Ecole Nationale
Supérieure
de l'Electronique
et de ses Applications

Lévy Benjamin

Rapport de stage

Stage effectué aux **Studios Puce Muse**
du 19 juin au 6 septembre 2006
sous la direction de Serge de Laubier

Je remercie Serge de Laubier de m’avoir accueilli dans les Studios Puce Muse pour ce stage et de m’avoir guidé tout au long de celui-ci. Je lui suis également reconnaissant pour les contacts qu’il a bien voulu me donner.

Merci à Vincent Goudard pour son accueil, ses conseils de programmation et toute la musique qu’il m’a fait découvrir !

Je remercie également Amélie Piron pour son soutien administratif ainsi que Ludovic Becker.

Merci à Emmanuel Fléty d’avoir pris le temps de m’aider à trouver ce stage et à Guillaume Collavizza.

Merci également à Emmanuel Jourdan et Sylvain Reynal pour leurs suggestions à propos de MaxMSP.

Sommaire

Introduction

Depuis longtemps, je cherche à allier musique et informatique. Les études à l'ENSEA me donnent une formation solide d'ingénieur tandis que le Conservatoire National de Région m'accueille pour une formation musicale ; je cherchais donc un stage qui me permette de réunir les deux dans une entreprise de musique assistée par ordinateur. Les Studios Puce Muse m'ont été indiqués par d'anciens élèves ayant fait des stages ou travaillent actuellement dans de telles activités. Intéressés par mon profil, ceux-ci m'ont aussitôt fait une proposition de stage et j'ai rencontré les responsables, Serge de Laubier et Vincent Goudard, peu après.

Le projet sur lequel je pouvais travailler était initialement prévu pour un stage d'une durée de six mois mais Puce Muse a accepté de me laisser en poser les premières pierres en deux mois et demi d'été. Le sujet promettait de me donner toutes les bases que je souhaitais acquérir pour démarrer dans le domaine.

Structure du Studio

Objectifs des Studios Puce Muse

L'activité principale des Studios Puce Muse concerne la musique assistée par ordinateur. Depuis 1982, les Studios ont conçu et fabriqué trois générations de Méta-Instrument (MI) : une interface corporelle destinée à contrôler la création de son et d'image sur un ordinateur par l'intermédiaire du logiciel de programmation graphique Max/MSP/Jitter. Les Studios développent également une interface ludique de création sonore et visuelle collective : la Méta-Malette, contrôlée par plusieurs joysticks. De nature très intuitive, cette dernière vise un public de jeunes enfants, collégiens ou handicapés. Un projet avec le Ministère de l'éducation nationale, de l'enseignement supérieur et de la recherche est sur le point de voir le jour. Les outils nés de ces activités de recherche permettent à Puce Muse de proposer des créations musico-graphiques en temps réel.

Structure administrative

Les Studios Puce Muse constituent légalement une association loi de 1901. Le président et la trésorière ne participent pas aux activités et n'ont qu'un statut légal. Directeur artistique et fondateur de Puce Muse en 1981, Serge de Laubier a le statut d'intermittent du spectacle et est rémunéré au cachet, de même que Pierre Galais, responsable de régie lors des concerts. Les Studios comptent trois salariés : Amélie Piron, employée à mi-temps pour la gestion administrative, Ludovic Becker, chargé de la diffusion des productions, à mi-temps également, et Vincent Goudard, programmeur à plein temps et responsable de la vente du Méta-Instrument.

Organisation matérielle

Situés dans la zone industrielle Silic à Rungis, les locaux sont divisés en trois parties correspondant aux activités des Studios. La pièce centrale, consacrée à la création sonore, dispose d'un système audio octophonique constitué d'une table de mixage (48 pistes), d'un ordinateur Macintosh G4 et de nombreux autres appareils dédiés (synthétiseurs, enregistreurs...).

Partenaires

Aides institutionnelles :

- Ministère de la Culture et de la communication et DRAC Ile de France
- Conseil régional d'Ile de France

Aides spécifiques :

- SACEM
- Le DICREAM[1] a soutenu *La Volière Puce Muse, Traverse de Façade et le Concert Immersif.*
- La DMDTS[2] participe au développement de la Méta-Mallette et a soutenu la programmation du Méta-Graphique.
- La Délégation aux Arts Plastiques est intervenue sur la première phase de développement du Méta-Graphique.
- La DRRT[3] et le Conseil Général de l'Essonne ont soutenu la Méta-Mallette en 2004.

Partenariat avec des laboratoires, écoles ou universités :

- LAM : Laboratoire d'Acoustique Musicale, Paris
- LIMSI d'Orsay, Laboratoire d'Informatique pour la Mécanique et les Sciences de l'Ingénieur.
- LABRI, Laboratoire Bordelais de Recherche en Informatique
- IRCAM, l'Institut de Recherche et Coordination Acoustique/Musique
- IUT de Cachan Paris XI : accueil régulier de stagiaires en génie électronique sur la portabilité du Méta-Instrument.
- Le Lycée technique de Corbeil-Essonnes (BTS de productique) auquel sont confiés des travaux spécifiques de mécanique.

Comité Scientifique :

Sa fonction est d'évaluer, de conseiller, d'orienter et de faire rayonner le travail de recherche scientifique de Puce Muse. Il est constitué de : Jean Haury (INPI[4]), Norbert Schnell (IRCAM), Claude Cadoze (ACROE[5]), Emmanuel Favreau (INA GRM[6]), Hugues Genevois (LAM) et Christian Jacquemin (LIMSI).

[1] : Dispositif pour la Création Artistique Multimédia
[2] : Direction de la Musique, de la Danse, du Théâtre et des Spectacles
[3] : Délégation Régionale à la Recherche et à la Technologie
[4] : Institut National de la Propriété Industrielle
[5] : Association pour la Création et la Recherche sur les Outils d'Expression
[6] : Institut National de l'Audiovisuel Groupe de Recherche Musicales

Historique

1982 Fondation des Studios Puce Muse.

1983 Début de la recherche sur la spatialisation du son.

1986 Invention du Processeur Spatial Octophonique (brevet n°8600159).

1987 Début de la recherche sur le Méta-Instrument.

1988 *Puce Muse* : premier concert composé en trois dimensions.

1989 *Puce Muse 3* pour deux Méta-Instruments et deux nacelles automotrices.

1992 Premier concert *Lumière* pour deux Méta-Instruments et filaments incandescents récompensé par le FAUST d'Or.

1993 Orchestre de Sonocannes, concert pour 16 sonocannistes ambulants. Début de la recherche sur la deuxième génération du Méta-Instrument.

1994 *Puce Muse Lux* : concert pour mille sources lumineuses et sons électroniques. Grand Prix du Festival International Multimédia de Locarno.

1995 *Au Vif de la Mémoire* de Bernard Parmegiani pour Méta-Instrument et bande.

1996 *Les Sargasses de Babylone* pour les nouveaux Méta-Instruments. Nomination aux Janus du Design.

1997 *Cyclone* de Gyorgy Kurtag. Début de la recherche sur l'image numérique calculée en temps réel : le Méta-Graphique.

1998 *Puce Muse Black & Or* pour deux Méta-Instruments, vingt projecteurs robotisés télé-pilotés et musique octophonique. Enseignement du Méta-Instrument au Conservatoire de Dieppe.

1999 *Puce Muse Nuit* : concert graphique. *Monumental Puce Muse* : concert graphique sur mesure pour façades ou monuments ; 65 concerts dans l'année en France et à l'étranger.

2000 Orchestre National de Sonocannes pour dix échassiers musiciens professionnels. 44000 spectateurs assistent aux concerts durant l'année.

2001 *Satisfecit 2001*. Prix spécial du Jury pour le logiciel Méta-Graphique.

2002 *La Belle Porte le Voile, Oizoo* et *M. Teste.*

2003 *La Volière Puce Muse, Remix Puce Muse* et *Portes paroles.*

2004 *Traverse de Façade* et *Concert Interactif*. Participation aux spectacles *Autour d'Ulysse* de Pierre Sauvageot et au *Live Computer Music* de Daniel Petitjean.

2005 *Les Graphiphonistes, Puce Muse et les 40 Souffleurs*. Musique du spectacle *Geo 9h35* du Free Théâtre. Création de logiciels musicaux et graphiques pour l'*O10C* de Pierre Sauvageot. Création pour *Le Jardin des miracles* de Jean Louis Heckel. A l'occasion de *Cité Rêvée* à Montbéliard, création d'un aquarium géant, installation monumentale quadriphonique pour vingt joysticks audiographiques et un chef. Invitation aux *Jeux Méditerranéens d'Almeria* à Washington dans le cadre de l'année mondiale de la physique.

Projet 2PIM

Le but du stage s'insère dans un projet plus global mené par Puce Muse et soutenu par le Ministère de l'éducation nationale, de l'enseignement supérieur et de la recherche. Il s'agit du développement d'une **Plate-forme de Programmation Interactive Multimodale autour du Méta-Instrument 3 et de cinq projets pilotes associés**. Sept structures associent leurs compétences sur ce projet :

- Puce Muse comme concepteur et utilisateur du Méta-Instrument et pour son travail sur la Musique Vivante Visuelle Virtuelle,
- le LAM (Paris VI - CNRS) pour sa connaissance en lutherie, en psychologie perceptive et ses capacités d'évaluation,
- le LIMSI (Paris XI - CNRS) pour ses travaux en synthèse vocale et en réalité virtuelle audio et visuelle,
- le LaBRI (Bordeaux I - ENSEIRB - CNRS) pour sa compétence en modélisation pour la composition musicale,
- l'Université MacGill de Montréal pour sa recherche sur le geste musical et en tant qu'utilisateur du Méta-Instrument 3 (MI3),
- l'IRCAM pour ses compétences sur le traitement du geste et sur la synthèse audio et
- la Grande Fabrique comme utilisateur averti et développeur d'instruments logiciels pour le MI3.

Le projet s'articule en deux temps :

- **Développement de la Plate-forme de Programmation Interactive Multimodale :**

Il s'agit de permettre le co-développement et l'échange de projets de recherche utilisant le MI3 ; il est donc nécessaire de définir, normaliser et développer une Plate-forme de Programmation Interactive Multimodale (2PIM). Cette plate-forme est destinée à héberger des projets d'instruments logiciels audios et visuels prototypes, fonctionnant en temps réel dans un environnement 3D.

- **Exploration des relations entre geste, mouvement, son et instrument dans cinq projets pilotes, utilisant le MI3 et la 2PIM.**

Le premier objectif de cette partie est de développer et d'évaluer cinq instruments de musique virtuels visuels dirigés par le MI3, sur la 2PIM et de mettre en évidence les interrelations entre mouvement gestuel, mouvement visuel et synthèse sonore. C'est ensuite la rédaction d'un programme de synthèse vocale contrôlée par le MI3 sur la 2PIM. Le contrôle gestuel de la synthèse est un enjeu important tant pour la production musicale que pour la recherche en analyse et synthèse de parole expressive. C'est aussi présenter un modèle de composition musicale basée sur une représentation hiérarchique liant micro- et macro-structures musicales et développer un environnement de haut niveau insérable dans 2PIM pour connecter facilement le MI3 à des logiciels "grand public" (plugins VST et VSTI™, Live™, Reason™). La finalité étant de porter la banque d'instruments logiciels utilisés par les générations de MI sur 2PIM et de documenter ce patrimoine musical.

Projet de Stage

Intitulé:

"Intégration des oeuvres musicales composées pour le Méta-Instrument depuis sa création, et de ses périphériques hardware dans un environnement logiciel normalisé."

Domaines:

Traitement du signal et synthèse audio/graphique, organisation de ressources musicales.

Contexte:

Le Méta-Instrument est une interface instrumentale, pensée pour le contrôle temps-réel de la synthèse sonore et graphique pour la création musicale. Depuis sa création en 1987, 3 générations de Méta-Instruments ont vu le jour; la dernière permettant de transmettre à un ordinateur par liaison ethernet ou WIFI, les données de 54 capteurs, mesurant toutes les 2ms la pression et la position des mains et bras du "méta-instrumentiste".

Problème 1:

Plus d'une centaine d'oeuvres ont été réalisées pour le M-I, par des compositeurs, musiciens, artistes multimédias... Malgré les évolutions, les oeuvres crées pour lui restent potentiellement jouables à ce jour, mais sont dispersées dans divers centres de création artistique.

Problème 2:

D'autre part, la puissance des machines ayant considérablement augmenté, les périphériques autrefois nécessaires pour assurer la synthèse de son, d'images, le mixage (sampler, table de mixage, interface MIDI... etc.) ne sont plus indispensable et donne une lourdeur au dispositif, qui ne se justifie plus.

Travail à effectuer:

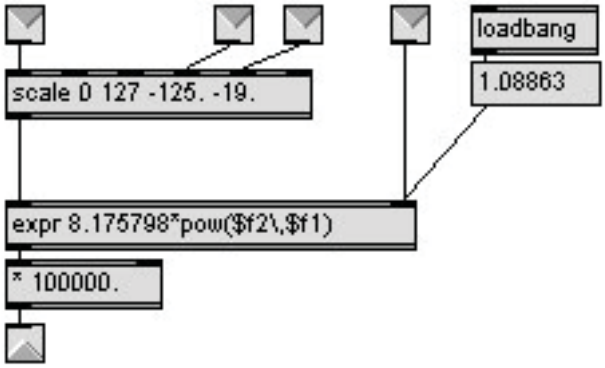
- Dans un premier temps, il s'agit de comprendre le fonctionnement de l'EMU e-6400 et d'en réaliser une émulation/adaptation. L'EMU e-6400 est un synthétiseur/échantillonneur modulaire, avec de nombreuses possibilités de combinaisons d'oscillateurs, d'enveloppes, et de filtres.
- Recueillir les oeuvres (son + image + programmation) composées pour le Méta-Instrument depuis sa création.
- Les compiler dans une bibliothèque "Répertoire" accessible au Méta-Instrumentistes.
- Analyser les parties communes de la programmation des divers instrument pour les factoriser, et en normaliser les accès.

Cette dynamique globale représente un chantier très vaste qui nécessite de toute évidence plus de 6 mois de travail; il s'agit donc dans le cadre du stage, d'en poser les bases, d'en récolter les éléments, et le documenter exhaustivement pour qu'il puisse être continué par la suite. Le contenu qui touche à des domaines assez divers depuis le relationnel, jusqu'à la programmation d'algorithmes complexes, pourra se focaliser uniquement sur l'émulation de l'EMU e6400 (programmation / traitement du signal), ou bien sur la conception de la bibliothèque répertoire (orientation programmation / gestion), selon la spécialité ou les motivations de l'étudiant-stagiaire.

Environnement de travail

Max/MSP/Jitter

Toute la programmation graphique et sonore s'effectue dans un environnement logiciel constitué de trois programmes quasiment indissociables Max, MSP et Jitter. Tous trois sont développés principalement par l'IRCAM et produits par Cycling74. Il s'agit d'un langage de programmation graphique ; Max gère les instructions, les messages et les signaux MIDI tandis que MSP exécute le traitement des signaux audios ; Jitter concerne la partie vidéo. La programmation se rédige en connectant des boîtes fonctionnelles nativement programmées en langage C sur une feuille de travail appelée "*patch*". Parmi la grande variété de boîtes fonctionnelles, les trois types principaux sont les suivants : les **objets**, dotés d'entrées et de sorties spécifiques, réalisent une fonction dont les arguments sont les paramètres d'entrée ; les **messages** permettent de contrôler les objets en leur donnant des instructions et les **nombres**, flottants ou entiers, utiles à toute programmation.



Le "*patch*" ci-contre effectue par exemple une conversion entre l'échelle MIDI (entiers de 0 à 127) et une échelle de temps non linéaire allant environ de quinze millisecondes à une minute trente.

L'objet "*scale*" effectue une conversion linéaire de l'intervalle [0;127] sur l'intervalle [-125.;-19.] puis l'objet "*expr*" applique à son entrée l'expression mathématique rédigée en langage C. La sortie est multipliée par 100 000 par l'objet "** 100 000*".

E-MU 6400 Emulator IV

Un des outils externes à tout ordinateur utilisé avec le MI est un synthétiseur numérique de chez E-MU nommé Emulator IV sorti en 1994. Son architecture reproduit celle d'un synthétiseur analogique modulaire. A partir d'échantillons audio disposés sur toute l'échelle des notes de musique, l'Emulator IV est capable de jouer 128 voix simultanément. Chaque voix peut être modulée par des enveloppes temporelles et des effets de type oscillateur basse fréquence (LFO), filtres... Cet appareil est conçu autour de circuits électroniques spécifiquement développés et dédiés au traitement audio, ce qui lui confère une grande puissance pour ces traitements. De même que tout le reste de l'installation sonore, l'E-MU fonctionne en octophonique (8 sorties son distinctes).

Déroulement du stage

Au cours des deux mois et demi de stage, le travail s'est organisé en quatre parties successives.

Prise en main des outils

Les deux premières semaines du stage ont été consacrées à l'apprentissage de la programmation dans Max/MSP à l'aide de la documentation et des exemples en anglais fournis par le développeur. Une soixantaine de tutoriaux permettent de comprendre les fonctions de Max, des plus basiques telles que la transmission et la gestion des nombres aux plus avancées telles que le *scripting* (automatisation de la création d'objets). Une trentaine d'autres tutoriaux, également en anglais, aident de même pour MSP.

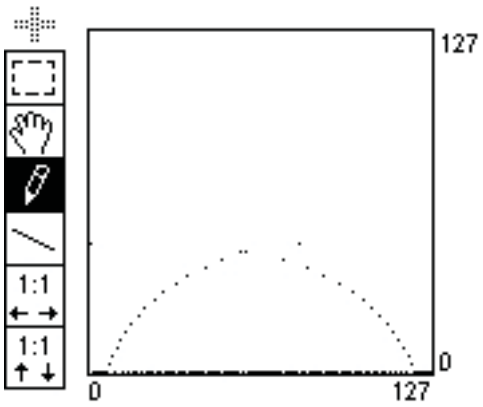
En vue d'utiliser le MI (génération 2) comme interface de contrôle du synthétiseur émulé, des recherches en bibliothèque sur la structure et l'utilisation du protocole MIDI (Musical Instruments Digital Interface) ont été nécessaires. Ce protocole codé sur un ou deux octets est en effet largement utilisé pour la création musicale assistée par ordinateur. Il permet de déclencher des notes, contrôler certains de leurs paramètres, les éteindre, choisir le timbre de l'instrument utilisé... Il comporte trois principaux types de messages : les messages de déclenchement, "*note on*", "*note off*", "*sustain*" etc, les messages de changement de configuration, "*program change*" qui contrôlent en général le timbre instrumental utilisé et les messages de contrôle en temps réel "*control change*".

En parallèle de ce travail, l'objectif d'émulation nécessitait d'appréhender et de comprendre en profondeur le fonctionnement et les caractéristiques de l'Emulator IV. Celui-ci possède une architecture à quatre niveaux.

En bas de l'architecture, l'échantillon audio "*sample*" contient, en plus des données audio, des paramètres de boucle, de hauteur, ainsi que d'autres réglages de lecture. Un ou plusieurs "*samples*" sont réunis dans une voix, "*voice*". A chaque voix sont associées trois enveloppes temporelles : une pour le volume, une pour le filtre, la troisième attribuable par l'utilisateur ; deux oscillateurs basse fréquence et un filtre paramétrique. D'autres réglages sont également disponibles à ce niveau : il est notamment possible, grâce à un tableau de correspondance, d'attribuer chacun des enveloppes, LFO et paramètres MIDI à un contrôle sur le son (contrôle de hauteur : "*pitch*", de vélocité...) ou même de les associer pour une configuration plus complexe.

Le niveau supérieur se nomme "*preset*", il rassemble plusieurs voix interchangeables au cours du jeu instrumental sans discontinuité. Ces changements peuvent être contrôlés par des commandes MIDI en temps réel. Enfin une "*bank*" est une collection de "*preset*", chacun étant assigné à un canal de message MIDI. Il est donc possible de changer de "*preset*" par un message MIDI de type "*program change*".

La principale difficulté que présente l'Emulator IV, en dehors de sa structure élaborée, réside dans la non linéarité des échelles utilisées. A l'instar de l'oreille, toutes les grandeurs majeures de l'Emulator IV sont calculées exponentiellement : volume en décibels, fréquences en octaves, temps logarithmique, répartition spatiale à valeur efficace au carré constante.



Ci-contre le tracé d'un fondu-enchaîné, "*crossfade*", entre deux échantillons audio. En abscisse, la valeur (de 0 à 127) du paramètre MIDI contrôlant cet effet ; en ordonnée, le volume audio de chacun des échantillons en décibels ramené de 0 à +48.

Travail sur les enveloppes et autres fonctions

Les enveloppes de l'Emulator IV constituent un autre point fort de cet appareil. En effet, toutes les enveloppes présentes (trois par voix) peuvent être contrôlées en temps réel au cours du déroulement de la note. Ce sont des enveloppes à six segments, deux segments pour chaque phase de la note : attaque, déclin, relâchement. Dans Max/MSP, seul le signal audio échantillonné au minimum à 44,1 kHz réagit suffisamment rapidement pour permettre un contrôle aussi précis que dans l'Emulator IV. Cependant, tous les outils préexistant dans le logiciel pour créer des enveloppes temporelles fonctionnent en message logique de type Max, beaucoup plus lent, et non en signal audio. Environ deux semaines de recherches en programmation ont donc été nécessaires pour mettre au point une enveloppe temporelle fonctionnelle en signal. Pour y parvenir, l'aide de programmeurs travaillant à l'IRCAM a été sollicitée. Ceux-ci, loin de fournir solution au problème, ont cependant suggéré des pistes de recherche poursuivies par la suite sur Internet grâce à une communauté active et des forums anglophones d'utilisateurs de Max/MSP. Ci-joint en annexe le "*patch*" obtenu, premier aboutissement du stage, accompagné de quelques explications.

Par la suite, sept autres fonctions de l'Emulator IV ont été portées dans Max/MS :

- Un oscillateur basse fréquence (Low Frequency Oscillator). Il possède les cinq paramètres présents sur le synthétiseur externe : forme d'onde, fréquence, synchronisation avec la note, délai de déclenchement, variable aléatoire sur la fréquence.

- Un filtre passe-bas du quatrième ordre à fréquence de coupure et coefficent de qualité variable. Les filtres présents nativement dans MSP n'allant que jusqu'au deuxième ordre, la mise en cascade de deux filtres du deuxième ordre a été la solution retenue.

- Un panoramique entre deux sorties audio. Pour garder le volume apparent invariable, la somme des valeurs efficaces au carré devait rester constante. Cet équilibre entre droite et gauche est paramétrable en fonction de l'angle et de la distance entre les deux sources sonores, ainsi que de la distance de l'auditeur aux sources.

- Une fonction de variation de ton au cours du jeu avec l'instrument, "*pitch bend*", contrôlée en MIDI.

- Une fonction permettant de raccorder chacune des sorties des différents éléments précédemment mentionnés à tout paramètre d'entrée signal ou messages. Celle-ci tente de reproduire les possibilités infinies de connexions entre les différents éléments que gère l'Emulator IV. Elle possède en outre une adaptation d'échelle (linéaire) pour contrôler au mieux chacune des connections.

- Une fonction de génération de nombres aléatoires dans un intervalle donné. Comme dans l'Emulator IV, l'aléatoire introduit une dimension très intéressante en musique et création visuelle.

- Plusieurs lecteurs d'échantillon audio. Deux types différents de lecteurs étaient nécessaires pour approcher les possibilités du synthétiseur modèle. En effet, celui-ci permet non seulement la lecture de plusieurs échantillons simultanément mais aussi la répartition des échantillons selon la hauteur du son désirée, la vélocité de la note et un paramètre MIDI choisi par l'utilisateur. Un lecteur basique, pour un seul échantillon, est donc utilisé dans Max/MSP lorsque cela est possible alors qu'un lecteur plus complexe, possédant sa propre mémoire et une échelle interne de répartition des échantillons, a été programmé pour les réalisations plus délicates.

Construction de la structure et rappel de configurations

Devenu opérationnel, le synthétiseur créé par l'assemblage des différentes fonctions développées a nécessité un important travail de structuration et d'organisation. L'objectif étant de livrer un ensemble de "*patchs*" fonctionnels et pratiques d'utilisation, une réflexion sérieuse sur la manière d'emboîter les éléments les uns dans les autres a été entreprise. Les solutions retenues devaient allier la lisibilité de la structure pour l'utilisateur et la légèreté d'utilisation tout en conservant une grande étendue de possibilités de construction d'un instrument virtuel. Une nomenclature a donc été développée : les fonctions mentionnées ci-dessus doivent être connectées entre elles par l'utilisateur pour créer un *instrument* virtuel à base d'échantillons audio. Un modèle standard pour tous les instruments servant de base de construction a été élaboré et inclus dans un "*patch*" principal, lui aussi à l'état de modèle. Le "*patch*" principal gère les différents instruments construits ainsi que les chemins empruntés par les signaux MIDI. Il permet également aux différents instruments d'être joués en polyphonie. Pour l'utilisateur il ne reste alors qu'à organiser et construire ses propres instruments à partir du modèle et à choisir le nombre de voix de polyphonie et les contrôles MIDI qu'il souhaite utiliser. Bien entendu, l'instrument modèle a été conçu en octophonique dans la tradition des Studios Puce Muse !

Le synthétiseur virtuel devait permettre également de gérer - c'est à dire créer, mémoriser et rappeler par la suite - les différentes configurations que l'utilisateur aura inventées pour ses instruments. La question de la mémoire des paramètres s'est donc imposée. Les seules choses non mémorisables (bien qu'automatisables) dans Max/MSP sont les connexions entres les différentes boîtes fonctionnelles. La solution du modèle de création trouve ici son intérêt ; il est modifié dans sa structure par l'utilisateur lors de l'invention d'un nouvel instrument qui est alors enregistré dans un nouveau fichier, de sorte que le modèle est conservé intact pour une utilisation ultérieure. Les connexions sont donc enregistrées dans le nouvel instrument. Seuls les paramètres (concernant les nombres choisis, principalement) sont à mémoriser par Max/MSP dans un système de sauvegarde interne. La mémorisation interne reste cependant atteignable depuis l'extérieur puisque qu'elle est stockée dans des fichiers au format *xml*, modifiables par un éditeur de texte.

Réalisation de l'interface et documentation

Le dernier point abordé à la fin du stage a été la réalisation d'une interface graphique pratique et esthétique pour l'ensemble des fonctions du synthétiseur. L'utilisateur devait pouvoir, depuis le "*patch*" principal, intervenir directement sur les réglages d'enveloppe, de LFO... Il fallait donc renvoyer et actualiser les différents paramètres de l'instrument à l'interface globale. Un système automatisé de création d'interface a donc été nécessaire. Des problèmes d'interférences entres les noms de variables ont dû être résolus.

Le but du travail, situé bien au-delà de ce stage, a impliqué une réelle précision et clarté de formulation des objectifs atteints et à atteindre. Ayant posé les bases de ce synthétiseur virtuel il a été nécessaire de définir et justifier les solutions, choix ou organisations développés. La documentation du travail réalisé en vue d'une poursuite ultérieure a donc été la conclusion indispensable de ce stage.

Résultats

Synthétiseur modulaire virtuel

Architecturé selon les caractéristiques précédemment mentionnées, le synthétiseur modulaire virtuel obtenu permet de nombreuses configurations au choix de l'utilisateur. Il possède cinq enveloppes contrôlables en temps réel dans la version atteinte en fin de stage : une enveloppe de volume variant de 0. à 1. et quatre enveloppes identiques variant de -1. à 1. , deux servant aux filtres et deux libres d'utilisation. Deux filtres passe-bas du quatrième ordre sont disponibles ainsi que deux LFO. En plus de ces fonction principales, deux panoramiques, un variateur de ton et un générateur aléatoire peuvent être connectés grâce à quatre fonctions de raccordement. Quatre lecteurs d'échantillons, deux de chaque modèle précédemment expliqués créent le son ; la répartition entre les huit sorties est également prise en charge.

Seule la puissance de l'ordinateur utilisé limite le nombre de voix de polyphonie ainsi que le nombre d'instruments simultanément joués. Tous les paramètres disponibles dans le synthétiseur sont assignables à des signaux MIDI au gré de l'imagination de l'utilisateur.

Deux exemples d'instruments

Au fur et à mesure de la construction de ce synthétiseur, deux instruments déjà existants sur l'Emulator IV ont été portés dans Max/MSP. L'intégralité de ces deux instruments est fonctionnelle et utilisable sur l'ordinateur. La rapidité et la facilité de construction de ces instruments ainsi que le résultat sonore ont été comparés à ceux du synthétiseur externe. Les résultats atteints ont été très satisfaisants en terme de rapidité et de résultat sonore. La lisibilité de construction et les performances de l'ordinateur restent des points à améliorer.

Le premier instrument est constitué de trois parties : une partie de flûte, une partie de percussion et une basse d'orgue. Le bras droit (angle horizontal) du MI contrôle la valeur des notes : noires, croches, doubles croches. L'index droit déclenche les sons de flûte, le majeur droit les percussions. La hauteur des sons est modulée sur des arpèges dont l'étendue est contrôlée par la pédale gauche et la note de base par l'angle vertical du bras droit. Le pouce droit agit comme une pédale de soutien et des paramètres aléatoires sont introduits par les autres doigts de la main droite. La main gauche, quant à elle, contrôle la basse d'orgue. Enfin, un volume général se trouve dans la pédale de droite.

Le second instrument possède deux parties : une partie de flûte, une partie de guitare. Le principe de base est le suivant : un groupe de notes est auto-entretenu (une note se déclenche dès qu'une autre se finit) pour chacune des parties. La partie de flûte est contrôlée par le bras droit, la guitare par le bras gauche de manière symétrique. Le nombre de notes dans le groupe est contrôlable pour chacun des instruments sous les index respectifs. La hauteur et la vitesse de renouvellement des notes sont régies par les angles verticaux et horizontaux des bras et différents paramètres aléatoires agissant sur la hauteur des notes sont disponibles sous les autres doigts.

Poursuite du projet

L'émulation de l'E-MU Emulator IV a été bien avancée mais de nombreuses améliorations ont été discutées. L'interface graphique de contrôle peut par exemple être étendue à d'autres paramètres en plus de ceux déjà présents. Un système intégré pour chaque échantillon devrait remplacer judicieusement le système de lecture de plusieurs échantillons. La gestion des lecteurs eux-mêmes pourrait être améliorée... Cette émulation est donc à poursuivre.

Le résultat obtenu au bout des deux mois et demi de ce stage ont été appréciés par Serge de Laubier, maître de stage. Cependant le projet proposé initialement pour un stage de six mois est loin d'être fini. Seul le premier point du travail à faire a été presque entièrement réalisé et reste à consolider. Malgré une vitesse d'avancement au cours du stage jugée bonne, trois mois supplémentaires minimum sont nécessaires pour mener à bien l'intégralité du projet.

Satisfait du déroulement du stage et du travail fourni Serge de Laubier envisage de m'associer à la poursuite de ce projet durant l'année 2006-2007 (quelques heures hebdomadaires).

Bilan personnel

J'ai eu la chance de trouver un stage dans ce champ qui m'intéresse particulièrement. Ayant comme projet à long terme de travailler dans la musique assistée par ordinateur, cette expérience m'a avant tout permis d'acquérir de solides bases - tant théoriques que pratiques - dans ce domaine.

D'une part j'ai pu apprendre à utiliser le logiciel Max/MSP très largement répandu dans le traitement audio en temps réel. C'est un logiciel aux abords assez difficiles puisqu'il s'agit d'un véritable langage de programmation graphique. Une immersion suffisamment longue dans cet outils est donc nécessaire pour en maîtriser l'utilisation. Ce stage a donc été, en premier lieu, une excellente formation logicielle.

L'étude de la synthèse sonore réalisée au fur et à mesure de la découverte et de la programmation de l'Emulator IV a également été fort intéressante. J'ai pu y appliquer des notions enseignées à l'ENSEA, notamment d'informatique et de systèmes échantillonnés.

D'autre part l'environnement professionnel dans lequel j'ai évolué m'a beaucoup apporté sur la gestion d'un projet destiné à être pris en main et poursuivi par une autre personne. Posant les toutes premières bases de ce synthétiseur virtuel, j'ai dû apprendre à envisager de nombreuses solutions logicielles différentes pour une même fonction, à les justifier, les modifier ou les éliminer pour en discuter par la suite avec le programmeur ou mon maître de stage.

De plus, ce premier pas dans le monde professionnel de l'électroacoustique m'a permis de prendre de plus nombreux contacts dans ce domaine. Non seulement Serge de Laubier m'a indiqué aimablement d'autres portes où frapper, mais j'ai également eu besoin pour résoudre certains problèmes de contacter d'autres programmeurs à l'IRCAM par exemple ou dans d'autres structures.

Enfin ce stage dans une structure de type association loi de 1901, développant des produits artistiques et techniques, m'a montré des voies pour réaliser des projets musicaux alliant originalement la pratique instrumentale et des outils informatiques évolués. En tant que trésorier de l'Association d'Informatique Musicale de l'ENSEA, de nombreuses perspectives de créations et de productions se sont ouvertes. Cette expérience me pousse également à développer des projets personnels et à continuer à mener parallèlement le cursus de l'ENSEA et celui du Conservatoire National de Région.

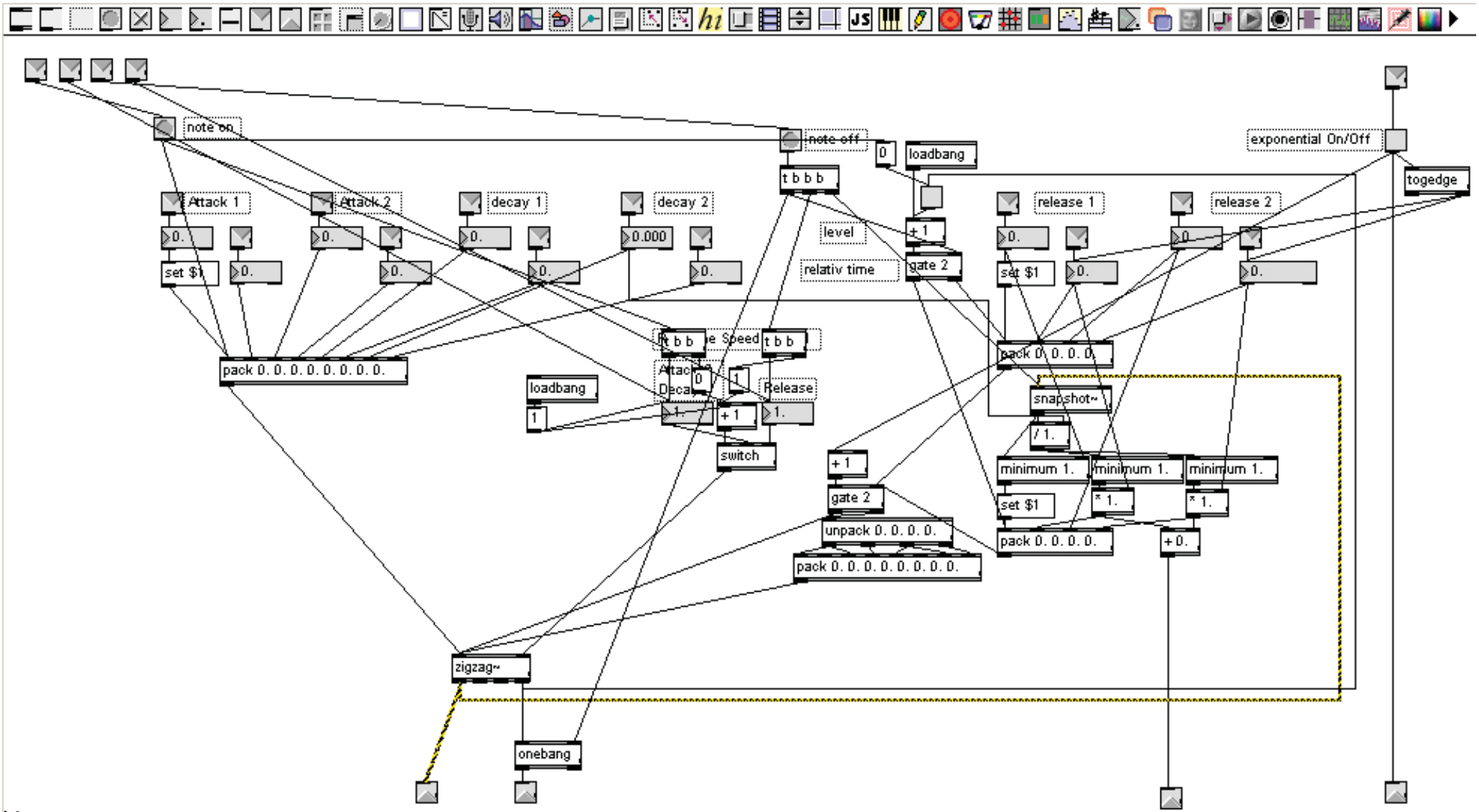
Je suis très content de l'ensemble des aspects de ce premier stage. Faute de temps je n'ai pu résoudre tous les problèmes posés ni approfondir suffisamment certains points restés en suspens ; mais l'éventuelle proposition de poursuite de ce travail au cours de l'année me laisse l'espoir de continuer à contribuer à l'aboutissement du projet.

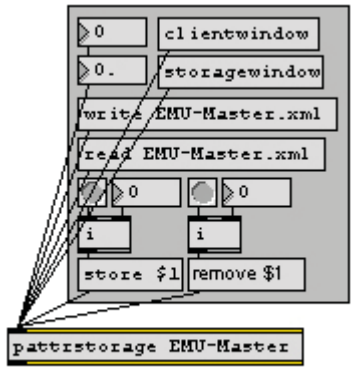
Annexes

Le Méta-Instrument 3

Max/MSP/Jitter

“*Patch*” d’enveloppe





Ce pattrstorage Master (à renommer selon votre jeu) contient les preset à charger dans chacune des instances d'instrument et les chemin de fichier échantillons à fournir aux buffers.

Ces sous patch contiennent les chemins de fichiers échantillons ainsi que les buffers associés. Dupliquez bufferEMU-Inst et renommez-le pour charger les échantillons de chaque instrument

p bufferLFO

p bufferEMU-Inst

Chaque poly~ représente un instrument joué (par défaut EMU-Inst, à remplacer par vos propres instruments) en n voix de polyphonie. L'argument à donner au poly~ l'identifie. On peut ainsi ouvrir plusieurs poly~ du même patch instrument mais charger des preset différents. L'argument du poly~ est à reporter dans le textedit de la GUI pour modifier toutes les instances du poly~ correspondant. Dupliquer les poly~ et leurs outils à volonté.

Identifie le poly~dont vous modifiez les instances

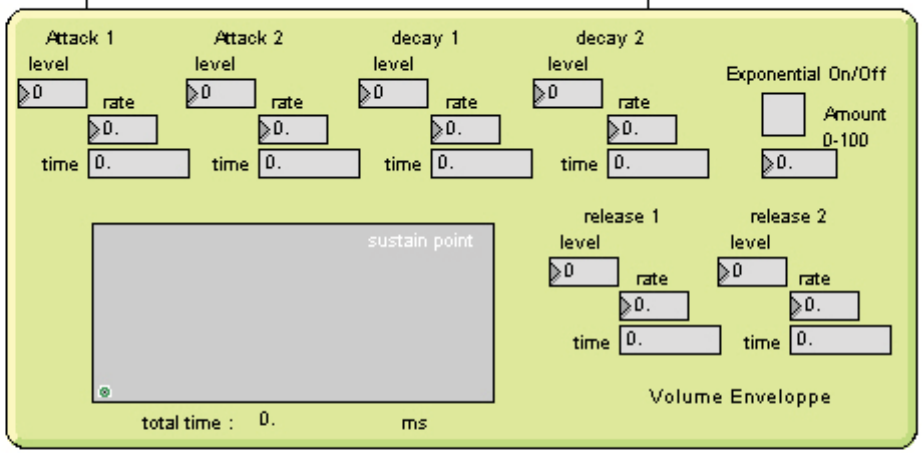
Inst1

Selectionne le groupe de paramètres sur lesquel vous intervenez.

Amp Enveloppe

Petit utilitaire externe pour caler les échantillons et leurs paramètres.

load CalesSon

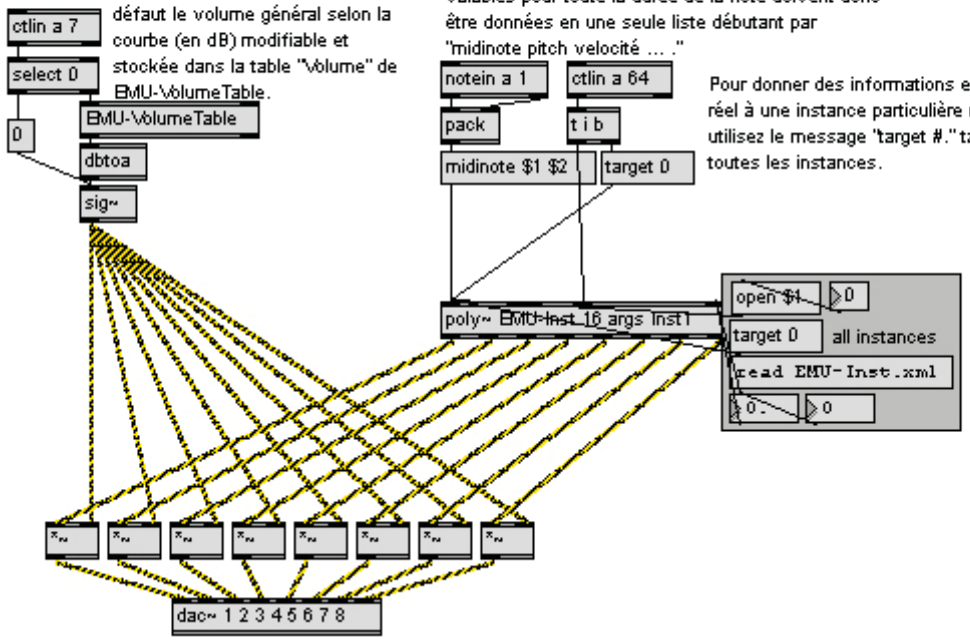


Toutes les fonctions présentes dans le modèle EMU-Inst apparaissent dans la GUI qu'elles soient utilisées ou non dans votre instrument. Si vous ajoutez des fonctions dans votre instrument elles n'apparaitront pas automotiquement, vous devrez modifier EMU-GUI.

Le controler midi 7 contrôle par défaut le volume général selon la courbe (en dB) modifiable et stockée dans la table "Volume" de EMU-VolumeTable.

Le message "midinote" suivi d'arguments permet de gerer les voix de polyphonie (cf poly~.help). Toutes les informations spécifiques à une seule note et valables pour toute la durée de la note doivent donc être données en une seule liste débutant par "midinote pitch velocité ... ."

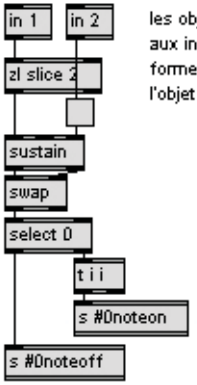
Pour donner des informations en temps réel à une instance particulière (une note) utilisez le message "target #." target 0 = toutes les instances.



Permet de contrôler les pattrstorage de toutes "(target 0)" les instances du poly~.

"Patch" Principal

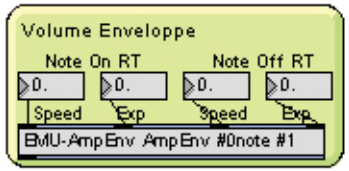
"Patch" Instrument

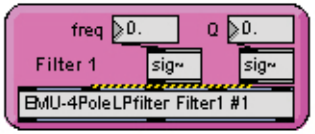


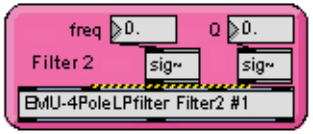
les objets in(~) et out(~) peuvent être dupliqués ou ajoutés. Attention toutefois aux numéro et aux infos. Les informations spécifique à une note doivent obligatoirement être fournies sous forme de liste par l'entrée in 1 en commençant par "midinote" puis pitch et velocité pour que l'objet poly~ répartisse les voix de polyphonie. cf "Master"

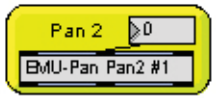
Gardez uniquement les boîtes fonctionnelles dont vous avez besoin. Supprimez les autres. Vous pouvez également les dupliquer, vous avez alors à changer le premier argument et le nom (cmd + ') de l'EMU-Objet pour les voir apparaître proprement dans le pattrstorage. Cependant, une boîte ajoutée ne figurera pas dans la GUI du "Master" à moins que vous ne modifiez celle-ci.

Le second argument des EMU-objets ne doit pas être changé, il est utilisé pour envoyer des note-on et note-off uniques. De même le troisième est utilisé pour l'acces aux paramètres par les "sws" de la GUI.

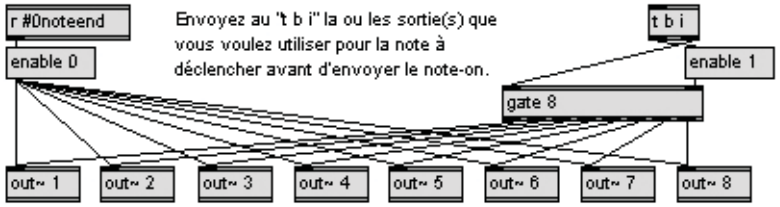




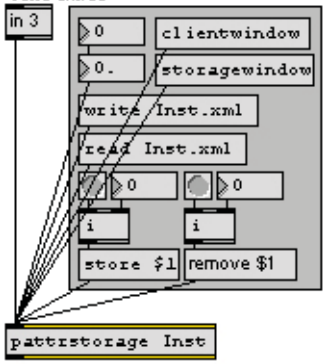




Tous les paramètres apparaissant ici sont modifiables en temps réel. Vous devez pour cela, soit les passer dans la liste "midinote" dans l'entrée "in 1" lors du déclenchement de la note, soit créer une entrée supplémentaire.

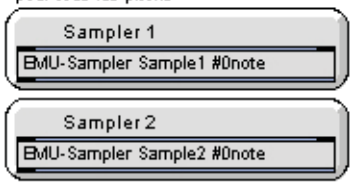


Vous pouvez contrôler le système pattr de toutes les instances d'un même instrument dans le "Master" par cette entrée

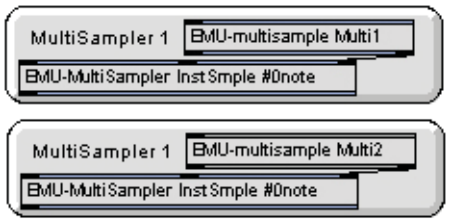


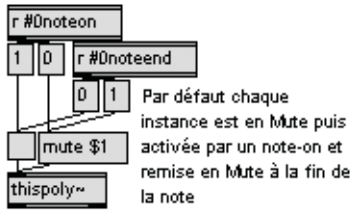
Il est préférable de donner à ce pattrstorage le nom de votre instrument pour le reconnaitre dans le pattrstorage du Master

Utilisez ces sampler simples si vous lisez un (ou les) même échantillons pour tous les pitchs



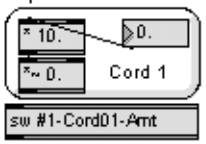
Utilisez ces multisamplers si vous voulez répartir des échantillons selon les pitchs

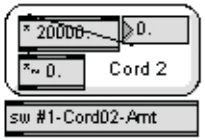




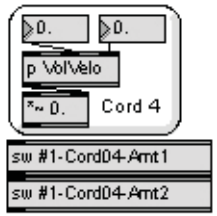
Par défaut chaque instance est en Mute puis activée par un note-on et remise en Mute à la fin de la note

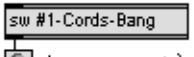
Utilisez les cords 1 & 2 pour relier ce que vous voulez



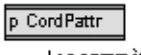


Les cords 3 & 4 sont plus spécifiquement dédiées à l'adaptation du volume en fonction de la vélocité.





Les sw servent à modifier ces paramètres depuis la GUI. Leur nom sera donné par l'argument donné au poly~ du "Master". Pour tous les autres paramètres, les sw sont contenus dans les EMU-Objets. Les Bang servent à actualiser la GUI dès son ouverture.



Les paramètres des cords sont stockés dans ce sous-patch

Si vous rajoutez des cords, pensez à changer les sw correspondant et à ajouter les pattr necessaire dans les sous-patch ci-dessus. Vous aurez besoin de les ajouter également à la GUI.