

MULTICURVE : UN NOUVEL OBJET GRAPHIQUE DE GESTION DE DONNEES DE CONTROLE DANS MAX/MSP.

Jean-François Baud

CICM-Centre de recherche Informatique et
Création Musicale
Maison des Sciences de l'Homme Paris Nord
Université de Paris8
cicm@univ-paris8.fr

RÉSUMÉ

Dans cet article, nous décrivons un objet Max/MSP proposant de nouvelles stratégies de visualisation, de stockage, d'édition et de "lecture" de données de contrôle. Ce développement logiciel s'appuie sur une recherche et des travaux concernant l'interfaçage du corps humain avec un environnement interactif temps réel. Cet objet trouve sa place dans une démarche interdisciplinaire de la sphère Art, Science, Technologie, visant à implémenter certains outils scientifiques sur une plateforme logicielle utilisable sur le terrain des Arts interactifs, telle que Max/MSP. Notre conclusion porte sur la consistance d'une telle démarche, l'utilité d'un tel objet et les perspectives de développement futures.

1. INTRODUCTION

Ce texte relate un développement logiciel au service d'une démarche artistique en interaction avec des exigences scientifiques et technologiques.

Cet objet Max/MSP a été développé en fonction de deux problématiques. En premier lieu proposer une solution répondant à l'absence d'un objet multi-courbes performant dans Max/MSP. En second lieu offrir une interface de contrôle très fine et supportant une fréquence d'échantillonnage proche de la milliseconde. Ce développement fait partie d'une démarche d'intégration des outils et des technologies scientifiques sur des plateformes logicielles utilisables sur le terrain des Arts interactifs.

J'exposerai tout d'abord le contexte général au sein duquel ce projet a été réalisé. Je présenterai premièrement l'approche théorique originale, celle-ci prend place dans un cadre de recherche en sciences humaines, sciences de l'art, spécialité musique, alliant la création artistique, les sciences et les technologies¹. On s'intéressera également au cheminement expérimental, on insistera sur la collaboration pluridisciplinaire et sur les précédents projets à l'origine du développement de

cet objet. Je ferai aussi un bref état des lieux des objets Max/MSP destinés à la représentation graphique de courbes de valeurs, j'insisterai en particulier sur les limites de ces objets et l'intérêt de développer un objet plus performant.

J'introduirai ensuite l'objet Multicurve. J'exposerai en premier lieu son architecture générale, ses fonctions graphiques et ses fonctions relatives à la gestion de données. Je m'attacherai ensuite à expliquer le mécanisme de certains algorithmes lui permettant de réaliser des performances intéressantes au niveau de la consommation des ressources du processeur.

J'évoquerai finalement les résultats des tests effectués avec Multicurve afin de quantifier ses performances techniques.

L'article ci-dessous présente le développement d'un objet Max/MSP à travers le contexte d'expérimentation particulier dans lequel il trouve son origine puis à travers un angle plus technique et descriptif. Il conclut sur des questions d'interaction entre les outils technologiques scientifiques et artistiques, et la place d'un tel développement dans cette démarche.

2. LE CONTEXTE À L'ORIGINE DE CE DÉVELOPPEMENT LOGICIEL

2.1. Le contexte théorique

Dès 2008 je me suis intéressé au développement d'interfaces logicielles et physiques permettant la captation de signaux corporels et leur utilisation comme contrôleurs sonores² [1]. C'est dans le cadre de ce sujet que je me suis orienté vers les sciences cognitives. Cela m'a permis d'étudier de plus près le fonctionnement "sensori-moteur" de l'être humain qui est, à mes yeux,

¹ Licence, Master et Doctorat au département de musique de l'université de Paris 8, Ecole Doctorale "esthétique, sciences et technologies des arts." <http://artweb.univ-paris8.fr> et www.edesta.fr,

² Jean-François Baud, Anne Sèdes, « *Vers une approche physiologique de l'interface corporelle pour les arts interactifs* », prix du jeune chercheur de l'Association Française d'Informatique Musicale, JIM 2009.

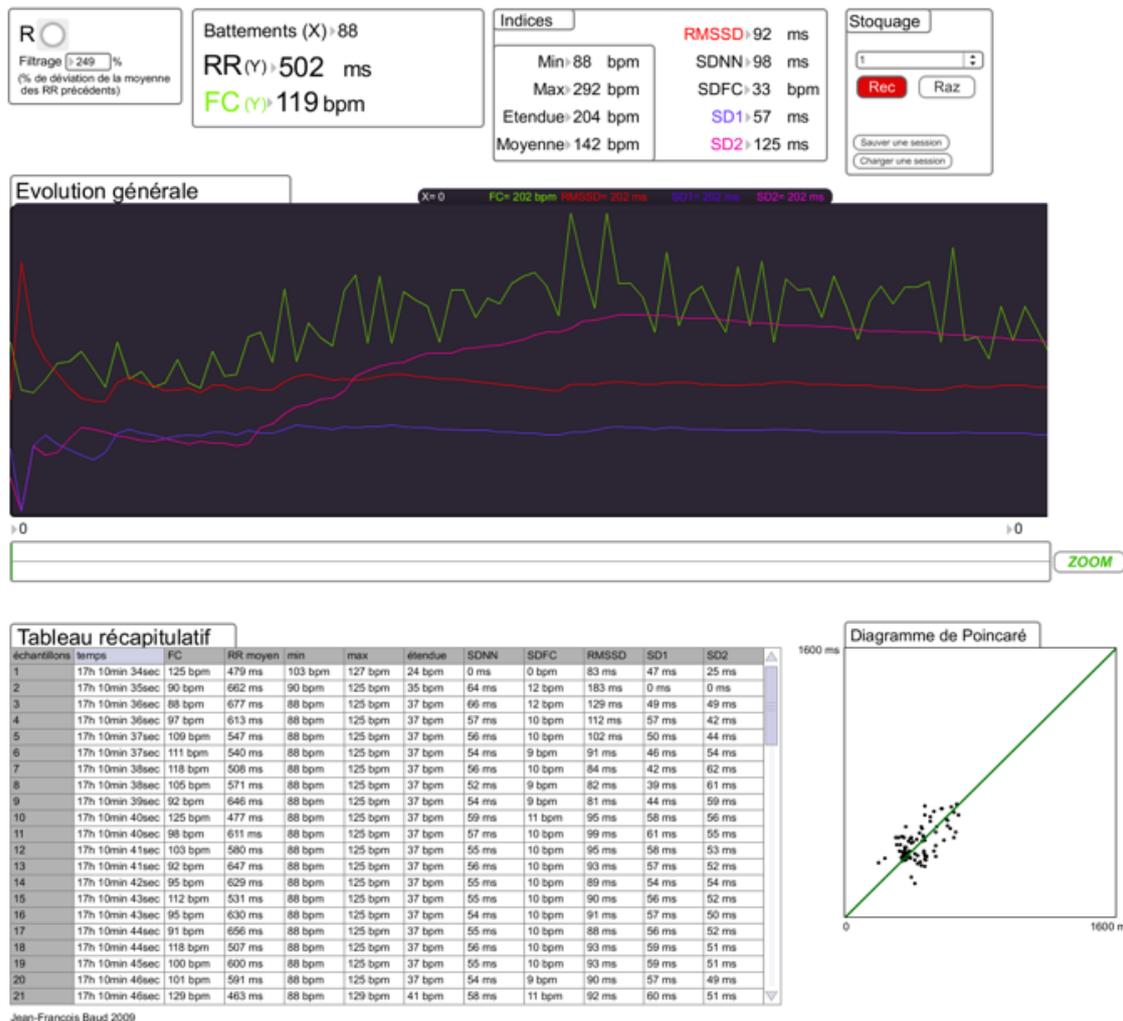


Figure 1: fenêtre principale du logiciel Varialog

l'élément principal au sein de tels dispositifs interactifs. Je me suis alors plus particulièrement tourné dans la direction des théories de Francisco Varela [2] concernant l'enaction et la corporéité (l'embodiment). Ces théories exposent le lien d'interaction particulier entre les processus cognitifs et leurs conséquences physiques ou physiologiques. Cette thématique est également reprise par Alain Berthoz qui y ajoute la problématique de la place des émotions au sein de ce processus. Dans son ouvrage *La décision* [3] il aborde la question du lien entre les émotions et la décision et plus précisément l'ancrage somatique des émotions. On y apprend que les premières conséquences corporelles de la perception seraient détectables dans la régulation des fonctions automatiques de l'organisme et seraient donc significatives de l'implication émotionnelle du sujet de la perception.

M'appuyant sur ce postulat, j'ai donc adopté comme idée directrice qu'un dispositif d'interfaces physiques et logicielles captant et analysant la régulation de certaines fonctions automatiques de l'organisme nous permettrait d'obtenir des informations sur l'état psychologique, émotionnel du sujet de la captation. Ces informations intéressantes seraient ensuite exploitées comme contrôleurs d'un dispositif multimédia.

2.2. Une collaboration pluridisciplinaire

L'analyse de la régulation des fonctions automatiques de l'organisme étant un objet d'étude important de la physiologie, j'ai alors entrepris une collaboration avec l'équipe de chercheurs de la plate-forme de recherche en Sciences et Techniques des Activités Physiques et Sportives (Staps) de l'UFR de Santé, Médecine, Biologie Humaine de l'Université de Paris 13, laboratoire "Réponses cellulaires et fonctionnelles à l'hypoxie". Cette équipe, animée par Aurélien Pichon, étudie plus particulièrement la variabilité cardiaque et son interprétation à travers le prisme de la physiologie et de la psychologie [4] [5].

C'est au sein de cette collaboration que j'ai réalisé le logiciel Varialog me permettant de m'approprier les outils d'analyse de la variabilité cardiaque des physiologistes et de les intégrer sur la plate-forme logicielle Max/MSP utilisable sur le terrain des Arts interactifs. Le logiciel fut finalisé en mars 2009.

2.3. Un premier prototype d'objet de visualisation de plusieurs courbes

Ce logiciel d'analyse temps réel de la variabilité cardiaque permet de calculer et d'enregistrer les valeurs des intervalles temporels entre chaque battement cardiaque et d'opérer une série d'analyses à l'aide de différents indices statistiques, l'évolution des quatre indices principaux est représentée en temps réel sous forme de courbes, ces valeurs sont ensuite stockées dans un tableau.

C'est lors de la programmation de la fenêtre de visualisation des quatre courbes que je me suis heurté à une difficulté technique, en effet Max/MSP ne dispose pas d'un objet efficace permettant la visualisation de plusieurs séries de valeurs qui s'implémentent en temps réel. Pourtant c'est une fonction principale présente dans la plupart des logiciels destinés à l'usage scientifique, tel que MatLab. Techniquement le logiciel réalise cela en augmentant automatiquement la taille de l'échantillon pour chaque nouvelle valeur reçue, la ou les courbes sont alors redessinées en ayant donc été incrémentées d'un nouveau point. Ce procédé permet visuellement d'avoir en permanence l'ensemble de l'échantillon représenté et donc de percevoir l'évolution globale des valeurs dans le temps.

Je me suis alors tourné vers les objets existants pour trouver une solution alternative.

2.4. Les limites des objets Max destinés à la représentation graphique de courbes de valeurs

Les objets principaux destinés à cet usage sont Multislidier, Itable et Function. Ils sont cependant limités à une courbe³⁴ et ne peuvent être superposés du fait de l'absence du mode "transparent". L'objet Lcd ne peut également pas répondre à cette problématique car il ne garde aucune mémoire de quelconques valeurs, il est seulement destiné à la représentation graphique.

Pour résoudre ce problème technique j'ai donc couplé plusieurs Multislidiers avec un Lcd, ce qui m'a permis d'utiliser la mémoire des Multislidiers conjointement à la capacité graphique du Lcd. Cette solution alternative, difficile à programmer mais opérationnelle, m'a permis de finaliser mon programme. Elle présente cependant des inconvénients majeurs :

Une sollicitation du processeur trop importante et une limitation du nombre de valeurs maximales utilisables :

³ L'objet Multislidier en mode "scroll" peut afficher plusieurs courbes simultanément cependant elles n'apparaissent pas dans le même repère spatial, il s'agit en fait d'une superposition de repères spatiaux ce qui ne répond pas à notre problématique technique. De plus le mode "scroll" n'augmente pas la taille de l'échantillon pour chaque nouvelle valeur reçue, mais efface les valeurs les plus anciennes.

⁴ Dans la bibliothèque FTM développée par l'IRCAM il existe l'objet Vectorsdisplay qui permet la visualisation de 5 courbes simultanément. Il fonctionne soit avec des *list* de *floats* soit avec les autres objets de FTM. Je n'ai pas retenu cet objet car ce système de *list* m'a paru inadapté à mon utilisation (une *list* est limitée à quelques centaines de valeurs), de plus la prise en main de cette bibliothèque est relativement complexe.

De manière générale, tous ces objets Max destinés à la représentation graphique de valeurs "consomment" un pourcentage élevé des ressources du processeur lors de leur fonctionnement. Cette consommation est évidemment proportionnelle au nombre de valeurs stockées dans la mémoire des objets.

En effectuant des tests sur ces objets, je me suis rendu compte des limites qu'ils présentaient réellement : éditer la courbe d'un Multislidier au maximum du nombre de ses sliders (32767) ou éditer une Itable de 10 000 valeurs ou bien encore écrire rapidement (toute les 10 ms) de nouveaux points dans un objet Function représente une consommation de 60% à 80% du processeur⁵. Cette consommation est trop importante pour de simples opérations de gestion de données de contrôle, de plus elle présente de réels risques de *crash* de Max/MSP.

On peut donc considérer que ces objets Max permettent une gestion et une visualisation efficace de données dans la limite d'une courbe de quelques milliers de valeurs au maximum. Ils ne sont donc pas compatibles avec un échantillonnage trop rapide ou avec une durée d'enregistrement trop longue⁶.

J'ai alors entrepris le développement d'un objet dépassant ces limites de fonctionnement et répondant aux exigences du contexte scientifique et artistique dans lequel se trouvent mes travaux.

3. PRÉSENTATION ET FONCTIONNEMENT DE MULTICURVE

3.1. Présentation générale et construction des courbes

Dans un patch Max/MSP, on peut lancer Multicurve en inscrivant son nom dans une boîte objet ou en le sélectionnant directement dans la palette des objets graphiques. Il se présente sous forme d'une fenêtre noire comprenant 5 entrées et 5 sorties. Les 4 premières entrées reçoivent 4 flux de données de contrôle qui correspondent aux 4 courbes représentées dans la fenêtre de l'objet (par défaut l'entrée 1 correspond à la courbe verte, le second à la courbe rouge, le troisième à la courbe bleue puis le dernier à la courbe violette⁷). Ces flux de valeurs sont échantillonnés en fonction de l'arrivée de nouvelles valeurs dans l'entrée 1 à la façon d'un objet Pack, autrement dit dès que l'entrée 1 reçoit une nouvelle valeur il "autorise" les 3 autres entrées à intégrer de nouvelles valeurs. Ce procédé permet de donner à ces 4 flux de données la même temporalité. Il convient donc de lier à l'entrée 1 le flux de données le plus rapide afin d'avoir la meilleure résolution temporelle.

⁵ Ces tests ont été réalisés avec un MacBook doté d'un processeur de 2.16 GHz, 1 Go de RAM, Mac 10.6.2 et MaxMSP 5.1

⁶ Les limites pour une consommation de CPU acceptable (aux alentours de 30%) se situent en réalité à 5000 valeurs maximales par objet et une écriture de nouveaux points tout les 200ms au minimum.

⁷ Les couleurs des courbes ainsi que celles de tous les repères visuels sont paramétrables dans l'inspecteur de l'objet.

Après l'échantillonnage, les quatre courbes sont donc incrémentées d'un nouveau point correspondant au "paquet" des 4 valeurs échantillonnées. Graphiquement, pour chaque nouveau "paquet" toutes les courbes sont redessinées et auto-ajustées à la taille de la fenêtre, ce système permet visuellement d'avoir en permanence l'ensemble de l'échantillon. L'axe des abscisses est donc ajusté dynamiquement, les ordonnées par contre correspondent à une échelle, un *range* paramétrable dans l'inspecteur.

La sortie 5 exprime le nombre total de points par courbe (soit le nombre total de "paquets" ou encore la taille de l'axe des abscisses).

L'entrée 5 quant à elle est un index de lecture des courbes. Elle considère chaque valeur reçue comme une abscisse et renvoie respectivement dans les sorties 1, 2, 3, 4 les ordonnées des quatre courbes, correspondant à cette abscisse.

3.2. Repères visuels

Lors du développement de l'objet, un critère m'a paru très important : le caractère intuitif de l'interface graphique. C'est dans cet esprit que j'ai implémenté l'objet de nombreux repères visuels dynamiques facilitant la lecture des courbes. Le principe est d'amener certaines informations de lecture des courbes à l'endroit où l'utilisateur focalise son attention.

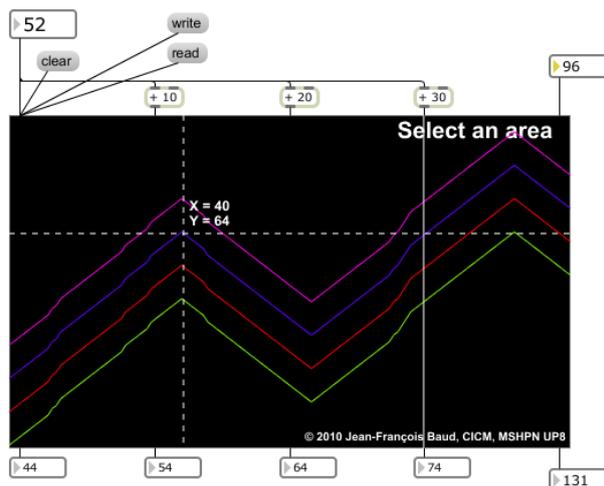


Figure 2: fenêtre de visualisation de Multicurve et repères visuels.

Par exemple lorsque la souris est à l'intérieur de la fenêtre, ses coordonnées sont exprimées dans l'échelle des courbes et suivent visuellement la trajectoire du curseur. Ce procédé permet de prendre connaissance des coordonnées d'un point simplement en plaçant le curseur sur celui-ci et donc sans se référer à un repère fixe tel un axe des abscisses. Notre repère de coordonnées est donc dynamique et plus intuitif pour l'utilisateur. Deux axes pointillés (un vertical et un horizontal) sont également associés à la trajectoire du curseur dans la fenêtre, ils facilitent la lecture des courbes et en particulier la

comparaison entre les courbes ou entre les différents "pics" des courbes.

Lors de la lecture des courbes, l'abscisse envoyée dans l'entrée 5 est matérialisée dans l'objet par une barre verticale ce qui permet de suivre visuellement cette lecture. Ceci est d'autant plus important lors d'une lecture rapide⁸.

Dans le cas d'une échelle des ordonnées comprenant des valeurs négatives et positives, un axe correspondant à Y=0 est matérialisé par une barre horizontale.

3.3. Modes de fonctionnement

Comme nous l'avons vu, Multicurve interagit avec la souris simplement lorsque celle-ci est à l'intérieur de la fenêtre de visualisation. Les "clicks" de la souris sont également pris en compte puisqu'ils donnent accès à deux modes de fonctionnement. Ces modes correspondent à deux fonctions principales de l'objet : la visualisation et l'édition précise de plusieurs courbes de données.

Lorsque le curseur entre dans la fenêtre le message "Select an area" apparaît en haut à droite de l'écran, il suffit alors de sélectionner une zone en cliquant pour opérer un zoom sur celle-ci. Sur cette fenêtre sélectionnée, il est alors possible d'effectuer deux opérations : zoomer ou éditer. Pour zoomer il faut procéder comme précédemment, en sélectionnant une zone. Il est possible de zoomer jusqu'à sélectionner une zone composée seulement de quelques points. Pour revenir ensuite à la fenêtre de visualisation de l'ensemble de l'échantillon, il faut simplement faire sortir le curseur de la fenêtre de l'objet.

Pour éditer une courbe il faut placer le curseur sur un de ces points, la courbe apparaît alors en surbrillance, il suffit alors de maintenir enfoncée la souris et de rectifier la courbe.

Le mode de fonctionnement dans lequel on se trouve est indiqué en haut de la fenêtre à droite : "ZOOM" ou "EDIT".

Il est possible de réaliser ces opérations à tout moment, même durant l'écriture ou/et la lecture des courbes.

⁸ Un simple objet Metro associé à un Counter lié à l'entrée 5 permet de réaliser un module de lecture à vitesse variable, contrôlable à la milliseconde près.

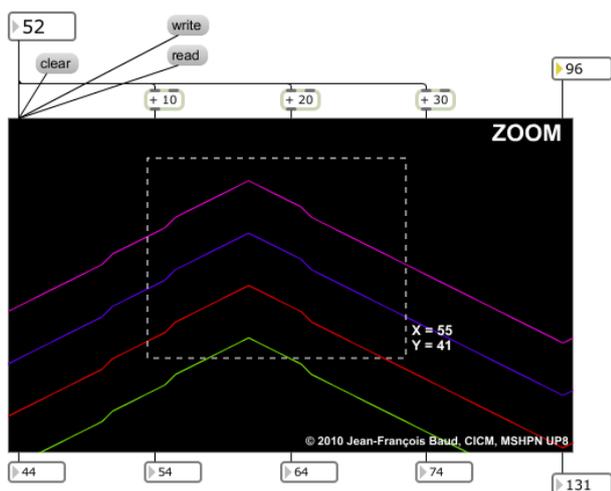


Figure 3: Multicurve en mode zoom.

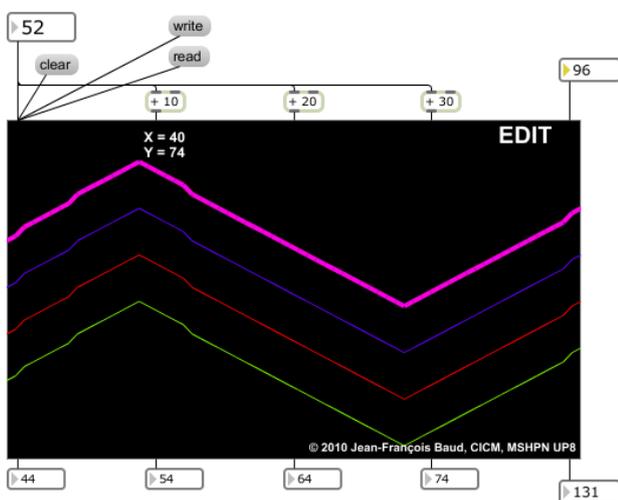


Figure 4: Multicurve en mode édition.

3.4. Messages et fonction de stockage

Multicurve est sensible à trois messages venant de Max/MSP. Le message "Clear" efface toutes les données implémentées dans l'objet et par conséquent les courbes. Le message "Write" active une fonction de stockage des courbes sous forme de fichier. Plusieurs fenêtres correspondant au nombre de courbes s'affichent successivement et demandent à l'utilisateur de nommer le fichier contenant les valeurs de la courbe puis de le placer à un endroit du disque dur. Le message "Read" effectue l'opération inverse en chargeant dans l'objet des courbes sauvegardées sous forme de fichier. L'utilisateur peut ainsi sauvegarder ses données facilement mais également charger dans l'objet des courbes appartenant à différentes "sessions d'enregistrement", si les courbes sont de longueurs différentes, un ajustement sera fait en fonction de la plus grande.

Bien que le nombre d'entrées et sorties soit constant, il est possible de paramétrer le nombre de courbes dans l'inspecteur, dans la limite de 4 maximum. Ce paramètre est intéressant car il permet de réaliser un gain de CPU suivant le nombre de courbes.

3.5. Caractéristiques techniques, tests et performances de l'objet

Comme je l'ai précisé, l'intérêt d'un tel développement est d'obtenir un objet plus performant sur les critères d'économie de ressources processeurs, de fonctionnement avec un grand nombre de valeurs, un échantillonnage très rapide et la possibilité d'écrire plusieurs courbes.

L'optimisation de la consommation des ressources CPU est donc le critère central de ce développement. J'ai résolu ce problème technique grâce à deux implémentations. La première est un mécanisme de limitation de la fréquence de rafraîchissement de la fenêtre graphique de l'objet, cette fréquence est calculée dynamiquement et assure une faible consommation de CPU sans altérer la "réactivité graphique" de l'objet. La seconde implémentation est un algorithme qui s'active lorsque le nombre de points des courbes est plus important que le nombre de pixels de la fenêtre graphique, pour réduire alors la taille des courbes il effectue alors une "mise en paquets" des valeurs de la courbe puis effectue des moyennes de ces paquets qu'il propose alors comme les nouvelles courbes. La taille des paquets est définie dynamiquement dans un rapport de proportionnalité au nombre total des valeurs brutes. Cet algorithme permet un rendu visuel des courbes le plus fidèle possible à leurs valeurs brutes.

Grâce à ces implémentations Multicurve consomme peu de ressources processeur et répond aux critères de développement évoqués plus haut. J'ai effectué de nombreux tests tout au long du développement en le poussant dans ses limites de fonctionnement, d'une manière générale il ne dépasse quasiment jamais 30% de CPU. Plus précisément, pour un objet de quatre courbes et une fréquence d'échantillonnage à 1 ms, sa consommation se situe tout d'abord aux alentours de 20% de CPU puis elle dépasse les 30% lorsque l'objet contient un million de valeurs par courbe. J'ai donc borné l'objet à 1 500 000 valeurs par courbes pour rester sous les 40% de CPU.

Lors des opérations d'édition et de zoom, même avec un million de valeurs, la consommation ne dépasse jamais les 20%, et toujours dans ces mêmes conditions si on lit les courbes à une vitesse d'un X par ms le processeur consomme aux alentours de 25% de ses ressources.

4. CONCLUSION

Dans la suite de ce développement, je pense étudier certaines possibilités d'optimisation intéressantes afin de les implémenter à Multicurve. Une première idée d'optimisation serait de donner à chaque courbe sa

propre échelle, son propre axe des ordonnées. Cela permettrait visuellement de faire cohabiter des courbes ayant une échelle de valeurs très différente.

Il serait ensuite intéressant d'implémenter un mode "oscilloscope" permettant de visualiser les flux de données de manière dynamique.

Il conviendrait également de pouvoir "faire glisser" une fenêtre de zoom, afin de naviguer dans tout l'échantillon seulement avec cette fenêtre choisie.

La version actuelle de Multicurve est téléchargeable sur le site du CICM⁹.

Dans cet article, le développement de cet objet a été présenté au sein d'une démarche de récupération et d'adaptation des outils scientifiques à un contexte artistique¹⁰. Cette démarche se veut au service des Arts Interactifs, considérant que ces contributions scientifiques et technologiques nouvelles sont porteuses de consistances artistiques.

Dans cette approche, la captation n'est pas simplement considérée comme génératrice d'un flux de données de contrôle dynamiquement intéressant à exploiter. Elle est avant tout considérée comme un moyen d'interfacer un aspect de la réalité et d'en restituer, après une phase d'analyse, un flux de données de contrôle soulignant cette réalité. Par exemple une pulsation cardiaque exploitée telle quelle a un potentiel expressif limité car ses variations sont assez régulières et ont peu d'amplitude, cependant si on l'exploite avec des outils statistiques analysant la variabilité, on obtient alors un flux de données beaucoup plus expressif parce que soulignant, exprimant les moindres variations subtiles du rythme cardiaque brut.

Cette approche d'appropriation des outils scientifiques est donc au service de l'expressivité. Multicurve est un objet se situant dans cette démarche car il répond aux exigences de ces outils d'analyses scientifiques, en particulier sur le terrain de la précision et de la résolution temporelle.

Le développement de cet objet nous permet ainsi d'exploiter les relations fructueuses entre arts, sciences et technologies, dans un contexte profondément interdisciplinaire.

5. RÉFÉRENCES

- [1] Jean-François Baud, Anne Sèdes, "Vers une approche physiologique de l'interface corporelle pour les arts interactifs", JIM 2009.
- [2] Francisco Varela, Evan Thompson, Eleanor Rosch. *L'inscription corporelle de l'esprit*, sciences cognitives et expérience humaine, p. 235 Seuil, Paris, 1993.
- [3] Alain Berthoz : *La décision*, Odile Jacob, 2003, Paris.
- [4] Aurélien Pichon, Claire de Bisschop, Véronique Diaz, André Denjean : "*Parasympathetic airway response and heart rate variability before and at the end of methacholine challenge*", Chest, 2005.
- [5] Aurélien Pichon, Chapelot Didier, Nuissier Frédéric : "*Heart rate variability is an indicator of changes in psychological states over a year in university students*", psychophysiologie, 2008.

⁹ <http://cicm.mshparisnord.org/>

¹⁰ Cet objet sera d'ailleurs utilisé pour effectuer des analyses de variabilité cardiaque sur des instrumentistes en diverses situations d'improvisations, dans le cadre d'un projet dirigé par Cécile Vallet de la MSH Paris Nord.