

Numéros / n° 2 - automne 2012

« Diffuser, distribuer, bloguer pour préserver : le cas de SuperCollider »

Jean-Pierre Quinquenel

Résumé

À travers l'étude d'un logiciel important dans la création musicale contemporaine : SuperCollider, open source et gratuit, nous étudierons les nouveaux enjeux et stratégies mises au point par les développeurs, musiciens ou artistes qui privilégient Internet comme lieu et moyen de préservation de leurs créations. Nous baserons notre étude sur l'expérience liée à notre navigation personnelle au travers des sites en relation avec ce programme dans la recherche des éléments suivants : le patch, le tutoriel et le fichier audio. Nous évoquons pour finir l'intérêt de la théorie des communautés de pratique pour traiter ce genre d'étude.

Introduction

L'oeuvre musicale une fois pensée puis réalisée et interprétée, entre dans une nouvelle phase de son existence, celle de la diffusion/préservation. Les oeuvres recourant aux technologies audionumériques sont confrontées à une série de nouvelles difficultés dont les portées économiques, esthétiques ou même pratiques sont considérables et probablement encore mal perçues. La plus importante qui nous préoccupe dans cet article est la pérennisation non seulement du concept (le patch au sens générique du terme, c'est-à-dire le document, vierge au départ dans lequel sont inscrites toutes les données nécessaires à l'interprétation de l'oeuvre) mais aussi de l'environnement technologique le rendant possible. Celui-ci conditionne la re-production d'une oeuvre dans d'autres contextes, soit sur scène, soit dans l'ordinateur d'un éventuel utilisateur qui l'aura téléchargée sous forme de patch sur un site dédié. À travers l'étude d'un logiciel important dans l'univers de la création musicale contemporaine, SuperCollider, programme open source et gratuit, nous étudierons les nouveaux enjeux et nouvelles stratégies mises au point par la communauté des utilisateurs et développeurs de ce logiciel emblématique de l'informatique musicale qui privilégie Internet comme lieu et moyen de préservation de la création.

Nous baserons notre étude sur l'expérience liée à notre navigation personnelle au travers des sites en relation avec ce programme dans la recherche des éléments suivants : le patch (créé par quelqu'un d'autre et modifiable), le tutoriel (indispensable pour la bonne compréhension du logiciel) et le fichier audio (qui fait entendre le résultat en ligne avec une option de téléchargement).

Les idées exposées ci-dessous sont le fruit de nos observations personnelles. Par commodité, nous adopterons au préalable l'abréviation SC tout au long de notre article pour désigner le logiciel SuperCollider, Pd pour Pure data. Enfin nous appellerons patch le fichier-document dans lequel sont inscrites les lignes de code d'une oeuvre conçue dans cet environnement spécifique.

1. Présentation du logiciel

1. 1. Qu'est-ce que Super Collider

SuperCollider est un logiciel créé par James McCartney au début des années 1990. On lira sa courte et savoureuse préface du *SuperCollider Book* quant aux années de conception de son programme : deux programmes Synt-O-Matic (1990) et Pyrite (1994) précèdent SC dont la première publication scientifique sur SC date de 1996 consultable ici : <http://www.audiosynth.com/icmc96paper.html>

SC est décrit ainsi dans Wikipédia : « SuperCollider est un environnement et un langage de programmation pour la synthèse audio en temps réel et la composition algorithmique ». McCartney fit le choix de développer son logiciel sous un système d'exploitation libre et open source de type Licence publique générale GNU (GPL GNU). Celui-ci a vite gagné en notoriété internationale tant par la puissance et la qualité du rendu sonore que grâce aux opportunités de communication à l'échelle planétaire permise par Internet.

Ainsi, à l'instar de Pure data ou de Csound, SuperCollider représente un des logiciels de création musicale les plus en pointe, étudié dans de plus en plus d'universités et utilisé par beaucoup de musiciens venant d'horizons très différents. Il est désormais une référence incontournable avec de véritables « fans ». Il est à remarquer que de plus en plus de développeurs/programmeurs proposent des alternatives au langage « Slang » natif du à la séparation entre la partie « cliente » (dans laquelle on écrit le programme) et la partie serveur audio (qui crée le son), les deux communiquant par OSC ce qui reste une originalité propre à ce logiciel. Cet aspect dont nous reparlons plus bas est essentiel et ouvre littéralement SuperCollider vers l'extérieur, vers de nouvelles initiatives en matière notamment de *live coding* ⁽¹⁾.

« Composer » et créer avec SC est emblématique de ces logiciels, open source pour la plupart, demandant au musicien « traditionnel » une forme de reconversion à la fois intellectuelle et esthétique : il faut programmer son projet par l'écriture parfois complexe de lignes de code ⁽²⁾. Les flux sonores ou musicaux qui en résultent sont d'un rendulaissant rarement indifférent. SC permet d'explorer des matières, des textures sonores de façon pointue et très innovante sur le plan musical. Quant au tutoriel glané sur Internet (en dehors du workshop), il reste la source d'information quasi-unique de ce nouvel utilisateur lambda cherchant à « jouer » rapidement de son nouveau logiciel, dont il a pu apprécier quelques réalisations en écoutant par exemple en ligne des extraits musicaux. La quête du patch vient donc de façon concomitante ou plutôt de notre point de vue, ce sont trois moments (le patch, le tutoriel et le fichier audio) de la « pratique » du logiciel qui fonctionnent ensemble. De fait, la communication au sens large du terme pour tout logiciel en fin de compte est de se faire connaître afin d'attirer de nouveaux utilisateurs en proposant ateliers, tutoriels ⁽³⁾, extraits de concerts et d'œuvres (pour les débutants par exemple).

Notre schéma ci-dessous tente de montrer que cette communauté d'utilisateurs de SC se développe, s'informe et échange des contenus (patches et tutoriels pour l'essentiel, cf. le forum Codelab) grâce et par une circulation entre sites considérés comme site SC de référence (la page d'accueil SC sur SourceForge et quelques autres sites) et d'autres sites plus en périphérie (les réseaux sociaux, les blogs personnels) permettant de partager des patches, des projets ou des idées. D'une manière générale, ces modes de circulation sur le Web ne sont pas réservés aux logiciels libres ou open source. Ils nous semblent désormais être le fait d'une grande partie de l'informatique musicale. L'« ancienneté » sur le marché ne joue aucun rôle : que ce soit pour Cubase, Csound ou Max, les utilisateurs (ou clients pour les formats propriétaires) « collaborent » et interagissent de manière identique au sein d'une même communauté de pratique.

Ce partage de patches, de tutoriels et de conseils entre artistes constitue une certaine garantie dans le temps pour ces contenus puisqu'ils seront stockés dans des disques durs ou des bases de données. Mais pour combien de temps ? Le stockage d'informations dans des disques durs ou en nuage ne garantit pas une pérennité, ni une visibilité sur le net à moyen ou long terme puisque de nouveaux contenus numériques (patches, fichiers audio ou documents) se voient relégués très vite au second rang par l'arrivée de nouveaux contenus. La problématique de fond, on l'aura compris, est de (per)-durer par l'utilisateur qui reste le seul garant d'une forme de continuité (celle de son créateur). Ainsi, SC adopte, à l'initiative à la fois des concepteurs/développeurs du logiciel (autour de James McCartney) ou de ses « fans », une arborescence internet complexe mais claire tout à fait à même de préserver concepts et réalisations qui découlent d'une création particulière.

1. 2. Ses spécificités

À l'ouverture du programme et d'un nouveau document vierge appelé patch, à l'identique de Pd par exemple, on accède à une simple fenêtre vide à l'écran dans laquelle on *écrit* avec des caractères alphanumériques (en majuscule ou en minuscule) des lignes de codes qui doivent respecter une syntaxe avec ses règles propres nécessitant un apprentissage spécifique. Comme nous l'avons écrit plus haut, le nouvel utilisateur doit aller chercher sur Internet le ou les tutoriels afin de s'initier.

Le langage propre à SC, le « slang » (raccourci pour SuperCollider Langage) utilisé dans un patch communique au serveur par OSC garantissant ainsi une rapidité d'exécution sonore. Le format texte (par exemple, un format .rtf) dans lequel on inscrit le programme offre l'avantage de sauvegarder facilement la totalité d'une oeuvre dans un document aisément archivable et échangeable ⁽⁴⁾ sur le net par le biais des forums ou sites dédiés pendant une certaine période (mais combien de temps ?). Le langage de programmation spécifique au programme lui-même est-il une garantie (ou une opportunité supplémentaire) pour pérenniser des oeuvres suivant les versions du logiciel (patches plus ou moins complexes ou Sctweets en 140 caractères) ? Sur ce point précis, notre propre constat est le suivant : certains patches téléchargés fonctionnent et d'autres plus... ou pas ! Paradoxalement, nous avons pu « évaluer » certains patches relativement récents (des SCTweets de 2011-2012 pour l'essentiel) dans une version beaucoup plus ancienne (0.1(1.0) de 2001) de SC sans aucun problème.

Miller Puckette (Puckette, 2001) explique l'intérêt du format textuel en relation avec cette préoccupation sous-jacente à l'informatique musicale de sa propre pérennité par la sauvegarde systématique de toutes les données ⁽⁵⁾. Puckette met en avant qu'un fichier texte, même contenant toutes les métadonnées, est négligeable du point de vue de la mémoire informatique. Le langage informatique utilisé reste facilement compréhensible (un langage identifié C, C++ etc.) pour de futures générations qui auront l'avantage, continue-t-il, de bénéficier des progrès de ces technologies audio-numériques du point de vue qualitatif (des machines plus puissantes, plus stables avec de meilleurs rendus sonores). Les avantages en question sont évidemment similaires pour SC mais sont-ils vraiment synonyme de pérennisation (comme l'est une partition finalement qui survit à plusieurs générations de musiciens) ? Et quelle est la perspective temporelle sous-entendue ? Une décennie ? Plusieurs décennies ?

D'autres questions importantes qui se posent sont liées en particulier au logiciel lui-même, à sa propre survie ou à sa « migration » technique ou technologique ⁽⁶⁾. L'exemple de Csound est intéressant puisque certains anciens patches ont été « ré-écrits » pour fonctionner correctement sur QuteCsound (dans notre cas la version 0.6.1) : « Stria » de John Chowning par exemple a été adapté conjointement (« with modifications » est-il écrit dans les commentaires) par Kevin Dahan en collaboration avec le compositeur. Mais qu'en est-il d'autres patches plus anciens ?

Ainsi, SC permet de sauvegarder sur un disque dur l'intégralité de son travail sous un format texte quelconque avec la possibilité (par un clic droit) de le réouvrir ensuite directement dans SC dans son format spécifique .sc ; ou par un simple copier-coller directement dans un nouveau patch. L'évaluation des lignes de code se fait manuellement sur le clavier ou dans la barre de menus afin de faire entendre les flux sonores.

L'édition en temps-réel (en situation de *live coding* par exemple) (Collins, 2005) est possible grâce à une option graphique intéressante : celle d'une interface graphique (la GUI ou *Graphic User Interface*) permettant d'éditer certains paramètres choisis à l'avance (hauteur, filtres, granularité, tempi etc.). Les changements en temps-réel se font avec la souris, ou sur un contrôleur MIDI afin d'utiliser de vrais boutons et curseurs apparaissant à l'écran (également contenus dans les lignes de code). Une GUI ⁽⁷⁾ configurée pour une session de scène peut contenir alors dans une seule instance les lignes de code ainsi que des éléments tels qu'une table de mixage par exemple contrôlée par le biais de la souris. Le logiciel commercial Live d'Ableton que nous étudions dans notre thèse a finalement la même perspective conceptuelle : une instance unique (et plus intuitive) pour une vision globale du jeu sur scène. Cette dimension temps-réel ouvre SC vers de plus en plus d'opportunités de scénarios de musiques mixtes avec un soliste ou un ensemble instrumental et un ordinateur dialoguant de façon décalée ou non. Jouer et « évaluer » (informatiquement parlant) sont dans ce contexte très proches en termes de résultats bien que les processus puissent être distingués du point de vue du « faire ». Nous n'insisterons pas ici sur ces aspects esthétiques.

2. Nouveaux usages d'Internet

2. 1. Internet et l'hyperlien comme stratégie de

préservation et de diffusion

La « hiérarchie » web propre au logiciel SC, tel que nous l'explorons nous-même, s'organise autour de plusieurs sites principaux (dont quelques-uns spécifiquement en français) la plupart mentionnés dans la Webographie, qui permettent au débutant en quête de ressources pédagogiques ou musicales d'accéder rapidement à des tutoriels ou sites de diffusion en ligne. De manière plus générale, la première prise de contact peut se faire à priori par n'importe quel site listé dans un moteur de recherche mais à un moment précis, il va être renvoyé sur une « arborescence web » faisant référence qui le re-dirigera ensuite vers une seconde arborescence plus « spécialisée » (dépôt de codes, tutoriels etc.). Ainsi, les sites-référents sont à notre avis clairement identifiables : celui de James McCartney créateur de SC puis le site faisant office de vitrine officielle ⁽⁸⁾ cf. la Webographie. L'hyperlien ensuite laisse l'internaute naviguer librement au travers d'un nombre important de sites corrélés directement ou non, grâce auxquels il peut expérimenter par lui-même les tutoriels ou patchs proposés en libre téléchargement. Mais nous observons que certains sites, ceux mentionnés plus haut comme « référents » créant un premier niveau élevé d'informations (avec une deuxième arborescence, un niveau secondaire en relation avec le dépôt de codes par exemple) reviennent plus fréquemment parce qu'offrant plus de contenus pertinents (le débutant et l'utilisateur averti cherchent des informations officielles ou mises à jour). Le schéma récapitulatif plus bas tente de synthétiser, non exhaustivement, ces circulations complexes associées à l'utilisation de SC qui créent des jeux de miroirs favorisant de notre point de vue la pérennisation non seulement des oeuvres elles-mêmes (patchs personnels ou non), mais l'environnement complet (mises à jour logicielles, librairies supplémentaires, ressources pédagogiques réactualisées, créations musicales en ligne, etc.).

La complexification, l'extension progressive des technologies web ont élargi considérablement les opportunités tant du côté de l'offre que de celui de la demande en matière d'informatique musicale : la duplication ou la reproduction à l'infini des fichiers (textes, sons, images, programmes) par la simple action d'un clic de souris garantit à priori une forme de persistance dans l'espace-temps puisque dupliquant à volonté des copies du même patch avec les métadonnées qui lui sont corrélées. Nous revenons plus précisément sur ce point plus bas.

2. 2. Le forum, le blog, le microblogging et l'hyperlien : la Home-page et le cas Twitter

Les jeunes générations de musiciens utilisant SC comme outil créatif sollicitent particulièrement les réseaux sociaux comme modèle collaboratif dynamique et moyen de sauvegarde. En allant sur Facebook avec la page dédiée à ce logiciel, il est facile d'observer ce que nous décrivons plus loin comme moyen de sauvegarde dans ces modes de circulation de l'internaute. Les posts ou messages mis en ligne sur cette page renvoient d'une part à ces sites-référents de premier niveau, puis de fait à des sites autres (secondaires) offrant la possibilité d'écouter des créations originales, hébergées sur d'autres sites (Soundcloud, Youtube, Vimeo etc.). La navigation peut continuer en cliquant sur des hyperliens pour se diriger vers d'autres pages internet proposant colloques, ateliers etc., eux-mêmes probablement référencés dans cette première arborescence (1^{er} niveau) ; et ce, en boucle et à l'infini.

D'autre part, ces réseaux sociaux (Twitter plus spécifiquement en termes d'interactivité) sont un moyen rapide, simple pour contacter directement musiciens, artistes ou développeurs afin de solliciter leur aide ou conseil soit en répondant à un de leurs tweets ou par un mail privé directement à partir du réseau social. Twitter est un cas intéressant parce que son mode de fonctionnement contraignant (140 caractères maximum dans un post ou un message) est là-aussi mis à contribution pour communiquer de deux façons :

La première est évidemment l'hyperlien ; la deuxième est d'après nous nettement plus originale, puisque l'on peut poster des lignes de code SC dans la limite prescrite (cf. <http://supercollider.sourceforge.net/sc140/> qui recense certains de ces « tweets ») qui peuvent se présenter tels quels dans l'interface Twitter ⁽⁹⁾ dans un exemple de Nathaniel Virgo :

```
{LocalOut.ar(a =CombN.ar(BPF.ar(LocalIn.ar(2)*7.5+Saw.ar([32,33],0.2),  
2**LFNoise0.kr(4/3,4)*300,0.1).distort,2,2,40)) ;a}.play//
```

Il n'y a aucune possibilité de déroger à la règle sinon le moindre caractère manquant rendrait le code

inutilisable dans l'interface SC. Le profil associé à l'icône de son auteur (dans le tweet original) peut renvoyer vers un site personnel pouvant contenir des hyperliens du premier, deuxième niveau etc. pour reprendre la terminologie employée un peu plus haut. Le jeu de renvois est là aussi infini et renforce la communication faite autour du logiciel.

3. Synthèse, schéma récapitulatif et perspectives de recherches

Paul Griffiths dans son ouvrage célèbre (Griffiths, 1978) faisait remarquer que « l'activité artistique paraît toujours incohérente et multiforme aux contemporains » et qu'« Aujourd'hui, il n'existe pas de vérités communes », avec l'apriori que le présent musical de son époque ne pouvait être qu'incohérence et désordre masqué face à deux siècles fastueux qui précédaient. Au regard de ce que nous observons avec SC, ces « vérités communes » nous apparaissent mieux à l'observation des nouveaux usages de ces nouvelles figures de musiciens et pour multiformes qu'elles soient, ceci n'exclut en rien l'idée que le « multiforme » ou le « protéiforme » soient l'apanage de ces « nouvelles musiques ».

Comment synthétiser de façon globale ces « stratégies de survies » élaborées conjointement par les utilisateurs et concepteurs de SC ? Nous pensons qu'une partie de la réponse se trouve dans les modes de circulation à l'intérieur de cette arborescence internet, schématisés ci-dessous mettant en évidence ce jeu de renvois systématiques entre sites-référents/sites personnels sauvegardant à la fois le tutoriel, le patch, le fichier audio ainsi que d'autres contenus qui y sont rattachés. Ainsi, chaque nouvelle « circulation » ou déambulation est une opportunité supplémentaire facilitant cette pérennisation soit par l'archivage dans le disque dur soit par la (re)mise en ligne de ces informations.

On peut noter plus généralement que le logiciel libre bénéficie d'une plateforme de développement tout à fait unique : Sourceforge (on pourrait aussi citer Github : <https://github.com/>). D'autres existent mais Sourceforge demeure un site incontournable pour des projets collaboratifs pour lesquels développeurs, artistes et musiciens souhaitent partager et interagir tout au long de la « vie » numérique d'un logiciel (débugage, mises à jour etc.) : SuperCollider, Csound ou Pure data y trouvent une opportunité supplémentaire importante quant à cette stratégie de survie qui prévaut pour tous les logiciels. La visibilité sur le Net renforce évidemment très nettement cette capacité à perdurer face à un nombre important de concurrents, eux aussi libres ou open source.

Nous observons de manière plus générale que ces modes de relations/connexions entre sites-référents ne sont pas l'exclusivité de SuperCollider, ou même des logiciels libres : la plupart des logiciels de musique avec lesquels nous travaillons (logiciel de type séquenceur traditionnel ou libre comme Pure data, Csound et d'autres), fonctionnent maintenant de la même façon avec des variantes propres à chacun d'eux. Une remarque particulière concernant Max/Msp : ce logiciel n'est pas en reste et son intégration en 2009 dans Live avec Max For Live (que nous étudions dans notre thèse) montre que ce type de communication est devenu la règle. Un format dit « propriétaire » ou « libre » ne semble pas jouer de rôle déterminant face à des usages complexes liés à Internet qui dépassent ces spécificités. L'utilisateur participe d'un écosystème stable mais en constante évolution qui est source d'une certaine pérennité quant aux modes d'échanges des différents contenus (patches, fichiers sons et tutoriels) mais n'implique pas la certitude de son bon fonctionnement technique au moment de l'évaluation dans la fenêtre du patch. Internet crée-t-il une forme de confusion entre la capacité à stocker et celle de la garantie du résultat musical ? La disparition un jour de James McCartney, (au même titre que Miller Puckette pour Pure data) constitue-t-elle un risque possible pour son programme ? La web communauté est-elle le moyen idéal pour éviter celui-ci ?

De ce point de vue, il nous semble, *mutatis mutandis*, qu'il n'y a pas de spécificité SuperCollider de ce point de vue par rapport à d'autres logiciels (Pure data, Csound, ChucK etc.) : toutes ces stratégies web-communautaires sont un résultat croisé entre les pratiques complexes de l'utilisateur d'un logiciel lambda et les éditeurs ou développeurs de logiciels.

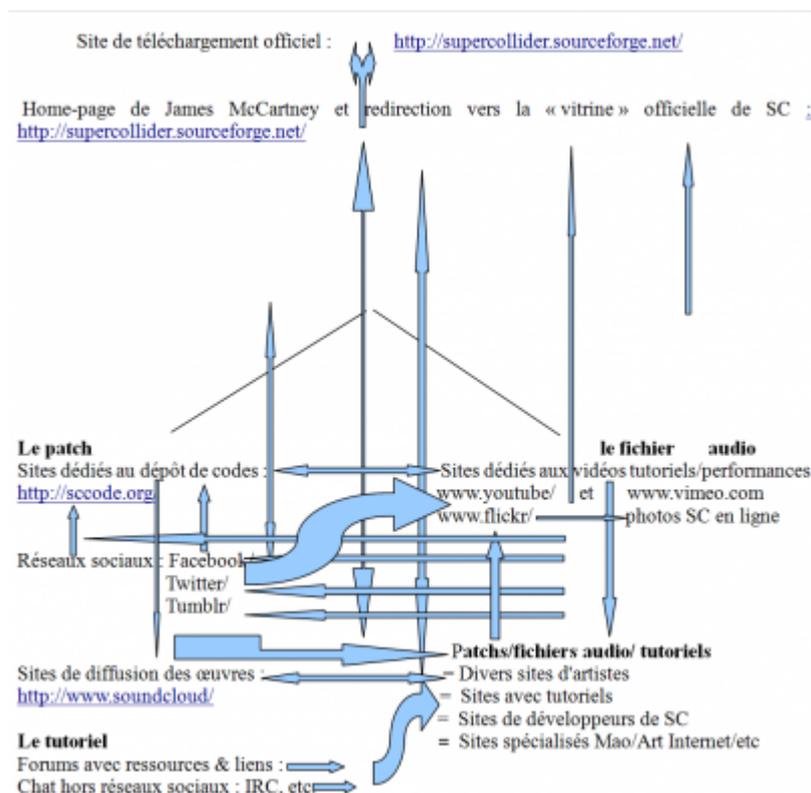
Diffuser, distribuer et bloguer sont, à notre sens, trois variantes sémantiques d'une démarche unique qui invite le musicien utilisant SC à considérer Internet comme le lieu privilégié d'une opportunité de sauvegarde globale et d'échange dans l'espace et le temps mais sans présager d'aucune certitude ni sur la durée ni sur le résultat musical.

Nous souhaiterions conclure cet article en faisant référence aux travaux de Wenger et Lave (Lave et Wenger, 1991 ; Wenger 1998) qui nous offrent de nouvelles perspectives d'analyse concernant ces « communautés de pratique ».

Ces travaux, qui concernent l'observation de l'apprentissage, les sciences de l'éducation et l'ingénierie des connaissances, mettent à notre disposition des outils méthodologiques qui pourraient être adaptés et fournir un cadre à notre étude sur la communauté des utilisateurs de Super Collider.

Le premier point est justement l'aspect dynamique que met en évidence leurs travaux dans cette dimension vivante de la connaissance partagée (ici la connaissance de SC) considérée sous l'angle du social et du relationnel entre membres de cette communauté. C'est ce que nous avons tenté de mettre en évidence dans cette « déambulation » de site en site en quête d'un tutoriel ou d'un enregistrement audio tel que décrit plus haut. Pour reprendre les termes de Wenger, cette dynamique de « participation » (appartenance et engagement social communautaire) et de « réification » (transformation d'une expérience en objet) se matérialise à notre avis tout à fait dans le patch, le tutoriel et le fichier audio qui sont les trois termes de notre angle d'analyse de SC.

La « réification » contribue à la fois à la notoriété du logiciel et de son appropriation possible par chaque membre de la communauté, complétant en quelque sorte la « participation » de chaque membre de la communauté. « Participation » et « réification » pour SC sont dans un rapport dynamique évident avec le concept de « répertoire partagé »⁽¹⁰⁾. Tous ces concepts devraient être évidemment précisés dans le cas d'une étude de la communauté de pratique de SC. L'application de la théorie de Lave et Wenger est une voie possible pour approfondir notre étude, car c'est bien la présence, le développement et la dynamique d'une communauté de pratique autour d'un logiciel et avec les moyens d'Internet qui garantit de nos jours la préservation d'un environnement logiciel tel que SC.



1. Nous insisterons moins sur la fenêtre « post-window » ou sur l'architecture logicielle particulière à SC avec ses deux serveurs, propre à la version 3 de SuperCollider, qui marque une avancée conceptuelle considérable moins en rapport avec notre problématique. Elle ouvre quasiment d'innombrables perspectives puisqu'elle permet à de nouveaux « clients » de communiquer avec le serveur interne créant les sons ou séquences musicales. « The success of the now open source SC3 server has seen a proliferation of client

architectures making use of the successful OSC communication protocol. »: (http://impromptu.moso.com.au/extras/sorensen_acmc_05.pdf). Nous pensons ainsi aux travaux passionnants, que nous suivons de près, de Thor Magnusson (Ixi lang), d'Andrew Sorensen (Impromptu utilisant le langage Scheme avec une bibliothèque optionnelle développée thor Magnusson afin de contrôler le serveur audio de SC dans Impromptu (<http://impromptu.moso.com.au/libs.html>) ou Sam Aaron (avec d'autres contributeurs au projet Overtone en langage Clojure) par exemple. On trouvera sur Vimeo bon nombre de vidéos pour ces trois programmes (www.vimeo.com).

2. Nous adoptons dans cet article le raccourci « code » pour désigner l'ensemble des lignes de codes informatiques constituant un programme et à fortiori un projet musical quelle que soit sa longueur. Ces raccourcis linguistiques traduisent bien ce « retour du code » dont on consultera les analyses passionnantes dans *Art++* ou dans l'ouvrage disponible en format PDF de Camille Paloque-Bergès : « Poétique des codes sur le réseau informationnel : une investigation critique ». Les deux ouvrages cités apparaissent dans la bibliographie.

3. On peut déplorer que pour les francophones peu de tutoriels existent en français hormis quelques pages intéressantes mais plutôt techniques sur le site de Vincent Rioux et de quelques autres trop rares contributeurs (cf. la Webographie). Un enjeu véritable existe de ce point de vue pour la communauté des utilisateurs français de SC en observant que pour mener à bien un tel projet, une connaissance approfondie de SC est bien sûr au préalable nécessaire. Pd ou Csound (certes avec une histoire déjà plus ancienne) bénéficient pleinement d'une démarche tutorielle comme celle proposés par les Flossmanuals (<http://fr.flossmanuals.net/>) consultables en ligne ou sous forme de PDF, des « spécialistes » mettent bénévolement leurs compétences en commun afin de réaliser des documents pédagogiques devenus depuis de vraies références. Le tutoriel est le pendant incontournable du programme gratuit ou non et décliné sous des formats spécifiques (.pdf, en ligne, vidéo, etc.). Il est certainement depuis quelques années le vecteur de communication privilégié pour beaucoup d'éditeurs de logiciels et de développeurs qui y apportent un soin tout particulier méritant à lui seul une étude particulière.

4. Dans l'aide SC, on trouve la précision suivante : « SuperCollider documents generally have .sc appended to their file names; however, SuperCollider can read and write documents in Rich Text Format (.rtf) and several other formats, as well. ». L'interchangeabilité du format texte est évidemment en lien direct avec notre propos.

5. Blank Pages est un cas intéressant puisque ces artistes adoptent une démarche *live coding* plus « radicale » selon laquelle rien ne sera enregistré ou archive?. L'instant de la création dans ce contexte esthétique est unique et toute « trace » sauvegardée est finalement sans intérêt. (www.blankpages.com)

6. Sur ce point, nous signalons que ce thème de réflexion est dans l'actualité d'une des expositions qui fut présentée au ZKM de Karlsruhe du 29/10/2011 au 12/02/2012, « The Challenges of Conservation » ([http://on1.zkm.de/zkm/stories/storyReader\\$7715](http://on1.zkm.de/zkm/stories/storyReader$7715))

7. Enfin, nous constatons ce retour en force du live coding sur la scène des « musiques actuelles » ou électroniques valorisant des stratégies d'une informatique musicale débarrassée de toute interface graphique (GUI) pré-conçue, privilégiant le patch ou un éditeur de texte pour partir d'une fenêtre vide rappelant souvent le principe du terminal. L'éditeur de texte open source E-macs est parfaitement intégré à SC (dans la version Linux) et apparaît dans le menu général de SC (et depuis juin 2012 dans Overtone, l'option d'installation Vim et Emacs existe mais « # Run at your own risk ! » est-il précisé : <https://github.com/overtone/emacs-live/tree/master/installer>).

8. La *Home page* SuperCollider est en quelque sorte « partagée », « distribuée » : on lira dans l'ouvrage de Jean-Paul Fourmentaux (Fourmentaux, 2010) des considérations intéressantes sur cet aspect que l'auteur analyse plus particulièrement dans le contexte du Net-Art. Par ailleurs, le site wiki SC de Hambourg contient des ressources intéressantes sur le *live coding* et nous préparons à ce sujet un article sur le « laptop orchestra » cité plus haut, (un orchestre d'ordinateurs composés de 4 musiciens) utilisant SC en session de live coding : <http://www.the-mandelbrots.de/>

9. D'autres initiatives du même ordre existent en rapport avec Twitter et notamment « Tweet a sound » (<http://www.soundplusdesign.com/?p=1621>) développé avec Max/MSP.

10. « Le répertoire d'une communauté de pratique comprend des routines, des mots, des outils, des

procédures, des histoires, des gestes, des symboles, des styles, des actions ou des concepts créés par la communauté, adoptés au cours de son existence et devenus partie intégrante de la pratique. Ce répertoire combine des éléments de réification et de participation et il comprend les interprétations des membres de même que les styles qui permettent d'afficher leur appartenance et leur identité ». Wenger 2005, p. 91.

Pour citer ce document:

Jean-Pierre Quinquenel, « Diffuser, distribuer, bloguer pour préserver : le cas de SuperCollider », *RFIM* [En ligne], Numéros, n° 2 - automne 2012, Mis à jour le 19/09/2012

URL: <http://revues.mshparisnord.org/rfim/index.php?id=203>

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