

AGÔNE: UN OUTIL ADAPTÉ À UNE DÉMARCHE DE CRÉATION EN COMPROVISATION ÉLECTROACOUSTIQUE

Jullian Hoff

jullian.hoff@gmail.com

RÉSUMÉ

Il est discuté de la improvisation comme démarche artistique visant à cumuler les avantages du temps réel et du temps différé en musique mixte ainsi que de la réalisation et du fonctionnement des différents mécanismes d'*Agône*, outil dédié à la improvisation électroacoustique pour guitare électrique, synthétiseurs, échantillonneurs et projecteur vidéo. Il est également discuté de l'enjeu pour un compositeur – instrumentiste – improvisateur d'envisager le développement de ses propres outils favorisant l'élaboration d'une démarche artistique unique. Enfin, la création d'une première improvisation créée avec *Agône* est commentée.

1. COMPROVISATION ÉLECTROACOUSTIQUE

Agône est né de l'envie de créer une musique électroacoustique qui allierait les possibilités de la composition en studio – temps différé [1] – (gestation de la réflexion, contrôle paramétrique [6]) à celles de la performance instrumentale – temps réel – (musicalité du geste instrumental [3], contrôle musical [6]), ainsi qu'à l'improvisation (spontanéité des affects, forme évolutive). La volonté de fusionner ses temporalités m'a naturellement amené à la improvisation comme démarche artistique.

Dudas [2] identifie : «...two basic species of composition-improvisation relationships intrinsic in working with electronic and computer music: (1) composing an "instrument" that can be improvised upon in performance, and (2) improvising with tools in order to create pre-compositional material. ». *Agône* s'inscrit dans la première approche. Il propose un environnement permettant – lors de la phase de composition – le paramétrage des instruments qui seront impliqués lors de la performance improvisée. La volonté initiale était que ces instruments puissent être totalement autonomes lors de la performance, donc pleinement gérés par les algorithmes mais également adaptatifs envers le jeu de l'improvisateur et donc, capables d'improviser avec lui.

2. ENTREPRENDRE DE DÉVELOPPER UN OUTIL UNIQUE

Dudas [2] résume la définition que propose Tanaka [9] d'un *Computer music performance system* comme suit : *...generally contains an input device to acquire data, mapping algorithms to translate data into musical information, a sound synthesis engine to be played by the live input, a compositional structure to define the musical progression of the work and an output system to diffuse/perform the resulting sounds in the performance space.* Cette définition correspond tout à fait à mes besoins. Plus spécifiquement, j'avais besoin d'algorithmes génératifs et d'une interface utilisateur pour les paramétrer lors de la phase de composition, d'outils d'analyses du signal de la guitare et de possibilités de *mapping* entre la guitare et ces algorithmes ainsi que de sorties diverses et assignables vers des synthétiseurs, échantillonneurs ainsi que vers un logiciel de *Vjing*.

Développer mon propre outil a été un processus particulièrement chronophage (en fait constamment inachevé) et souvent frustrant voire pénible. Ce temps passé au développement de l'outil n'est pas du temps consacré au perfectionnement de l'utilisation d'instruments acoustiques et électroacoustiques ni à l'élaboration de stratégies de composition et à la résolution des problèmes qu'elles impliquent. Pourquoi ai-je alors choisi de développer *Agône*?

Je souhaitais réunir dans un seul environnement tous les éléments nécessaires à ma démarche de création et avoir un contrôle intime sur ceux-ci afin qu'ils répondent spécifiquement à mes besoins. En ce qui concerne les algorithmes génératifs, on trouve des logiciels particulièrement performants et ergonomiques à ce niveau, comme *Nodal* [14], *Noatiki* [13] ou *Bloom* [11], qui chacun à leur manière permettent à l'utilisateur de construire du matériel mélodico-rythmique ou tramé au travers d'interfaces principalement graphiques et/ou tactiles. Ces outils se trouvaient inadaptes à ma démarche par le fait que je souhaitais des algorithmes totalement autonomes lors de la performance afin de me concentrer sur la guitare et l'improvisation. Du côté de l'analyse en temps-réel du jeu de l'improvisateur en vue de générer en

conséquence du matériel musical improvisé par l'ordinateur, *OMax* [15] propose des solutions de réinjection et d'analyse qui vont bien au delà de ce que j'ai implémenté dans *Agône* et que je détaille plus loin. Seulement, *OMax* est orienté vers le temps-réel et la génération de clones musicaux en fonction de l'apprentissage du jeu de l'instrumentiste. Pour ma part, je souhaitais un outil de composition en temps-différé mais qui pourrait également s'adapter en temps-réel à l'improvisateur et serait capable de générer du matériel de façon autonome au besoin, même sans données entrantes ou stockées. Je souhaitais que l'ordinateur puisse aussi bien prendre le rôle du soliste que de l'accompagnateur ou même agir comme un orchestrateur.

L'agglomérat de fonctionnalités complémentaires issues de plusieurs logiciels aurait été une solution envisageable. Ainsi, le projet *Native Alien* [7] qui poursuit la réalisation d'un environnement de improvisation complet utilise plusieurs outils issus de l'*IRCAM* dont *OMax* et *ModalyS* [12] et leurs adjoignent différents programmes développés par les membres de leur équipe de recherche.

Je pense cependant que le développement d'outils uniques offre au créateur, en recherche de postuler un langage singulier, des avantages intéressants: une connaissance approfondie des processus impliqués qui permet d'en conceptualiser leurs améliorations et de poursuivre leur développement continu parallèlement à l'élévation de la démarche artistique. Je crois également au poésis de l'outil, que la réalisation de celui-ci dicte d'une certaine façon le résultat que l'on en obtiendra et que de fait, réaliser l'outil c'est déjà entamer le processus de création de l'œuvre. *Agône* a été développé dans l'environnement de programmation *MaxMSP*.

3. UN LOGICIEL POUR QUELLE VISION MUSICALE?

Avec *Agône*, l'enjeu était de pouvoir créer des improvisations polyphoniques et polytimbrales combinant à la fois des caractéristiques musicales, des fonctions et des processus prédéterminés en amont par le compositeur [8], mais également des comportements adaptatifs sujets à réagir en fonction des décisions spontanées de l'instrumentiste et de sa capacité à improviser. C'est donc cette notion de dialogue musical qui a orienté les choix de développement. Pour cela, je voulais un logiciel permettant à la fois une abstraction de paramètres musicaux comme les hauteurs et les durées paramétrables lors de la phase de composition allié à une capacité d'analyse du jeu instrumental en temps-réel afin de générer du matériel musical improvisé par des algorithmes génératifs [10]. Également, je souhaitais intégrer aux différents algorithmes une part importante d'aléatoire contrôlé afin que les programmes puissent

"improviser" à leur façon. Enfin, je souhaitais une dimension multimédia et donc qu'*Agône* puisse gérer le déclenchement de contenu vidéo interactif et polycanal.

Il n'est pas toujours aisé de comprendre les intentions du compositeur pour un auditeur. Il ne lui est pas non plus toujours évident de comprendre la relation d'un instrumentiste sur des contrôleurs électroniques et leurs effets sur le rendu musical entendu même si ceux-ci sont projetés sur un écran tactile transparent comme la mode actuelle tend à le proposer. Il est par contre relativement plus acquis pour la plupart des auditeurs de comprendre les relations entre les gestes musicaux sur un instrument traditionnel comme la guitare électrique et leurs effets sur le rendu musical (au niveau des modes de jeu de base en tout cas). Les décisions d'interactions entre *Agône* et la guitare et les stratégies de *mapping* utilisées pour la première improvisation réalisée par cette alchimie¹ ont été portées par ces considérations. L'instrumentiste joue littéralement avec *Agône* et j'ai cherché à montrer cette relation de jeu au public autant que faire se peut. Idéalement, je cherche à personnifier *Agône*, à le rendre perceptible par les spectateurs, à lui conférer une présence. Les choix de *mapping* aident à créer cette illusion. Par exemple, le registre des hauteurs des instruments joués par *Agône* peut suivre celui du guitariste ou au contraire jouer dans un registre opposé. Cette relation est assez évidente pour le public et mène à la perception d'un joueur invisible. Mais le rendre visible est tentant, j'ai donc réalisé pour la première improvisation réalisée avec ce logiciel des fragments vidéos que j'ai associés à certains de ces instruments. Par exemple, pour une section particulière, *Agône* contrôlait un instrument harmonique jouant 4 notes, soit en homorythmie, soit selon des motifs syncopés. Chaque note de ces accords déclenchait aléatoirement un fragment vidéo (et l'éteignait, par le jeu des *notes-on* et *notes-off*) représentant l'ondulation abstraite d'une onde. Par *composition* de ces fragments vidéo, la « voix » harmonique d'*Agône*, qui faisait écho au jeu de l'instrumentiste, devenait sensible à l'écran. Je pense que les mécanismes de rhétorique du dialogue sont une clé pour la démarche que je poursuis avec ce logiciel.

4. PRINCIPE GÉNÉRAL D'AGÔNE

Agône procède à l'abstraction de l'idée musicale en un ensemble de classes correspondant à différents paramètres dont le registre, les motifs rythmiques, la dynamique, les hauteurs, la conduite des voix, les motifs mélodiques, l'approche chromatique, les durées de notes, l'accentuation, la polyphonie et le tempo. Afin d'obtenir des algorithmes génératifs le matériel musical nécessaire à la réalisation de son idée, le compositeur définit des

¹*Denise, Agnus & Paula*, créée le 5 Nov. 2014 à l'Usine C, lors du festival Akousma XI. <http://vimeo.com/111930919>

contraintes et conditions au travers d'interfaces utilisateurs comprenant des *multisliders*, des matrices de boutons, des tables de données et des boîtes à chiffres. En sortie, l'ensemble des paramètres musicaux sont compilés en données MIDI puis envoyés via plusieurs canaux qui sont reçus en temps-réel par des synthétiseurs et des échantillonneurs afin que le compositeur puisse y travailler sur la matière sonore, gérer les timbres et leurs relations et prendre en considération la spatialité de l'œuvre.

Plusieurs des paramètres musicaux gérés par la machine sont également altérés par des outils d'analyse en temps réel qui comprennent: La mémorisation à durée variable des notes jouées par le guitariste avec une indication des cordes sur lesquelles elles ont été jouées², le suivi d'enveloppe du signal de la guitare, le niveau d'intensité d'attaque corde par corde, le débit d'attaque au médiator, l'analyse des motifs rythmiques joués, le suivi du registre, celui de la position sur le manche au travers des axes horizontaux et verticaux, l'écoute du nombre de notes jouées simultanément et la réception de plusieurs contrôleurs dont des interrupteurs au pied et des pédales d'expression. Les relations entre les paramètres musicaux prédéfinis dans l'interface utilisateur d'*Agône* et les différents paramètres des synthétiseurs et échantillonneurs logiciels avec leurs altérations par les outils d'analyses correspondent à la phase de *mapping* [5]. Un *mapping* réfléchi par le compositeur et maîtrisé par l'instrumentiste aide à ce que la pièce gagne en caractère interactif, mais pas seulement. La plupart des paramètres d'*Agône* reposent sur la gestion de tables de probabilité ou d'expressions générant des résultats aléatoires. La maîtrise de cette part d'aléatoire par le compositeur consiste en son contrôle du poids des probabilités dans le cas des tables et sur l'ambitus des valeurs générées dans le cas des valeurs *décidées* par la machine. L'improvisateur quant à lui, écoute en permanence le matériel musical généré en temps réel afin de pouvoir s'adapter et renforcer l'interaction à celui-ci.

5. DÉFINITIONS DES MÉCANISMES D'AGÔNE

(les exemples d'application de ces mécanismes sont tirés de l'œuvre *Denise, Agnus & Paula* [4], première improvisation exploitant le logiciel)

² *Agône* a été spécifiquement conçu pour fonctionner de paire avec une guitare équipée d'un microphone hexaphonique, dit microphone divisé. Cela consiste en un bobinage unique pour chaque corde, ce qui permet de faire du suivi de hauteur par corde et permet également de facilement connaître la position du guitariste sur le manche puisque chaque corde envoie son propre signal. Ce type de microphone permet bien plus de possibilité d'analyse que les microphones standards à bobinage sommatif.

5.1. La mise en mémoire des notes jouées par le guitariste

Les notes jouées par le guitariste sont stockées dans des mémoires de tailles variables. Il y a une mémoire par corde. La taille de la mémoire consiste en sa capacité à retenir les x dernières notes jouées, x étant décidé par le compositeur. Les mémoires consistent en des associations d'objets *coll* et *table* dans lesquels le logiciel va piocher selon des principes de probabilités. Ainsi, les notes les plus jouées par l'instrumentiste ont le plus de chance d'être choisies par le logiciel, ensuite, ces choix de notes sont soumis à des lois relevant du mécanisme de conduite des voix (voir 5.4). L'instrumentiste quant à lui à la possibilité d'activer ou de désactiver l'apprentissage des mémoires lors de la performance et de les rafraîchir à tout moment par pression sur une pédale. Le compositeur décide de l'assignation des 16 voix de polyphonie gérés par *Agône* aux différentes mémoires. Cela exacerbe le potentiel orchestral de la guitare, car les différentes voix de polyphonies sont ainsi reliées – notamment leurs hauteurs – au jeu du guitariste et à sa gestion polyphonique de la guitare. Par exemple, une voix de polyphonie peut être consacrée à une ligne de basse et puiser ses notes sur les mémoires associées aux cordes ⑤ et ⑥, tandis que 3 autres voix d'harmonisation pourront piocher des hauteurs jouées sur les cordes ③ à ⑤ et enfin, un élément mélodique pigera dans les hauteurs jouées sur les cordes ① à ③. Dans cet exemple, le guitariste pourrait jouer des hauteurs issues de la triade majeure de Do sur les cordes ⑤ et ⑥, de la triade mineure de Ré sur les cordes ③ à ⑤ et de la tétrade m7 de Mi sur les cordes ① à ③. *Agône* quant à lui ferait donc jouer sa basse avec des notes issues de Do m, ses 3 voix d'harmonisation jouerait les notes de Ré m et Do M tandis que son instrument soliste pigerait dans Ré m et Mi m7.

5.2. La gestion du registre

Le registre est déterminé pour chaque voix de polyphonie, il possède deux modes: le premier consiste en un offset positif ou négatif de x octaves par rapport au registre analysé en temps réel du guitariste. Le second mode est une assignation fixe sur une échelle de 9 octaves. Comme je l'expliquais précédemment, le registre est un élément basique mais efficace pour créer une illusion de jeu interactif concluant. Je l'ai utilisé pour qu'*Agône* me suive parcourant l'ambitus des hauteurs ou qu'au contraire, il se tienne à l'opposé du registre que j'occupe lors de l'improvisation.

5.3. La génération des cellules rythmiques

La génération des motifs rythmiques est la charge de 16 mécanismes, un par voix, nommés *Cellmakers*. Ces

mécanismes sont inspirés d'un algorithme rythmique présenté par Olivier Bélanger lors d'un cours sur l'algorithmie musicale donné à l'université de Montréal en 2011 et basé sur un algorithme utilisé par Pr. Jean Piché. Les *Cellmakers* ont la capacité de subdiviser un temps relatif en différentes cellules, allant de débits de croches, triolets, doubles-croches jusqu'aux quintolets en passant par de nombreuses combinaisons incluant des silences, pour un total de plus de 40 figures rythmiques différentes possibles sur 1 temps musical. Le compositeur décide, par le biais de tables de probabilités, du poids de chaque cellule et ce pour chaque voix. En fait, par souci de simplicité, les 16 *Cellmakers* partagent 3 tables de probabilités, chaque voix de polyphonie étant assignée à l'une d'elle. Afin d'avoir la possibilité d'utiliser des motifs rythmiques plus rapides ou plus complexes tels des neunolets ou des cellules rythmiques se déroulant sur plus d'un temps musical comme des triolets de noires, des multiplicateurs de rythmes sont assignables à chaque voix, laissant la possibilité au compositeur de multiplier les motifs de chaque voix par un facteur de 0.25, 0.5, 0.75 ou 1.5 etc. Ainsi, une figure de deux croches multipliée par 1.5 donneront deux croches pointées. Ce procédé de multiplication et division permet également de dépasser la limite auto-imposée des trois tables de probabilités de base sans besoin d'en ajouter de supplémentaires. La requête pour une nouvelle figure rythmique est envoyée selon un intervalle déterminé par le compositeur tous les x temps. Ainsi le comportement rythmique des instruments gérés par la machine sont décidés dans leur comportement global lors de la phase de composition (tel instrument jouera des motifs rapides, tel autre des sons entretenus) mais le détail des enchaînements des figures rythmiques relève de la probabilité induite par les objets *table* utilisés.

5.4. La conduite des voix

Elle dépend de plusieurs mécanismes. Il y a d'abord un mécanisme chargé d'envoyer la requête d'aller piocher dans les mémoires une nouvelle note selon le placement rythmique de celle-ci. Ainsi, le compositeur peut par exemple décider de changer de hauteur de note seulement sur chaque deuxième double-croche du temps pour la voix v_1 et sur les pulsations pour les voix v_2 et v_3 . La requête est ensuite soumise à un autre mécanisme déterminant si la note doit être changée ou si la hauteur précédente doit être répétée. Ce mécanisme est contrôlé par le compositeur au moyen de probabilités exprimés en %. Ensuite, la nouvelle note est soumise à un système de conditions déterminant si l'intervalle entre la nouvelle note et la précédente est compris dans un ambitus autorisé par le compositeur ou sinon une nouvelle requête est envoyée. Enfin, la nouvelle hauteur est soumise à un mécanisme d'approche chromatique, déterminant, selon une chance de probabilité exprimé en % par le compositeur si la note sera approchée

chromatiquement. Le choix d'une approche supérieure ou inférieure est laissée au soin d'*Agône*. Ici encore, le comportement global de la conduite des voix est grandement conditionnée lors de la phase de composition. L'on pourra choisir de restreindre une voix à des mouvements conjoints ou l'autoriser à de grands sauts d'intervalles, forcer telle autre à des approches chromatiques. On pourra également imposer à une voix de nombreuses répétitions de la même hauteur ou au contraire, favoriser plus de mouvement dans le phrasé. Mais dans le détail, les *choix* de notes seront toujours laissés au soin des algorithmes lors de la performance.

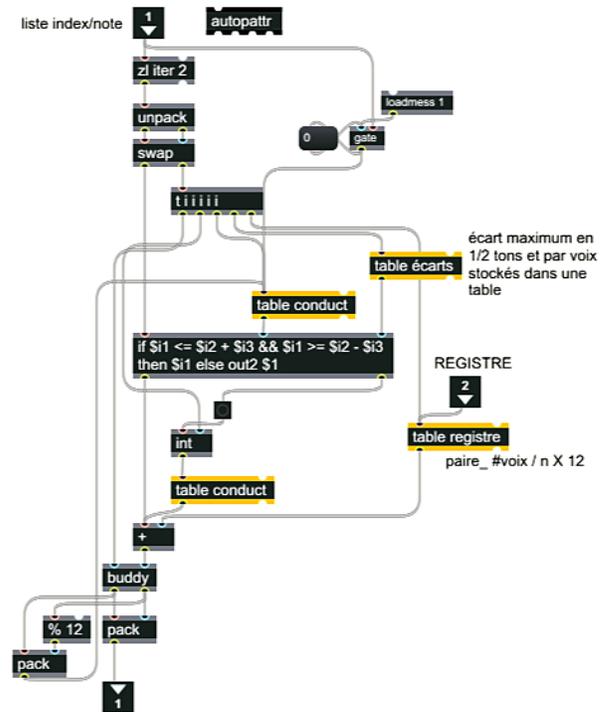


Figure 1. Le sous-module assurant le respect des sauts d'intervalles dans la conduite des voix.

5.5. Les profils mélodiques

Les profils mélodiques sont déterminés par 6 arpégiateurs, nommés *Rpeggiovano*. Il y en a un par mémoire. Le nombre de pas pour chacun d'eux est variable, chaque pas ayant une fonction de note cible assez complète (note la plus basse, la plus haute, intervalle miroir, aléatoire, note suivante, précédente etc.) et des possibilités d'offset de registre par pas ($\pm x$ octaves) ainsi qu'une gestion du phrasé (legato – staccato). Enfin, chaque voix de polyphonie à la possibilité d'être assignée à tel ou tel *Rpeggiovano*, offrant ainsi la possibilité d'affecter plusieurs voix de polyphonie au même arpégiateur et ainsi d'avoir des profils mélodiques discrétisés par plusieurs instruments. (cet outil est une amélioration par rapport à la

version du logiciel utilisée pour créer *Denise, Agnus & Paula* [4], il sera grandement utilisé dans la deuxième improvisation que je réalise présentement avec *Agône*.

5.6. Les durées

Elles dépendent de plusieurs facteurs: Tout d'abord, le compositeur décide d'une constante pour chaque voix exprimée en %. À 100%, une noire à 60 noires par minute fera 1 seconde. Ensuite, il y a un offset positif ou négatif de ce % qui est appliqué par un *Low Frequency Oscillator* aux variables déterminées par le compositeur. Enfin, une expression attribue une légère variation aléatoire à chaque nouvelle note. Les *Rpeggiovano* ont également une influence note à note sur ce paramètre. Comme expliqué précédemment, les motifs rythmiques sont obtenus à partir des *Cellmakers*. Les mécanismes altérant les durées ont été ajoutés afin de permettre le choix entre un jeu *legato* ou *staccato* ainsi que d'ajouter plus d'aléatoire à la musique générée par la machine et d'en *humaniser* le résultat.

5.7. La gestion du tempo

Il est contrôlé via 4 modes. Le premier est l'expression d'un tempo fixe prédéterminé par le compositeur pour chaque *preset* (tous les paramètres du logiciel sont sauvegardés dans des presets appelés par l'improvisateur à l'aide de pédales MIDI). Le deuxième est calculé selon un système de tap tempo contrôlé par l'instrumentiste. Le troisième consiste en un mécanisme de suivi du tempo du musicien, basé sur une analyse des différents débits rythmiques de base (croches et doubles-croches) et de leurs résultats statistiques avec la possibilité pour le compositeur d'ajuster la tolérance à ceux-ci (technique classique d'analyse des *delta-time*). Enfin, le quatrième mode permet la décision d'un nouveau tempo à chaque nouvelle note du guitariste basé sur l'analyse de la durée entre les deux dernières attaques de médiator. Au niveau improvisationnel, ces différents modes offrent beaucoup de choix. Ici encore, choisir l'option du *tempo follower* créera l'effet que la machine réagit à l'instrumentiste. Choisir que le tempo soit recalculé constamment selon les deux dernières attaques du guitariste donnera lieu à des changements de tempo drastiques ou pas, selon les décisions de l'improvisateur. Au contraire, un tempo fixe et prédéterminé donne plus de contrôle au compositeur.

5.8. Les intensités

Les intensités de chaque voix sont déterminées par 5 facteurs : Le suivi de l'enveloppe d'amplitude du signal de la guitare (somme de l'amplitude générale via l'entrée *jack*, sans considération de l'amplitude du signal de chaque corde), une constante par voix décidée par le compositeur

correspondant à une échelle en % allant de pas d'atténuation au silence total, une variation de ces constantes en temps-réel par le biais de LFOs, de légères variations calculées par des opérateurs mathématiques aléatoires, enfin, par une considération de la vitesse de corde par corde provenant du microphone MIDI dont la guitare est équipée. Il est clair que le jeu des nuances et des intensités est un levier de choix pour un musicien qui veut donner vie à ce qu'il joue. Pour de la musique générée par des algorithmes, inutile de lésiner à ce niveau là, d'où la multiplicité des mécanismes. Ici encore, mes choix d'implémentation ont cherché une balance variable de contrôle selon que j'ai le rôle de l'improvisateur ou du compositeur.

5.9. Les contrôleurs en temps réel

Outre la mémorisation des notes jouées par corde, le suivi de l'amplitude, la reconnaissance des vitesses MIDI et le suivi de registre du jeu de guitare qui ont déjà été discutés, différents outils d'analyses ont été mis à contribution afin d'allouer plus de contrôle au guitariste sur le matériel musical généré par *Agône* et de favoriser la dimension interactive. Ces mécanismes comprennent la reconnaissance du nombre de notes simultanées jouées par lui, (information très accessible via l'utilisation du microphone hexaphonique), de la région du manche sur laquelle il joue sur un plan x, y (quelle corde, quelle frette), de son débit d'attaques au médiator en *ms*, du centre de gravité spectral (*centroid*-) afin de déterminer si le guitariste attaque ses cordes au niveau de la touche ou du chevalet ou entre ces deux points, ainsi que la réception des données provenant de 2 pédales d'expressions. Tous ces paramètres sont utilisés en *mapping* multicanal avec les synthétiseurs et échantillonneurs pilotés par le logiciel selon différentes stratégies assignables pour chaque *preset*. Principalement, j'utilise ces informations pour moduler les paramètres des synthétiseurs en temps-réel et ainsi, obtenir une matière sonore plus évolutive et adaptative, la rendant ainsi plus complexe et selon mes critères, plus intéressante. Également, ces possibilités de *mapping* renforcent l'interactivité homme-machine.

5.10. Le déclenchement des clips vidéos

Agône propose un module pour déclencher des clips vidéos associés aux notes jouées par le guitariste ou par les instruments électroacoustiques. Par exemple, pour la improvisation discutée au point 6, environ 300 clips vidéos d'une durée moyenne d'une seconde ont été réalisés au moyen de synthétiseurs visuels numériques ou de photographies travaillées numériquement puis placés dans les séquences d'un logiciel de Vjing communiquant avec *Agône* via MIDI. Les clips sont ensuite partagés entre les différents canaux d'*Agône* (6 pour les synthétiseurs et

échantillonneurs, un 7è dédié au transit des notes jouées à la guitare) et permettent un système de *triggering* multicanal favorisant le développement d'une polyphonie visuelle qui est une idée en développement que je poursuis. Pour conserver l'idée "d'improvisation logicielle", ces clips sont déclenchés dans un ordre aléatoire. Comme je l'expliquait, la vidéo me permet de personnifier les voix du logiciel, de lui conférer une existence visible par le public. Également, dans ma démarche de création, je suis très sensible aux relations images-sons que j'explore dans le cadre de musiques visuelles auxquelles j'ai été sensibilisé par Pr. Jean Piché. Cela était donc important dans ma démarche d'inclure une portée visuelle au langage d'*Agône*.

6. LE PROCESSUS DE COMPROVISATION

L'outil a été développé avec en tête l'idée de laisser la place à la spontanéité de l'improvisateur en modulant certaines de ces variables en temps-réel à différents étages du programme en fonction du jeu de l'instrumentiste et comme il a été expliqué dans la section précédente. Aussi, l'utilisation intensive de tables de probabilités définies par le compositeur permettent une musique provenant des algorithmes au résultat en partie indéterminé. Cependant, j'ai souhaité que cet aspect aléatoire soit confiné dans une zone contrôlée et définie par le compositeur lors de la phase de travail en temps-différé.

Je souhaitais pouvoir prédéterminer une forme musicale et en organiser le contenu. J'ai décidé pour cela de structurer *Agône* de telle sorte qu'il autorise une approche compositionnelle en tableaux, avec l'ensemble de ses variables stockées dans des boîtes à *presets* que l'instrumentiste-improvisateur rappelle sur scène au moyen de pédales de déclenchement. Par contre, le logiciel ne permettant pas encore d'interpoler ses paramètres, il est difficile en l'état de travailler sur des structures évolutives, nous sommes pris avec des formes en tableaux. Également, l'abstraction qu'opère le logiciel est guidée par le paradigme musical mélodico-rythmique. Ce sont là deux limitations qui me gênent dans ma démarche de création et auxquelles les versions futures tenteront de remédier. Pour tenter de les contourner, j'ai recours à l'utilisation de l'écriture et de la pensée issue de la tradition acousmatique qui me permet d'accéder à une fluidité organique d'écriture que le logiciel ne permet pas encore.

Mon parcours de musicien et compositeur m'a amené à concevoir la musique selon les paradigmes en vigueur dans les musiques populaires, classiques et jazz puis plus tard, selon une approche matériologique et une pensée de la rhétorique musicale développée par le courant acousmatique. Le processus de comprovisation de *Denise*, *Agnus & Paula* avec *Agône* a tiré avantage de ses différentes expériences mais m'a demandé d'en adopter une nouvelle pour laquelle j'étais novice, la plus

importante en fait dans ce cas précis, qui m'a au départ dérouteré et qui est l'essence même d'*Agône* et le pain quotidien de bon nombre de compositeurs dont la musique repose sur l'informatique dans le processus de composition ou est orientée par des processus : l'abstraction d'idées musicales représentées au moyen de nombres (ou d'autres formes de représentations). Dans le cas d'*Agône*, comme bien d'autres logiciels, ces nombres sont accessibles par le compositeur au moyen de sliders, boutons, champs numériques, matrices et tables.

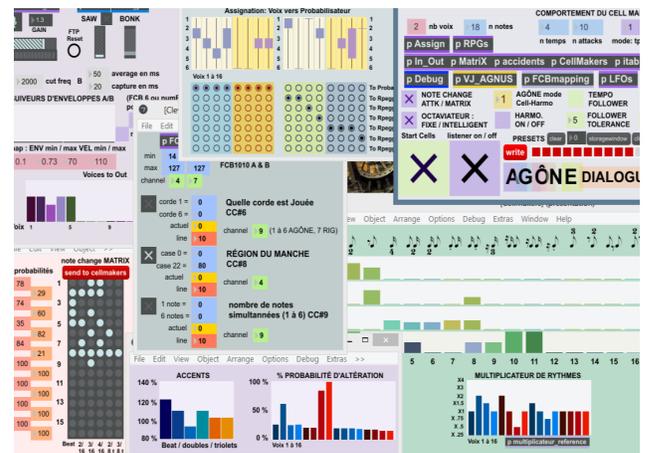


Figure 2. L'essentiel des IU accessibles par le compositeur

Également, la partie de guitare n'est pas écrite, elle est entièrement improvisée et suit des idées musicales que j'articule sous formes de directives assez sommaires. Ces notes spécifient également les interactions entre les paramètres de contrôle provenant de la guitare et leurs corrélations avec les synthétiseurs et autres, et ce par tableau (composition en tableaux).

Je n'ai pas encore assez de recul pour parler d'une méthodologie de comprovisation que j'aurais adopté. Il s'agit plutôt d'un constant va et vient entre l'improvisation sur l'instrument et le paramétrage sur l'ordinateur. Beaucoup d'improvisations que j'ai réalisées pendant le processus de création ont été enregistrées, ce qui me permettait de travailler sur le *sound design* des instruments électroacoustiques et des modélisateurs d'effets de guitare principalement, mais qui étaient également d'un grand intérêt pour définir la forme de la pièce et ses proportions par le décodage des données MIDI et audio enregistrées, de faire la découverte de relations qui m'auraient peut-être échappées en temps-réel et de poser des jalons clairs dans le cheminement des gestes musicaux.

Enfin, tout comme le compositeur de musique acousmatique peut être amené à refaire un tournage sonore en cours de composition pour de multiples raisons, le

développeur-comprovisateur peut-être amené à modifier son logiciel en cours de processus afin qu'il permette la réalisation d'une idée musicale qu'il n'avait pas anticipé et qui justifie selon lui ces allers-retours. Ainsi, j'ai régulièrement eu à modifier mes programmes afin de les adapter à une idée musicale ou tout simplement à en autoriser la réalisation.

7. CONCLUSION

L'enjeu principal de cet article s'intéressait à la question de la nécessité pour un créateur à la recherche d'un langage musical singulier et improvisé de développer ses propres outils. Je n'ai à date, et hormis une étude préliminaire, improvisée qu'une seule œuvre exploitant le logiciel. Je peux cependant déjà affirmer que le développement d'outils a fortement perturbée de façon positive mes conceptions sur la création musicale. Que ce soit au niveau d'une plus grande intégration de l'aléatoire dans mes techniques de composition, d'une liberté accrue dans les allers-venus entre le temps-différé et le temps-réel ou de l'intégration d'éléments interactifs et du multimédia. Je suis également convaincu, comme je l'affirmais précédemment, que l'outil conditionne le résultat. Donc qu'une connaissance intime de l'outil permet au créateur de le modifier selon une idée, elle-même influencée par l'outil, dans une boucle élémentaire de rétroaction à laquelle la création musicale ne fait évidemment pas défaut. *Agône* propose déjà un potentiel de improvisation immense mais je suis également très conscient de ces nombreuses limitations et nourrit beaucoup d'idées pour son développement futur. Un des enjeux majeur selon moi du développeur-créateur est de trouver le juste équilibre entre ces deux casquettes, l'une comme l'autre nécessitant beaucoup d'investissement, il est clair que l'équilibre est fragile.

8. REMERCIEMENTS

Je tiens à remercier Olivier Bélanger pour son enseignement de l'algorithmie musicale et l'intérêt pour cet art qu'il a su me transmettre, Pr. Jean Piché pour son soutien et ses conseils lors du développement des versions alpha d'*Agône* et l'opportunité de créer la première performance expérimentale et improvisée pour guitariste et dysclavier utilisant ce logiciel³, Louis Dufort pour m'avoir permis de produire *Denise Agnus et Paula*, première pièce multimédia exploitant *Agône* et ce, devant le public d'Akousma XI à Montréal, ainsi que Martin Marier pour avoir été une source d'inspiration pour la rédaction de cet article.

³*Pianalgo*, 23 Jan. 2013, Université de Montréal, faculté de musique. <http://vimeo.com/69984446>

9. RÉFÉRENCES

- [1] Karim Barkati. Entre temps réel et temps différé : *Pratiques, techniques et enjeux de l'informatique dans la musique contemporaine*. Thèse de doctorat de l'université de Paris 8. 2009.
 - [2] Richard Dudas. "Comprovisation" : The Various Facets of Composed Improvisation within Interactive Performance Systems. In *Leonardo Music Journal*, December 2010, Vol. 20, pp 29-31, 2010.
 - [3] Jean Geoffroy. Le geste dans l'œuvre musicale, la musique et le mouvement, In *Rencontres musicales pluridisciplinaires : Le Feedback dans la création musicale*, GRAME, Lyon, 2006.
 - [4] Jullian Hoff. Denise, Agnus & Paula : comprovisation interactive. Captation de performance durant Akousma XI, Usine C, Montréal, Nov 2014 <http://vimeo.com/111930919>
 - [5] Andy Hunt and Marcelo M. Wanderley. Mapping performer parameters to synthesis engines. *Organised Sound*, 7, pp 97-108, 2002.
 - [6] Martin Marier. La composition électroacoustique pour interface inventée. In *Actes des Journées d'Informatique Musicale (JIM 2012)*, Mons, Belgique, 9-11 mai 2012.
 - [7] Native Alien. <https://vimeo.com/album/1717549> (hyperlien vérifié le 25 mars 2015)
 - [8] Stéphane Roy. L'analyse des musiques électroacoustiques, modèles et propositions, éd. L'Harmattan, 2004.
 - [9] AtauTanaka. Sensor-Based Musical Instruments and Interactive Music, in *The Oxford Handbook of Computer Music* (Oxford Univ. Press, pp. 233-257, 2009.
 - [10] Todd Winkler. *Composing Interactive Music : Techniques and Ideas Using Max*, MIT Press, 2001.
- Logiciels cités en références**
(hyperliens vérifiés le 25 mars 2015)
- [11] Bloom – Generative music software. <http://www.generativemusic.com/bloom.html>
 - [12] Modalys – Modélisation physique <http://repmus.ircam.fr/omax/home>
 - [13] Noatiki – Generative music composer <http://intermorphic.com/noatiki/>
 - [14] Nodal – Generative music software. <http://www.csse.monash.edu.au/~cema/nodal/>
 - [15] OMax – Outil d'improvisation automatique interactif <http://repmus.ircam.fr/omax/home>