox est une interface graphique particulière pour SoX)
- QSox a été développé avec l'environnement de développement multiplateforme Qt <sup>(4)</sup>.

## 2. Présentation de QSox

## 2. 1. Introduction à QSox

Grâce à une interface graphique simple, QSox permet de choisir un ou plusieurs fichiers audio, puis une chaîne de traitements audionumériques à appliquer à ces fichiers. On peut écouter le rendu des traitements en temps réel, afin de vérifier que la chaîne de traitements est correcte. Une fois ces opérations effectuées, la chaîne de traitements est appliquée à chacun des fichiers (l'un après l'autre).

QSox est utilisable immédiatement une fois qu'il a été téléchargé : il ne nécessite aucune installation ni redémarrage, et ne demande pas de télécharger ni de lancer SoX au préalable. Il n'y a qu'un seul fichier à télécharger : il s'agit d'une archive comprenant tous les fichiers nécessaires ; ces fichiers doivent simplement se trouver dans un même répertoire, qui peut être placé n'importe où.

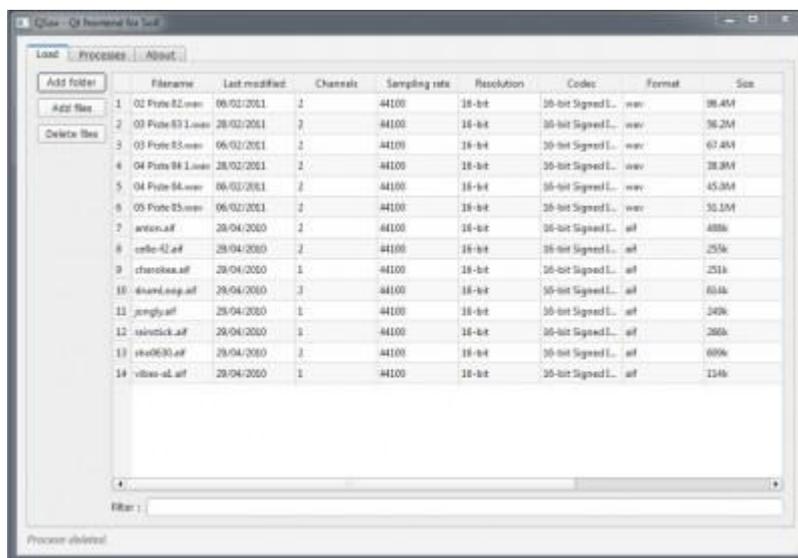
## 2. 2. Fonctionnement de l'interface graphique

L'interface graphique est composée de 2 onglets principaux :

- l'onglet "Load", qui permet de choisir quels fichiers on veut traiter
- l'onglet "Processes", qui permet de choisir les traitements, de les écouter et de les appliquer.

### 2. 2. 1. Onglet Load

Figure 1



Capture d'écran de l'onglet Load

Il s'agit essentiellement d'un tableau dans lequel se trouve la liste des fichiers audio à traiter. Soit on importe l'ensemble des fichiers audio d'un répertoire, soit on importe des fichiers un par un, et ces opérations peuvent être répétées plusieurs fois (exemple : si on veut charger tous les sons du répertoire A, tous les sons du répertoire B et quelques sons du répertoire C).

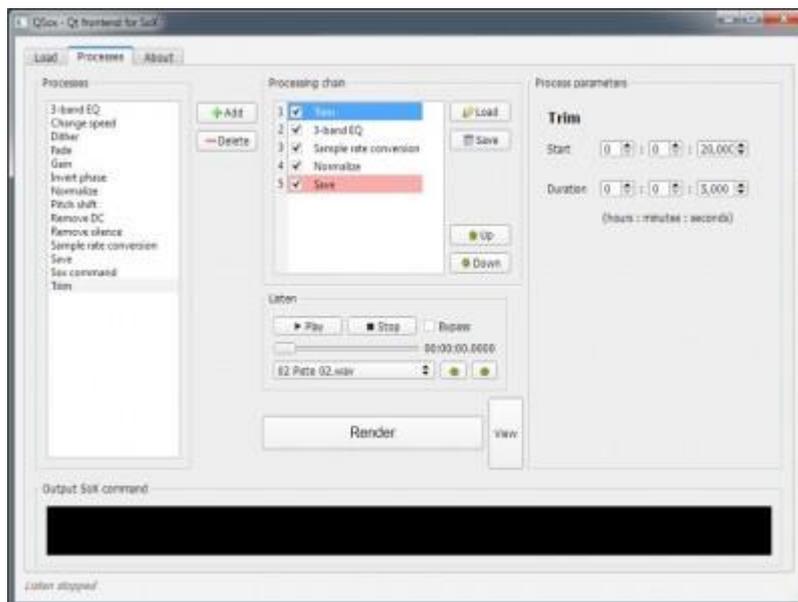
Inversement, les fichiers déjà chargés peuvent être sélectionnés afin de les supprimer du projet.

Ce tableau permet également d'afficher des informations usuelles sur les fichiers audio : nom, date de dernière modification, nombre de canaux, fréquence d'échantillonnage, résolution, codec, format, taille occupée sur le disque, durée, adresse où se trouve le fichier sur le disque. On peut classer les fichiers selon n'importe lequel de ces critères, ce qui est particulièrement utile quand on utilise beaucoup de fichiers.

Une fois que les fichiers ont été choisis, on peut paramétrer les traitements dans l'onglet "Processes", mais il est toujours possible de revenir à l'onglet "Load" en cours de travail.

## 2. 2. 2. Onglet Processes

Figure 1



Capture d'écran de l'onglet Processes

Cet onglet est divisé en 5 blocs :

- le bloc "Processes" correspond à la liste des traitements utilisables et permet de sélectionner un traitement pour l'ajouter à la chaîne de traitements (avec la touche "Add")
- le bloc "Processing chain" correspond à la chaîne de traitements ; on peut utiliser autant de traitements qu'on veut et chaque traitement peut être utilisé plusieurs fois (y compris avec des paramètres différents pour chaque instance) ; ce bloc permet aussi de sélectionner chaque traitement de la chaîne pour pouvoir l'éditer, le déplacer dans la chaîne (touches "Up" et "Down"), le supprimer de la chaîne (touche "Delete"), l'activer ou le désactiver ; enfin, il est possible de sauvegarder ou de charger une chaîne de traitement pour pouvoir l'utiliser plus tard avec d'autres fichiers audio
- le bloc "Process parameters" est une petite interface graphique permettant de régler les paramètres du traitement sélectionné dans "Processing chain" (elle est donc différente d'un traitement à l'autre) ; les paramètres sont également sauvegardés quand on sauvegarde la chaîne
- le bloc "Listen" permet d'obtenir un rendu audio en temps réel de la chaîne de traitement ; on peut choisir le fichier à lire, naviguer dans ce fichier et désactiver toute la chaîne de traitement
- le bloc "Output SoX command" affiche les lignes de commande envoyées à SoX (rappelons que SoX est utilisé comme moteur audio dans QSox).

Ainsi, chaque traitement correspond à un module graphique qui permet de modifier ses paramètres ; ces

modules peuvent être chaînés dans l'ordre que l'on veut et on peut les utiliser autant de fois qu'il est nécessaire dans la chaîne. 12 modules de traitement courants sont actuellement inclus dans l'interface graphique de QSox, ainsi qu'un module de sauvegarde et un module spécial permettant d'entrer sous forme de ligne de commande n'importe quelle fonctionnalité de SoX.

Voici la liste des traitements :

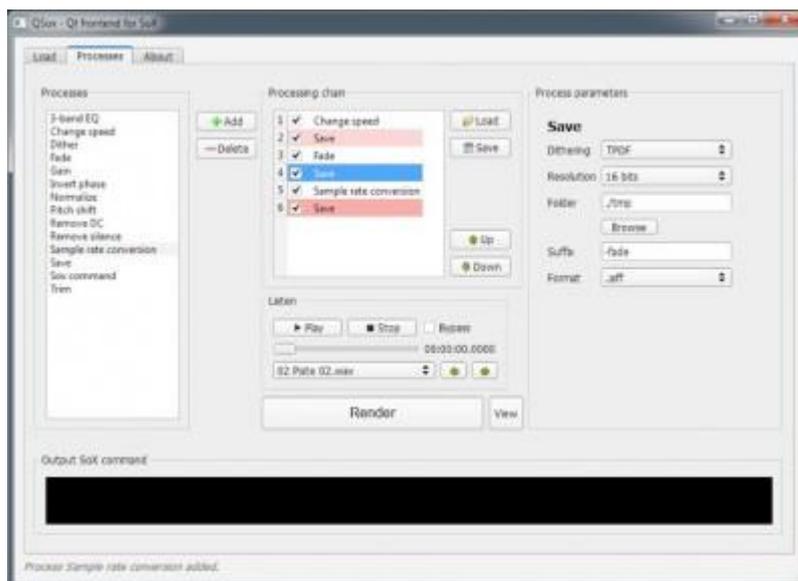
- 3-band EQ: égaliseur paramétrique 3 bandes (low shelving, peak/notch et high shelving)
- Change speed : permet de lire le fichier audio plus vite ou plus lentement
- Dither : permet de masquer les distorsions provenant d'erreurs de troncature lors d'une réduction de résolution numérique
- Fade : fondu sur le niveau sonore en début ou en fin de fichier audio (5 courbes différentes)
- Gain : modification du niveau sonore (en dB ou facteur linéaire)
- Invert phase : multiplication des valeurs des échantillons par -1
- Normalize : changement du niveau afin que le maximum atteigne une valeur fournie en paramètre par l'utilisateur
- Pitch shift : changement de la hauteur sans modifier la durée
- Remove DC : suppression des signaux continus
- Remove silence : permet de détecter et de supprimer les silences
- Sample rate conversion : changement de fréquence d'échantillonnage (rééchantillonnage)
- Save : sauvegarde avec choix du codec, du format et de leurs paramètres
- SoX command : permet d'entrer n'importe quelle ligne de commande valide de SoX
- Trim : permet de supprimer le début ou la fin du fichier audio

Deux traitements ont un statut particulier : "SoX command" et surtout "Save".

Le traitement "SoX command" permet d'entrer directement n'importe quelle ligne de commande valide de SoX dans une zone de texte, ainsi on dispose de tous les traitements de SoX sous forme de ligne de commande. Ceci est utile si on veut utiliser un traitement qui n'existe pas encore sous forme graphique, ou si on veut modifier un paramètre non inclus dans l'interface graphique. Ce traitement fonctionne de la même manière que les autres au sein de la chaîne, en particulier on peut l'utiliser plusieurs fois (y compris avec des lignes de commande différentes pour chaque instance).

Le traitement "Save" est un dispositif permettant de configurer les fichiers de sortie produits à partir des fichiers choisis dans l'onglet "Load". Il comprend une opération de *dithering* <sup>(5)</sup> (qui existe aussi en séparé dans les traitements), et permet de choisir la résolution, le codec, le format et le chemin des fichiers de sortie. On peut aussi ajouter un suffixe au nom de chacun des fichiers de sortie.

Figure 4



Capture d'écran d'un exemple de sauvegardes multiples

Comme on peut le voir sur cette capture d'écran, il est possible d'utiliser plusieurs traitements "Save" dans la chaîne, placés à des étapes de traitement différentes. Chaque "Save" donne lieu à la création d'un fichier qui correspond à l'étape de la chaîne où il se trouve. Ainsi, on peut obtenir des sauvegardes intermédiaires pour chacun des fichiers à traiter. Chacune de ces étapes de sauvegarde peut être associée à un suffixe spécifique pour le nommage des fichiers. Par défaut, il n'y a qu'un seul traitement "Save", situé en fin de chaîne, et il ne peut pas être supprimé.

Une fois qu'on a sélectionné les fichiers à traiter, qu'on a choisi les traitements et réglé leurs paramètres, et qu'on a configuré les fichiers de sauvegarde avec les traitements "Save", la touche "Render" lance le traitement par lots sur tous les fichiers (l'un après l'autre).

Enfin, la touche "View" sert à ouvrir le répertoire où ont été enregistrés les fichiers de sortie (correspond à "Save" situé en fin de chaîne dans le cas où il y a des "Save" intermédiaires).

## 3. Généalogie de QSox

QSox a été développé dans le cadre du projet HD3D-IIO <sup>(6)</sup>. En tant que partenaire académique de ce projet, le CICM <sup>(7)</sup> a travaillé sur divers aspects liés aux usages dans les métiers du son dans l'audiovisuel et notamment sur la conception puis le développement d'un éditeur audio adapté au métier de *servicing* <sup>(8)</sup> de film.

### 3. 1. Le premier éditeur audio de HD3D

Un premier prototype d'éditeur audio a été créé à l'aide du logiciel Max/MSP 4.6 <sup>(9)</sup>. Ce prototype était fonctionnel pour l'essentiel mais non utilisable sous Linux, ni déployable facilement. L'interface était organisée de la même manière que les tâches successives correspondant à la partie audio du métier de *servicing* de film.

Lors de la réalisation d'un film, le *servicing* comprend toutes les étapes de travail situées chronologiquement entre le moment où le film est masterisé et le moment où le film est diffusé. Cela comprend donc la fabrication des versions en langue étrangère, l'export des masters vers divers formats ou supports (fichier HD, fichier pour streaming, DVD, Blu-Ray, VHS, etc.) et la vérification des erreurs.

Dans le premier éditeur audio de HD3D, nous avons notamment couvert les étapes suivantes :

- dispositif pour assembler des bandes son fournies sous la forme de deux fichiers séparés (début et fin), généralement numérisés à partir de bandes analogiques
- entrelaçage et désentrelaçage des canaux d'un fichier audio multicanal ; changement de l'ordre des canaux au cas où ils ne soient pas dans l'ordre standard
- possibilité de remplacer un canal par un autre (utile pour les versions en langue étrangère notamment)
- affichage des informations sur un fichier audio (utile pour vérifier si les paramètres sont corrects), et visualisation des formes d'onde ainsi que des niveaux sonores
- correction des problèmes de niveau sonore
- correction des problèmes de durée
- *upmix, downmix*
- changement de codec ou de format
- affichage des temps universellement adaptable.

## 3. 2. Hsox

Une version libre et multiplateforme de cet éditeur a été ensuite développée en utilisant l'environnement de développement Qt. Comme SoX a été choisi comme moteur de traitements audio numériques, cette version a été nommée Hsox (« SoX pour HD3D »).

## 3. 3. Vers une version plus générique

Dans le but d'étendre la communauté des utilisateurs potentiels au-delà du domaine du *servicing* de film, nous avons décidé d'étendre les fonctionnalités de Hsox aux opérations audio numériques utilitaires les plus courantes, tout en conservant l'architecture générale de Hsox, mais en repensant son interface graphique.

Cette version générique a été appelée QSox car elle est toujours basée sur SoX et Qt.

# 4. QSox : architecture et développement

## 4. 1. Architecture

Dans le contexte du logiciel libre, il est dommage de redévelopper des traitements génériques existant déjà. C'est pourquoi nous avons choisi de ne pas développer notre propre moteur de traitements audio numériques. Si nous avons eu besoin de traitements audio numériques spécifiques ou nouveaux, au contraire il aurait fallu développer notre propre moteur de traitements (ou intégrer ces traitements à un moteur existant).

Ainsi, QSox est principalement une interface graphique développée en C++ <sup>(10)</sup> avec l'environnement de développement Qt. Cette interface génère des lignes de commande envoyées au logiciel SoX, qui est utilisé comme moteur de traitements audio numériques. À ce titre, on peut dire que QSox est une *front end* <sup>(11)</sup> pour SoX.

Il est important de noter que QSox n'est pas une interface graphique exhaustive, universelle et multi-paradigme <sup>(12)</sup> pour SoX, mais une interface particulière tournée surtout vers le paradigme du traitement par lots.

## 4. 2. Le choix de SoX

Habituellement décrit comme le « couteau suisse de l'audio numérique », SoX est un logiciel en ligne de commande multiplateforme (Windows, Linux, Mac OS X), libre et gratuit, qui peut convertir des fichiers audio numériques d'un codec ou d'un format à un autre. Par ailleurs, il permet d'appliquer divers traitements sur ces fichiers. Enfin, il peut être utilisé pour lire ou enregistrer des fichiers audio en temps réel.

Il est possible de grouper un ensemble de lignes de commande de SoX sous la forme d'un script sh (sous Linux et OSX) ou batch (sous Windows). Ainsi, SoX permet le traitement par lots.

Dans SoX, les lignes de commande peuvent contenir jusqu'à 5 types d'informations :

- options globales, correspondant à des options de configuration de SoX
- choix des fichiers d'entrée et de sortie, ce qui permet aussi de déterminer les éventuelles conversions de format ; si on ouvre plusieurs fichiers, on peut les mixer, les concaténer, les placer sur des canaux différents, etc.
- options de format, permettant de préciser certaines opérations au moment d'ouvrir ou de sauvegarder les fichiers (rééchantillonnage, etc.)
- effets (il s'agit des traitements audio numériques) et options des effets (paramètres de chaque effet) ; on peut en utiliser autant qu'on veut afin de créer des chaînes d'effets.

Dans le cadre de notre projet, les avantages de SoX sont les suivants :

- SoX existe depuis longtemps, d'où une stabilité de fonctionnement idéale, et son développement continue toujours
- c'est un logiciel libre, gratuit et multiplateforme
- il possède de très nombreux modes de fonctionnement grâce au système des options globales et des options de format
- la qualité des traitements de SoX est reconnue ; en ce qui concerne le rééchantillonnage, SoX est souvent cité parmi les références en termes de fidélité <sup>(13)</sup>
- la plupart des traitements de SoX sont des traitements génériques
- il y a un grand nombre de traitements disponibles
- les traitements possèdent pour la plupart un grand nombre de paramètres, qui sont remplacés par des valeurs par défaut s'ils ne sont pas explicitement précisés
- il est possible d'utiliser des scripts, ce qui est utile pour les traitements par lots, ou pour effectuer des traitements complexes utilisant plusieurs effets, options globales et options de format
- SoX comprend une documentation complète et détaillée.

Notons que SoX existe aussi sous la forme d'une bibliothèque appelée libsox. Cette bibliothèque permet d'intégrer directement les possibilités de SoX au code source d'un autre logiciel. Cependant, nous avons préféré une architecture séparant SoX de l'interface graphique pour les raisons suivantes :

- le développement est plus rapide et plus facile que celui d'un logiciel *standalone* utilisant la bibliothèque libsox
- les cycles de développement de QSox sont indépendants de ceux de SoX ; chaque version de QSox est fournie avec la dernière version de SoX avec laquelle elle a été testée.

## 4. 3. Avantages de l'interface graphique pour le traitement par lots

SoX est peu connu dans la communauté des utilisateurs de l'informatique musicale, car beaucoup de membres de cette communauté sont rebutés par les interfaces en ligne de commande. En informatique musicale actuelle, seules les interfaces de commande graphiques et les interfaces de commande physiques se sont imposées pour le grand public et dans l'industrie.

Il est certes possible de faire du traitement par lots avec SoX en utilisant des scripts de lignes de commande. Cependant, la syntaxe n'est pas aisée pour un utilisateur final habitué aux interfaces graphiques. De plus, la rédaction du script peut prendre beaucoup de temps.

D'où l'idée d'intégrer directement le concept de traitement par lots à l'interface graphique de l'éditeur audio, afin de gagner en simplicité et en rapidité d'utilisation.

## 4. 4. Le choix d'un langage objet

Les langages objet facilitent des développements informatiques modulaires et très structurés. Dans le contexte du logiciel libre, il est important de programmer de manière suffisamment claire et structurée pour que d'autres développeurs puissent ajouter facilement de nouvelles fonctions sans avoir le sentiment qu'il faudrait idéalement réécrire ce qui a déjà été fait au préalable. Par exemple, chaque module graphique associé à un traitement de SoX correspond à une classe et chaque classe de ce type se trouve dans un fichier séparé. Ainsi, de nouveaux traitements de SoX peuvent être facilement ajoutés à QSox dans des versions futures.

## 4. 5. Le choix de Qt

Qt est un environnement de développement complet adapté au développement de logiciels multiplateformes avec interface graphique. Cet environnement est basé sur le langage C++, qui est un langage objet performant, universel et standard.

Les avantages de Qt sont les suivants :

- couche d'abstraction au-dessus du C++, ce qui simplifie son utilisation et donne une méthode de programmation
- fourni avec des classes génériques couvrant tous les besoins du développement logiciel
- fourni avec un IDE très complet qui comprend un éditeur de code, une interface graphique permettant d'organiser les projets, un éditeur d'interfaces graphiques, un compilateur, un debugger, un outil de traduction, un gestionnaire d'aide très pratique, etc.
- permet d'avoir la même méthode de développement pour toutes les plateformes.
- en mode release, Qt est une bibliothèque légère qu'il est facile de télécharger en même temps que le programme et qui ne demande pas d'installation

- Qt est un environnement connu et éprouvé qui dispose d'une très grande communauté d'utilisateurs

L'environnement Java <sup>(14)</sup> est le principal concurrent de Qt dans un contexte multiplateforme, mais nous ne l'avons pas retenu car il faut le télécharger et l'installer au préalable sur les machines avec lesquelles on veut exécuter les programmes. Comme il s'agit d'un environnement assez lourd, cela ne nous a pas semblé compatible avec un logiciel qu'on doit pouvoir télécharger dans l'urgence et utiliser immédiatement. Au contraire, la bibliothèque correspondant à Qt est suffisamment légère pour pouvoir être téléchargée en même temps que QSox, et elle ne nécessite pas d'installation.

Par contre, le fait que le C++ soit un langage plus performant et plus universel que le Java n'était pas une contrainte déterminante pour le développement de notre interface graphique.

On peut également citer Mono <sup>(15)</sup> comme alternative ; cependant, cet environnement n'était pas encore assez élaboré au début du développement de QSox.

Enfin, il existe bien sûr des bibliothèques permettant de développer des interfaces graphiques multiplateformes, par exemple Gtk++ <sup>(16)</sup>, mais elles ne sont pas intégrées à un environnement complet qui simplifie l'organisation du développement de manière considérable.

## 5. Perspectives

En tant que logiciel libre, QSox est amené à évoluer à court terme et à long terme. Aussi, nous invitons l'ensemble de la communauté des utilisateurs et développeurs en informatique musicale à télécharger QSox et à nous faire part de leurs retours.

Nous faisons ci-dessous quelques propositions d'évolutions pour Qsox.

### 5. 1. Première version

QSox est actuellement en cours de développement, comme l'indique son numéro de version <sup>(17)</sup>.

Quelques détails doivent être réglés afin d'obtenir la version 1.0 :

- vérification des erreurs de code et de conception C++
- compilation possible d'un code source strictement identique sur n'importe quelle plateforme (pour l'instant il y a quelques lignes de code différentes entre les versions Linux, Windows et Mac OS X)
- détails de finition de l'interface graphique
- ajout d'une barre de progression pendant le traitement par lots, avec la possibilité d'annuler
- gestion des raccourcis clavier
- gestion des erreurs pouvant arriver pendant l'exécution
- gestion du processus d'installation
- tests fonctionnels complets sur plusieurs plateformes
- ajout de tests unitaires dans le code
- documentation pour l'utilisateur.

### 5. 2. Nouveaux modules de traitement

Pour l'instant, seuls 12 traitements sont supportés dans l'interface graphique. Nous avons choisi les traitements les plus courants et les plus génériques. Il est souhaitable d'ajouter d'autres traitements peu à peu, sur le modèle des traitements déjà existants. On peut même imaginer plusieurs modules graphiques différents pour un même traitement. Par exemple, il n'est parfois pas utile de pouvoir contrôler tous les paramètres d'un traitement et dans ce cas, le fait de tous les représenter surcharge l'interface graphique pour rien ; inversement, il arrive d'avoir besoin de modifier des paramètres inhabituels. Ainsi, pour un même traitement, on peut envisager deux modules graphiques différents (bloc "Process parameters" de l'onglet "Processes" ? voir 2. 2. 2.) : une interface graphique simple et une interface graphique avancée.

On peut distinguer 2 types de nouveaux modules de traitement :

- les modules de traitement correspondant à un traitement qui existe directement dans SoX sous forme d'effet
- les modules de traitement plus complexes, constitués d'une combinaison de plusieurs effets, options de format et options globales, par exemple des opérations sur les canaux : downmix, spatialisation, upmix, changement de l'ordre des canaux, etc.

La première solution est d'ajouter les nouveaux traitements sur le modèle de ceux qui ont déjà été implémentés. L'avantage de cette solution est qu'un développeur peut facilement ajouter un nouveau traitement. En effet, l'architecture du logiciel a été conçue de manière à faciliter l'ajout de nouveaux modules de traitement. Cependant, un utilisateur ne connaissant pas le développement informatique ne pourra pas ajouter de nouveau module de traitement.

Une autre approche de ce problème consiste à donner la possibilité aux utilisateurs de créer eux-mêmes des traitements de leur choix, ainsi qu'une interface graphique permettant de les contrôler. Pour cela, l'utilisateur pourrait associer une ligne de commande (ou un script) de SoX à un module graphique particulier. Dans ce cas, il faudrait développer un mini-éditeur graphique permettant :

- la création d'un module graphique "Process parameters" à partir de contrôles simples (boutons, potentiomètres, listes déroulantes, etc.) placés librement par l'utilisateur
- l'association des propriétés des contrôles du module graphique à des paramètres de la ligne de commande (ou du script) de SoX choisie par l'utilisateur.

Complexe à mettre en place, ce système peut cependant réduire considérablement la charge de développement informatique par la suite. En effet, une fois que le système est développé, ce n'est plus la peine de développer de nouvelles interfaces graphiques spécifiques pour chaque nouveau traitement : les nouveaux traitements et leurs interfaces graphiques peuvent être ajoutés directement par des utilisateurs depuis QSox, et ces utilisateurs peuvent partager leurs traitements sur Internet.

## 5. 3. Renommage avancé

Dans la version actuelle, on peut ajouter un même suffixe au nom de chacun des fichiers traités. Si on utilise plusieurs traitements "Save", on peut bien sûr utiliser des suffixes particuliers pour chaque "Save".

Cependant, il semble utile de développer un système de renommage plus complet, tel qu'on le rencontre dans les logiciels usuels de renommage. Une autre solution est de faciliter la communication avec un logiciel de renommage multiplateforme de référence.

## 5. 4. Forme d'onde et visualisation du niveau sonore

L'affichage de la forme d'onde aide à se repérer dans le fichier pendant la lecture et la navigation, et permet de sélectionner une portion du fichier, ce qui est utile pour de très nombreux traitements (par

exemple : « ne garder que la zone sélectionnée et effacer le reste du fichier ») ou pour lire une portion du fichier en boucle.

De plus, une forme d'onde peut donner des informations importantes sur un fichier audio.

Quand à la visualisation du niveau sonore (VU-mètres, PPM, etc.), il s'agit d'un outil très répandu dans le domaine du son et qui peut fournir des informations très utiles.

Pour développer ces nouvelles fonctionnalités, la bibliothèque Qwt <sup>(18)</sup>, une extension de Qt qui comprend des contrôles adaptés aux applications scientifiques, semble particulièrement adaptée.

\*

QSox est un projet auquel ont participé des stagiaires, doctorants et chercheurs du CICM :  
Jérôme Abel ? responsable du développement ; conception  
Julien Bréval ? responsable de la conception ; lignes de commande de SoX pour QSox  
Julien Colafrancesco ? développement  
Benoît Courribet ? participation à la conception, expérience utilisateur  
Guilherme Carvalho ? participation à la conception  
Anne Sedès ? coordination du projet au sein du CICM et avec HD3D ; validation des choix de conception et de développement

- 
1. Un traitement par lots est un enchaînement automatique de commandes sur un ordinateur, sans intervention d'un opérateur.
  2. WaveLab, Soundblade, SoX, Snd, Ecasound, Sound Forge, Peak, Audacity, Soundhack, Wavosaur, etc.
  3. SoX(<http://sox.sourceforge.net/>) est notamment un éditeur audio fonctionnant en ligne de commande.
  4. <http://qt.nokia.com/>
  5. Le dithering est un procédé visant à réduire le bruit de quantification induit par une baisse de la résolution d'un signal audionumérique. Pour plus de détails, on peut se référer au document intitulé « Le dithering » disponible à l'adresse <http://pfast.fr/?HD3D>
  6. HD3D-IIO (Haute Définition Trois Dimensions-Initiative pour une Industrie ouverte) a été un projet structurant du pôle de compétitivité Cap Digital, qui s'est tenu entre 2007 et 2010, réunissant académiques et entreprises du secteur audiovisuel d'Île-de-France pour développer des outils de gestion de production adaptés à la création de contenus audiovisuels numériques. Le suffixe IIO (Initiative pour une Industrie ouverte) se rapportait à l'aspect libre de ces développements et à la volonté de partager les outils logiciels entre partenaires, voire concurrents d'une même branche.
  7. CICM, Centre de recherche en Informatique et Création musicale, université Paris 8, MSH Paris Nord.
  8. Il s'agit de la branche qui exploite la distribution des catalogues de contenus audiovisuels, tout en assurant la compatibilité des essences avec les supports et formats de diffusion en perpétuelle évolution.
  9. <http://www.cycling74.com>
  10. <http://www.cplusplus.com>

11. Par exemple, RazorLame (<http://www.dors.de/razorlame/>) est une *front end* pour LAME.
12. Par exemple, le traitement en temps réel, l'édition graphique d'un unique fichier et le traitement par lots sont autant de paradigmes pour le domaine des traitements audio numériques.
13. On peut citer par exemple le comparatif des algorithmes de conversion de fréquence d'échantillonnage (*Sample Rate Conversion* ou SRC) <http://src.infinetwave.ca/>
14. Nous parlons ici du langage Java, des bibliothèques associées, de l'environnement d'exécution (dit « machine virtuelle ») et de l'environnement de développement <http://www.java.com>
15. Il s'agit du portage multiplateforme de l'environnement .NET de Microsoft, ouvert à tous les langages (pas limité au C++ ou au Java) <http://www.mono-project.net>
16. <http://sourceforge.net/projects/gtkpp/>
17. Généralement, on considère que les versions 0.x correspondent aux versions non terminées d'un logiciel ; la version 1.0 est la première qui soit complète.
18. <http://qwt.sourceforge.net>

---

**Pour citer ce document:**

Julien Bréval, « QSox : un éditeur audio libre, multiplateforme et adapté au traitement par lots » *RFIM* [En ligne], Numéros, n° 1 - automne 2011, Mis à jour le 12/10/2011

URL: <http://revues.mshparisnord.org/rfim/index.php?id=145>

Cet article est mis à disposition sous [contrat Creative Commons](#)