Numéros / n° 6 - Techniques et méthodes innovantes pour l'enseignement de la musique et du traitement de signal

Â«Â Quel intérêt pour des musiciens d?apprendre la programmation informatique ?Â Â»

Laurent Pottier

Résumé

Dans nos enseignements à l?université de Saint-Etienne, nous essayons d?initier les étudiants en musicologie, majoritairement littéraires plus que scientifiques, à la programmation informatique. De façon plus générale, nous nous interrogeons sur l?intérêt d?enseigner à un large public les bases de la programmation, qui permet de devenir acteur dans l?utilisation des ordinateurs et d?Internet et non simple consommateur. L?objectif n?est donc pas de transformer des musiciens en informaticiens, mais de susciter des vocations et de changer la perception de l?informatique et de son potentiel pour la création musicale.

Introduction

Nous exposons nos questionnements sur l?intérêt, dans des enseignements de musique liés aux technologies, d?essayer de sensibiliser des étudiants en musicologie, majoritairement littéraires plutôt que scientifiques, à la programmation informatique. De façon plus générale, nous posons la question de l?intérêt d?apprendre les bases de la programmation, pour devenir acteur dans l?utilisation des ordinateurs et d?Internet, plutôt que simple consommateur. L?objectif n?est pas de transformer tous les étudiants musiciens en informaticiens, mais de leur donner l?occasion de découvrir de quoi il s?agit, ce qui est ensuite susceptible de susciter des vocations, ou de simplement changer la perception qu?ils ont des outils informatiques et de leur rôle potentiel dans la création musicale.

La musique assistée par ordinateur se résume-t-elle à la connaissance des manuels des logiciels et à l?accumulation de matériel électronique divers ou doit-on plutôt parler d?informatique musicale, une discipline à travers laquelle le musicien peut combiner des instruments acoustiques et électroniques et développer ses idées musicales en créant son propre environnement, ses propres programmes sur mesure, sans se limiter aux injonctions des modes et des outils commerciaux ?

1. Certificats Informatique et Internet

Ces derniers temps, les ministères de l?Éducation nationale et de l?Enseignement supérieur ont mis en place, en France, des certificats destinés à valider les connaissances acquises par les étudiants ou par les enseignants dans le domaine des nouvelles technologies numériques.

Ainsi, tout étudiant destiné à une carrière dans l?éducation nationale est-il censé acquérir les compétences des référentiels du C2i et du C2i2e (Certificat Informatique et Internet Enseignant) [(1)].

Pour autant, quelles sont les compétences numériques de base attendues par un étudiant voulant obtenir ce certificat ? Neuf domaines de compétences sont à valider dans le C2i.

- A1 - Tenir compte du caractère évolutif des TIC
- A2 - Intégrer la dimension éthique et le respect de la déontologie
- B1 - S?approprier son environnement de travail
- B2 - Rechercher l?information
- B3 - Sauvegarder, sécuriser, archiver ses données en local et en réseau filaire ou sans fil
- B4 - Réaliser des documents destinés à être imprimés
- B5 - Réaliser la présentation de ses travaux en présentiel et en ligne
- B6 - Échanger et communiquer à distance
- B7 - Mener des projets en travail collaboratif à distance

Il n?est donc en aucun cas demandé de s?intéresser à un quelconque langage informatique. Il est uniquement question de communication, de travail collaboratif, de navigation Internet et de connaître les règles déontologiques régissant la publication de documents.

Pour le certificat enseignant, le C2i2e, rien de plus n?est indiqué sur la connaissance des technologies numériques requises et encore moins sur l?apprentissage des langages de programmation.

- I - Enseignement-apprentissage : conception, mise en ?uvre, analyse *a priori* et *a posteriori* de situations d?apprentissage mobilisant les TICE
- II - Pilotage, organisation, communication : conduite de projet, information des acteurs, valorisation d?actions, de projets
- III - Collaboration, interactions, formation : mobilisation des TICE pour se former, s?impliquer dans des projets de recherche, mettre en place une veille professionnelle, coopérer à des productions collectives, s?impliquer dans des réseaux professionnels
- IV - Responsabilité éthique et déontologie : exercer ses responsabilités éthiques et déontologiques en tant qu?enseignant/formateur mobilisant les TICE dans sa pratique professionnelle

Pourtant, au milieu des années 1980, le plan « Informatique pour tous » lancé par le gouvernement français avait des visées beaucoup plus ambitieuses au sujet de la façon de former les étudiants aux nouvelles technologies numériques. Il intégrait notamment l?apprentissage de la programmation avec les langages Basic et Logo. Dans les universités françaises, tous les étudiants passaient ainsi par l?apprentissage d?un de ces langages, à travers des applications souvent assez ludiques, comme de dessiner des figures géométriques en faisant se déplacer une tortue munie d?un crayon. Ces étudiants ne sont pas pour autant tous devenus des informaticiens, mais ils ont toutefois intégré des mécanismes importants pour la compréhension du fonctionnement des outils informatiques et pour la structuration de la pensée logique.

La programmation en LOGO est propice à provoquer une attitude de recherche. [?]

LOGO est un langage de programmation, certes rudimentaire, mais qui initie l?élève à la notion de procédure ou d?algorithme : il apprend vite à utiliser des boucles simples (la commande « répète »), et surtout à écrire des procédures appelant une ou plusieurs variables, et ce dès le CM2? Et rien n?est introduit gratuitement : les notions mathématiques apparaissent comme une réponse à un besoin des élèves, lorsqu?ils butent sur la réalisation d?un dessin. Ainsi, la variable devient assez vite un outil indispensable, que l?élève emploie de lui-même pour réaliser certains dessins qui seraient bien trop longs, voire impossibles à réaliser autrement [(2)].

Il est ainsi surprenant de voir aujourd?hui que, d?une part les langages se sont extrêmement développés,

simplifiés, intégrant des technologies graphiques et audio, et que de l?autre la distance séparant les utilisateurs de ces outils s?est considérablement accrue, un fossé ayant été creusé. Tous les étudiants à l?université utilisent quotidiennement un ordinateur, mais la très grande majorité d?entre eux est totalement réfractaire à l?idée même d?apprendre la programmation à travers un langage.

2. L?informatique musicale avant 1980

Pourtant les technologies numériques ne sont pas si nouvelles que cela, et en particulier dans le domaine de la musique.

Si les travaux d?Allan Turing ont préfiguré la naissance de l?informatique dans l?entre-deux-guerres, l?UNIVAC, premier ordinateur commercialisé, n?est apparu qu?en 1951. La musique s?est emparée très rapidement de ces nouveaux appareils, avant la fin des années1950, avec les travaux pionniers de Lejaren Hiller, qui a utilisé l?ordinateur en 1956 pour programmer la première pièce dont toutes les notes étaient programmées par ordinateur (*Illiac Suite for String Quartet*) et ceux de Max Mathews qui a inventé en 1957 le premier programme, Music I, un langage permettant de créer des sons de synthèse par ordinateur. Ce langage connaîtra de nombreux développements jusqu?à Music IV (1963) et Music V (1966) puis grâce aux travaux menés alors par plusieurs informaticiens : David Poole pour MUS10 (1966) sur PDP10 ; F.R. Moore pour C-Music (1980) ; Barry Vercoe pour Music 360 (1973), Music 11 (1978), puis Csound (1985) (Roads, 1996, p. 789), le langage spécifique le plus répandu ces trente dernières années pour la synthèse sonore par ordinateur.

Dans les années 1970, l?informatique musicale était réservée aux grands centres capables d?acquérir et d?utiliser des machines très coûteuses et complexes à programmer. De nombreux musiciens se sont intéressés aux technologies électroniques et ont expérimenté l?utilisation des synthétiseurs analogiques, que ce soit dans le domaine des musiques d?avant-garde (au San Francisco Tape Center avec les appareils Buchla) ou dans celui du rock progressif (ELP, Yes?), du rock planant (Tangerine Dream), des musiques techno (Kraftwerk) et du jazz (Chick Corea, Herbie Hancock).

Produire des sons avec des instruments comme le synthétiseur Moog modulaire nécessitait déjà certaines compétences en programmation, sans passer par un langage, mais en utilisant le concept de patchs, correspondant à des connexions électroniques qui permettent d?envoyer un signal d?une fonction vers une autre, afin de produire le son désiré.

3. L?informatique musicale après 1980

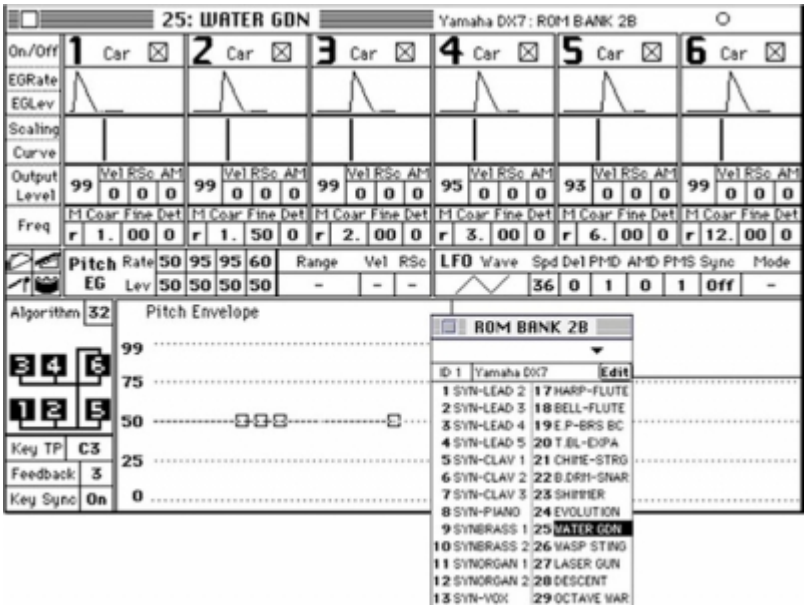
L?apparition du MIDI en 1983 a permis la connexion entre les synthétiseurs et les ordinateurs, donnant naissance aux séquenceurs (Pro24, Notator, puis Cubase, Logic ou Performer) et aux éditeurs de synthétiseurs (Synthworks, Galaxy) [(3)]. Si ces logiciels sont devenus rapidement assez populaires ? ils le sont restés jusqu?à nos jours, présents au c?ur des studios et représentant la majorité des outils utilisés par le grand public ? ils permettent déjà, pour certains d?entre eux, des approches de la programmation assez limitées, mais innovantes.

3.1. La programmation des synthétiseurs

Les éditeurs de la firme Steinberg, puis le logiciel Galaxy de la firme Opcode, devenu une référence, ont permis aux musiciens de programmer les synthétiseurs sur des logiciels spécialisés, offrant notamment la visualisation graphique des courbes d?enveloppes ou des représentations spectrales des filtres de fréquences. Enfin, ces programmes ont été utilisés pour créer les premières bases de données de sons de synthèse qui ont très rapidement profité des connexions mondiales par réseau (bien avant la naissance d?Internet) par modem, notamment en France grâce au Minitel et à certaines de ses applications spécialisées dans les années 1990. Le succès du DX7 Yamaha a également été dû au fait que des dizaines de milliers de sons ont ainsi pu être disponibles pour les musiciens sur ces réseaux.

Figure . L?éditeur Galaxy de la firme Opcode permet de visualiser les paramètres des sons du DX7,

parfois sous forme graphique (enveloppes, courbes de vélocité sur le clavier)

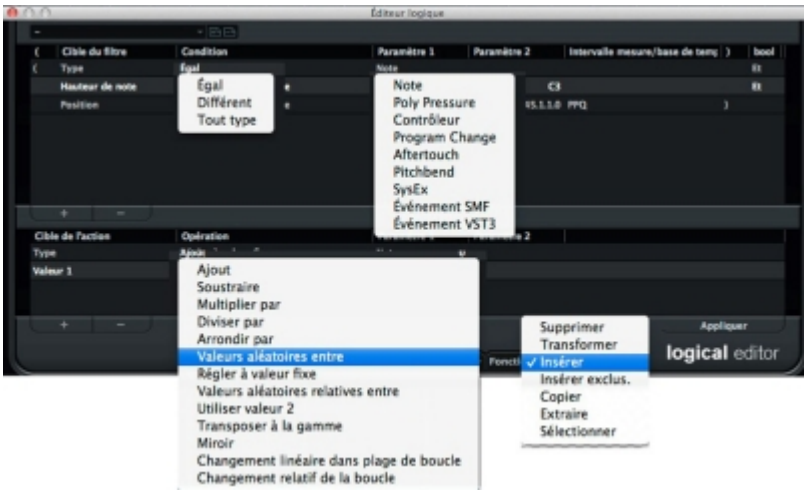


Source : auteur

3.2. Les éditeurs logiques

Le logiciel Cubase de la firme Steinberg, une des plus grandes références du marché des séquenceurs (et ce depuis sa première version sur ordinateurs Atari en 1986) a été le premier séquenceur à offrir un outil de transformation algorithmique de tout type d?événement MIDI. Cette fonctionnalité n?a pratiquement pas évolué depuis sa création, et elle est toujours absente de la plupart des séquenceurs commerciaux concurrents. Elle est certainement inconnue de la majorité des utilisateurs de Cubase, mais elle peut être une aide précieuse pour certaines manipulations d?éditions parfois laborieuses et c?est aussi une ouverture vers la composition algorithmique.

Figure . L?éditeur logique de Cubase (version Cubase8)



Source : auteur

3.3. Les langages de programmation pour la musique

Le langage FORTRAN est le premier langage de programmation de haut niveau à avoir vu le jour, en 1954, peu de temps avant le langage LISP (1958).

Ce langage a joué un rôle important pour le développement de l?informatique musicale, puisqu?il a permis aux langages Music N de Max Mathews, dans les années 1960, de sortir des laboratoires Bell et d?être déclinés en différentes versions : Music 6, Music V, Music IVF, avant que le langage C ne s?impose comme référence pour les dernières générations de ces programmes : CMusic et surtout Csound.

Des compositeurs comme Jean-Claude Risset ont réalisé des ?uvres de référence en utilisant le FORTRAN pour coder diverses fonctions de CAO pour le contrôle de la synthèse avec des commandes de haut niveau, en programmant des routines (PLF) pouvant gérer automatiquement l?évolution des paramètres de la synthèse, voire de la composition. Les fonctions PLF, écrites en FORTRAN dans MusicV par Jean-Claude Risset, ont aidé ce dernier a? effectuer des calculs répétés et en particulier a? modifier des structures de données. Il s?agit la? des fonctionnalités parmi les plus avancées du programme MusicV (cf. Pottier, 2006, p. 57-58).

Dans les années 1980, les tableurs ont été créés pour permettre de faire facilement des calculs sur les ordinateurs personnels. Les premiers enseignements que je me souviens avoir réalisés à l?université de Montpellier, dans un atelier d?informatique musicale, étaient initialement réalisés avec le programme Multiplan, prédécesseur d?Excel. Avec cet outil, nous entreprenions de générer des listes de notes, en utilisant des générateurs aléatoires, des fonctions de symétrie, des transformations d?échelles (hauteurs, rythmes), etc. Il fallait ensuite transcrire les résultats sur du papier à musique ou dans des séquenceurs MIDI.

Mais à cette période-là, le GRAME développait déjà un langage de pédagogie musicale et d?assistance à la composition en adaptant le langage LOGO pour qu?il puisse produire du MIDI et permettre de contrôler des synthétiseurs.

Logo Musical favorise le développement de la pensée musicale par une activité de composition [?]. Son originalité est double :

- il s?inscrit dans le champ peu abordé de la composition
- il utilise une approche pédagogique différente de l?enseignement programmé, en laissant le rôle actif à l?élève.

[?] Il permet de générer aussi bien des structures classiques de type rythmique et mélodico-harmonique que des motifs beaucoup plus complexes (Orlarey, 1984, p. 74-75).

L?environnement HyperCard développé par Apple (1987) a été également un logiciel conçu pour mettre à la portée de tous des outils de programmation simples, avec le langage HyperScript, notamment à travers le concept d?« Hypertexte ». Apple a développé dans cet environnement des outils intitulés HyperMIDI pour permettre de créer ses propres applications MIDI.

HyperMIDI est parfait pour des applications multimédias, vous donnant le même niveau de contrôle par script sur les événements musicaux que vous avez déjà sur le texte et les graphiques. Les enseignants l?utilisent pour créer des didacticiels interactifs. Les musiciens l?utilisent pour enregistrer et réécouter leurs idées, pour générer de la musique algorithmique, et pour des interfaces de contrôles graphiques de leurs périphériques MIDI [(4)]. (nous traduisons)

4. Les années 1990 et la CAO moderne

Les années 1990 ont vu apparaître les outils de la CAO moderne, notamment ceux développés à l?IRCAM. Carlos Agon décrit dans sa thèse les conditions qui ont permis de faire émerger la CAO moderne :

- la disponibilité des ordinateurs personnels,
- les progrès dans les interfaces graphiques,

- la création de standards, parmi lesquels, depuis 1983, la norme MIDI (Musical Instrument Digital Interface),
- les progrès réalisés dans le domaine des langages de programmation.

[?] L?utilisation d?un langage de programmation oblige le musicien à réfléchir sur les processus même de formalisation et lui évite de considérer l?ordinateur comme une boîte noire qui lui impose ses choix. Les langages de programmation offrent au compositeur une énorme liberté de décision, en échange d?un effort dans la formulation et la conception. (Agon, 1998, p. 10)

Issu de nombreuses années de développement ayant conduit, dans un premier temps, au langage FORMES (Rodet & Conte, 1985) implémenté en LISP (1982-1985), Patchwork, initialement conçu et développé par Mikael Laurson (Duthen & Laurson, 1990), a été l?environnement de l?IRCAM pour la CAO dans les années 1990. Basé sur la programmation graphique, sur le langage LISP, orienté objet (CLOS), intégrant des éditeurs de notation musicale et le MIDI, il a accueilli de nombreuses bibliothèques issues des compétences développées à l?IRCAM dans le domaine de la composition. À la fin de la décennie, c?est le programme Open-Music qui a pris la relève, développé par l?équipe Représentation Musicales de l?IRCAM (Assayag *et al.*, 1999). Plusieurs de ses bibliothèques sont par ailleurs consacrées à la synthèse des sons : les bibliothèques Om2Csound, pour le contrôle de la synthèse avec le moteur Csound, Chant pour le contrôle de la synthèse de la voix et des modèles de résonance, Modalys pour la synthèse par modèles physiques, ou encore OMChroma développée par Marco Stroppa.

5. Les années 2000 : Max MSP, FAUST et le temps réel pour les musiques mixtes

Depuis ses débuts, l?informatique musicale a toujours été scindée en deux domaines : le domaine du son et le domaine des notes, le domaine de l?audionumérique et celui de l?écriture, d?un côté la synthèse et le traitement de sons (DSP), de l?autre la composition algorithmique.

Tableau . Les deux grands domaines de l?informatiques musicale, celui du son et celui des notes et leurs champs associés

SON	NOTES
l'Audionumérique	le MIDI
Synthèse	Composition
Traitement	Ecriture
DSP	Contrôle
Flux pseudo continu	Flux discret
Taux d'échantillonnage	Evénements

Si les années 1990 ont surtout vu le développement des outils pour la composition assistée par ordinateur (CAO), c?est dans les années 2000, avec l?augmentation des puissances et des mémoires des ordinateurs, que le domaine du traitement du signal audio s?est vraiment développé, offrant la possibilité de manipuler le son en temps réel. Les technologies temps réel ont été initiées dans les années 1970, avec les premières stations de travail numériques comme le Synclavier aux USA, SYTER au GRM ou la 4X à l?IRCAM. Les recherches initiées avec la 4X ont permis par la suite le développement de l?environnement Max, initialement en MIDI, désormais utilisé pour la synthèse et le traitement du son en temps réel depuis l?intégration en 1997 des fonctions audio (MSP). Avec ces outils, il est devenu possible de transformer les ordinateurs en instruments de musique configurables à volonté. Les ?uvres du compositeur Philippe Manoury sont devenues des références dans le domaine des musiques mixtes interactives et elles ont beaucoup contribué à l?évolution du logiciel Max, notamment grâce aux travaux de Miller Puckette autour de *Jupiter* (1987), *Pluton* (1988), *La Partition du Ciel et de l?Enfer* (1989), *Neptune* (1991) et *En Echo* (1994).

En 2003, le GRAME a réalisé le langage FAUST (Gaudrain & Orlarey, 2003) qui est devenu un standard dans le domaine du traitement audio temps réel par ses qualités offertes, un langage concis, fonctionnel, efficace, fiable, portable, et par son caractère open-source qui lui a permis de fédérer une communauté active qui contribue fortement à l?élargissement de ses champs d?action.

6. Les années 2010 : ubiquité, réseaux, smartphones

Dans la dernière décennie, certains outils sont devenus ubiquitaires, disponibles dans toutes circonstances, sur tous types de plateformes. En particulier, le WEB est devenu une plateforme à travers laquelle il est possible de communiquer mais pour laquelle il est aussi possible de créer des applications diverses. Avec JavaScript et les Web Audio API on peut maintenant développer des applications de traitement du signal et leur associer des interfaces graphiques utilisateur performantes. Certes les performances en calcul, en interfaçage et en compatibilité ne sont pas encore toujours au rendez-vous, mais nous pouvons penser que ce n?est qu?une question de temps. Des applications comme le clavecin + (Cipierre & Pottier, 2015) ou Touchvoices [(5)] (Théo, 2016) en sont quelques illustrations.

Des appareils de plus en plus compacts comme les smartphones actuels ou les Raspberry Pi, Arduino et autres microcontrôleurs permettent, grâce à la programmation, de construire des instruments miniaturisés qui peuvent être utilisés simplement pour la capture du geste, grâce à des accéléromètres, gyroscopes et autres capteurs, mais aussi pour produire et transformer du son en temps réel. Les applications Smartfaust ont permis de produire des concerts dans des écoles et de réaliser des concerts participatifs faisant intervenir le public avec ses propres smartphones. Nous avons également utilisé ces technologies dans nos différents enseignements à l?université pour inciter les étudiants à créer des musiques dont ils construisent eux-mêmes les instruments sur leurs téléphones portables ou leurs ordinateurs.

Figure . Xavier Garcia dirigeant un concert de smartphones (Smartfaust, festival Musiques en scène, Lyon, 29 mars 2014) à gauche ; répétitions d?étudiants à l?UJM en création sur smartphones et ordinateurs (déc. 2015) à droite

Sources : Photographie de L. Pottier (à gauche) ; photographie de P. Landrivon (à droite)

7. Informatique musicale à l?UJM

Les enseignements que nous proposons en lien avec les nouvelles technologies à l?université de Saint-Étienne, dans le département de musique, sont nombreux et comportent des formations professionnalisantes au niveau licence (pro) et master (pro). Les enseignements regroupent des cours théoriques (histoire des outils et des musiques électroacoustique, acoustique), mais ils comportent surtout beaucoup de cours pratiques, alliant la plupart du temps création et programmation, dans des ateliers sur la composition électroacoustique, la lutherie numérique, la création sonore pour l?image, le temps réel, les outils de communication audiovisuels, etc. [(6)]

Conclusion

Tout est fait dans notre société pour nous inciter à la consommation. L?industrie commercialise des produits qui sont là pour répondre à nos besoins de consommation, besoins que les médias nous incitent fortement à faire correspondre aux standards de cette même industrie. Dans le domaine de la création musicale, il n?est, et ce depuis fort longtemps, plus besoin de démontrer les ravages que font ces médias pour nous faire consommer des musiques toutes faites sur le même modèle, et toujours à renouveler.

Il nous faut constater avec regret que les maisons de disques ? dont l?esprit est orienté vers le profit le plus grand et le plus rapide possible ? ne jouent pas le rôle que l?on pourrait attendre d?elles. Elles laissent passer maintes occasions d?enregistrement qui ne se présenteront plus quand le compositeur ne sera plus en vie. [...] En revanche, on est d?autant plus prompt à enregistrer les rengaines et autres airs à la mode. (Bartok, 1937, p. 196)

Pourtant, la création artistique part du besoin des créateurs de trouver leur identité et d?explorer des mondes nouveaux. La programmation peut nous aider car elle nous permet de développer notre imaginaire, d?aller au-delà de nos habitudes, de dépasser les limites des outils numériques préfabriqués, grâce à la formalisation, au traitement de l?information et à l?expérimentation.

Quelques pistes nous permettent d?entrevoir l?avenir avec un certain optimisme.

Jason Freeman réalise des enseignements (*Survey of Music Technology*) sur les technologies musicales, et notamment sur son logiciel très prometteur EarSketch (présenté ailleurs en détail dans ce numéro), à travers la plateforme Coursera, ouverte à tous.

Comment pouvons-nous utiliser les ordinateurs pour créer de la musique expressive et convaincante ? Et comment pouvons-nous écrire des logiciels pour nous aider à créer et organiser les sons de façon différente ? (7) (nous traduisons)

Nous citerons également à nouveau le programme FAUST du GRAME qui a fait des adeptes aux USA (CCRMA-Stanford), en Allemagne (Mainz), en Irlande (Maynooth), en Arménie (Tumo), en Chine (Shanghai, Taipei)? Ces outils sont directement accessibles sur Internet, et ne demandent nulle autre installation que celle d?un navigateur. Avec la technologie FaustPlayGround (8), ils permettent rapidement de construire des instruments sophistiqués à partir des nombreuses bibliothèques déjà disponibles.

Toutefois, le compositeur Marco Stroppa reste encore très méfiant sur la qualité des sonorités produites par des instruments DSP temps réel.

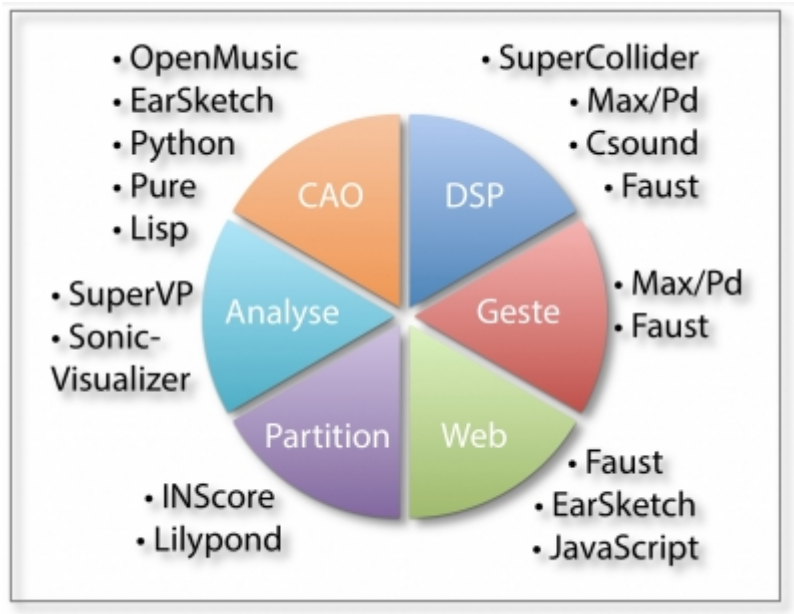
Ce qui me manque, c?est le choc esthétique. Nous avons un choc technologique, mais une musique minimale et une pensée minimale. Pouvons-nous produire quelque chose de plus intéressant que seulement de mauvais sons de guitare électrique avec la technologie temps réel ? (9) (nous traduisons)

Beaucoup de musiciens sont intéressés par la construction d?instruments électroniques, au traitement du signal, au temps réel, mais les questions liées à l?écriture, à la composition, à la création de sons à fort potentiel musical, à leurs transformations, et à leurs variations restent à explorer.

Il faudrait donc un logiciel qui permette d?appréhender la pensée musicale et de questionner l?élève sur chaque signe que l?on place sur la partition. Un logiciel qui permette de penser la musique en l?écrivant, et non pas de faire seulement de la mise en page (Rascle, 2014).

Le tableau suivant résume les différents champs de l?informatique musicale sur lesquels nous travaillons et les outils que nous utilisons dans nos divers enseignements à l?université Jean Monnet de Saint-Étienne.

Figure . Les principaux secteurs de l?informatique musicale autour de la synthèse et du traitement du son et quelques outils et langages associés



2. Site de l?IREM développant le logiciel GéoTortue, logiciel inspiré du langage LOGO pour enseigner les mathématiques et l'algorithmique.

http://www-irem.univ-paris13.fr/site_spip/spip.php?article32

3. Pro24 (Steinberg), Notator (C-Lab, puis Emagic), Cubase (Steinberg), Performer (Mark of the Unicorn), Synthworks (Steinberg), Galaxy (Opcode), etc.

4. « HyperMIDI is perfect for multimedia applications, giving you the level of scripting control over musical events that you already have over text and graphics. Educators use it to create interactive courseware. Musicians use it for recording and playing back ideas, for generating music algorithmically, and for virtual front panels for MIDI devices. Multimedia authors use it for interactive titles and presentations ».

http://www.earlevel.com/HyperMIDI/about.html

5. http://patheo.github.io/TouchVoices/

http://patheo.github.io/TouchHarp/

6. cf. http://musinf.univ-st-etienne.fr/pedagogie2.html

7. Jason Freeman : « How can we use computers to create expressive, compelling music? And how can we write computer software to help us create and organize sounds in new ways? »

https://www.coursera.org/learn/music-technology

8. http://faust.grame.fr/faustplayground/

9. Marco Stroppa, intervention lors du colloque DSP à Saint-Étienne le 3 novembre 2015 : « What is missing for me, is the aesthetic shock. We have a technological shock, but minimum music and minimum

thinking. Can we produce more than a cheap electric guitar sound with the real-time technologie? »

Pour citer ce document:
Laurent Pottier, Â« Quel intérêt pour des musiciens d?apprendre la programmation informatique ? Â», *RFIM* [En ligne], Numéros, n° 6 - Techniques et méthodes innovantes pour l'enseignement de la musique et du traitement de signal, Mis Ã jour le 22/06/2018
URL: http://revues.mshparisnord.org/rfim/index.php?id=488