

# **Preset de synthétiseur logiciel : mapping des macros-paramètres**

**Thomas GUILLORY**

**ENS Louis Lumière**

Mémoire de Master 2 - Spécialité son

Directeur de mémoire interne : **Nicolas MONTGERMONT**

Directeur de mémoire externe : **Antoine MARTIN**

Responsable universitaire : **Corsin VOGEL**

Examinateur : ?

Référent académique : **Laurent MILLOT**

Juin 2024

# Abstract

This dissertation reflects on the limitations of macro parameters in software synthesizer presets: little interaction and little invitation to explore.

We wish to consider presets as “micro-instruments” that require the same care and attention as any other instrument.

Drawing on the concepts of agentivity and affordance, we will consider how the evolution of the three fundamental blocks of plug-ins (signal processing, graphical interface and mapping) could enhance the use of presets.

In particular, we'll look at macro-parameter mapping, so that luthiers can offer users a more engaging interaction with their software instruments.

**Keyword** : preset, synthesizer, agency, affordance, mapping, DSP, GUI, macro-knob

# Résumé

Ce mémoire propose une réflexion sur les limites des macro-paramètres des pré-réglages intégrés aux synthétiseurs logiciels : peu d'interactions et d'invitation à l'exploration.

Nous souhaitons considérer les *presets* comme des « micro-instruments » qui nécessitent le même soin et la même attention que n'importe quel instrument.

En s'appuyant sur les concepts d'agentivité et d'affordance, nous envisagerons comment l'évolution des trois blocs fondamentaux des *plug-ins* (traitement du signal, interface graphique et *mapping*) pourrait améliorer l'utilisation des *presets*.

Nous étudierons en particulier le *mapping* des macro-paramètres, afin que les luthier·ère·s puissent proposer aux utilisateur·rice·s une interaction plus engageante avec leurs instruments logiciel.

**Mot clés** : preset, synthétiseur, agentivité, affordance, mapping, DSP, GUI, macro-paramètre

**Citation (APA)** : Guillory, T. (2024) *Preset de synthétiseur logiciel : mapping des macros-paramètres*. École Nationale Supérieure Louis-Lumière

## Remerciements

Merci à mes deux directeurs, Nicolas Montgermont et Antoine Martin pour avoir partagé leurs idées et participer à organiser les miennes.

À Corsin Vogel pour son efficacité et son suivi.

À Remy Müller, Rodrigo Constanzo et Julien Bloit pour avoir pris le temps d'échanger.

Au forum llllll.co qui a nourrit ma réflexion

À toute la promotion son 2024 et en particulier à mes compagnons de travail à la bibliothèque.

À ma mère pour avoir pris le temps relire mon mémoire.

À mes colocataires qui m'ont aimablement écouté parler de mapping pendant de longs mois.

<b>Abstract</b>	<b>2</b>
<b>Résumé</b>	<b>2</b>
<b>Remerciements</b>	<b>3</b>
<b>Introduction</b>	<b>7</b>
<b>Musique Assistée par Ordinateur : outils et processus</b>	<b>9</b>
<b>1.1 Station de travail audio-numérique</b>	<b>10</b>
<b>1.2 Évènement sonore</b>	<b>11</b>
1.2.1 Audio	11
1.2.2 Synthétiseurs	12
1.2.3 Effets	12
1.2.4 Modulations	13
<b>1.3 Plug-ins</b>	<b>14</b>
1.3.1 Contexte technique	15
1.3.2 Conception	16
1.3.3 DSP	18
1.3.4 Interface	18
<i>Affordance</i>	19
<i>Skeuomorphisme</i>	20
<i>Retour visuel et interface expérimentale</i>	23
1.3.5 Mapping	25
<b>1.4 Presets</b>	<b>28</b>
1.4.1 Histoire des presets	29
<i>Synthétiseur acoustique</i>	29
<i>Synthétiseurs analogiques</i>	29
<i>Synthétiseurs numériques</i>	30
<i>Synthétiseurs logiciels</i>	31
1.4.2 Plug-ins hôtes et macro-contrôles	32
<b>Limites et réflexions</b>	<b>35</b>
<b>2.1 Les presets et leurs limites</b>	<b>36</b>
<b>2.2 Agentivité : contextes et contraintes</b>	<b>38</b>
<b>2.3 Outil ou instrument ?</b>	<b>40</b>

<b>2.4 Sources d'amélioration potentielles</b>	<b>42</b>
2.4.1 DSP	42
2.4.2 Interface	43
2.4.3 Mapping	45
<b>Mapping, revue de littérature</b>	<b>46</b>
<b>3.1 Shaping</b>	<b>47</b>
<b>3.2 Topologies de Mapping</b>	<b>49</b>
<b>3.3 Exemple de mapping créatif</b>	<b>50</b>
<b>3.4 Couche de paramètres intermédiaires</b>	<b>52</b>
<b>3.5 Interpolation de preset</b>	<b>53</b>
<b>3.6 Modèle physique</b>	<b>55</b>
<b>3.7 Non-linéarité</b>	<b>56</b>
<b>Comparaison expérimentale des mappings</b>	<b>58</b>
<b>4.1 Contexte d'expérimentation</b>	<b>59</b>
<b>4.2 Conception des presets</b>	<b>61</b>
<b>4.3 Prototype de mapping</b>	<b>62</b>
4.3.1 Mapping one-to-one et one-to-many	62
4.3.2 Interpolation	63
4.3.3 Mapping non-linéaire	63
<b>4.4 Protocole de test</b>	<b>65</b>
4.4.1 Questionnaire	65
4.4.2 Déroulé de l'expérience	66
<b>4.5 Analyse des résultats</b>	<b>67</b>
4.5.1 Profil des participant·e·s	67
4.5.2 Calcul de l'écart type entre les sessions	69
4.5.3 Usage personnel	70
4.5.4 Mappings inhabituels	72
4.5.5 Espace timbral complexe	73
4.5.6 Automation	74

4.5.7 Contexte live ou studio	75
4.5.8 Amusement	75
4.5.9 Questions ouvertes	76
4.5.10 Note sur l'utilisation du pad	77
4.5.11 Influence des mappings sur la conception des presets	78
<b>4.6 D'autres espaces à explorer</b>	<b>79</b>
<b>Conclusion</b>	<b>82</b>
<b>Bibliographie</b>	<b>84</b>
Recommendation de lecture	88
<b>Table des Figures</b>	<b>90</b>
<b>Annexes</b>	<b>92</b>
Synthétiseurs des années 90	92
Questionnaire	93

# Introduction

*"An instrument is a way to pass knowledge"*

-Petter Blasser (Tagi, E. 2023, 11 juillet)

Les outils et instruments utilisés dans la composition musicale assistée par ordinateur sont souvent complexes. Les paramètres portent en général des noms techniques décrivant des phénomènes physiques. Les synthétiseurs sont un bon exemple de ces outils nécessitant un temps d'apprentissage non-négligeable. Pour faciliter leur utilisation, ces outils sont fournis avec des pré-configurations (*presets*). Il est parfois difficile de trouver des outils proposant un équilibre entre l'exhaustivité, la complexité des synthétiseurs, et la simplicité parfois limitante de leurs *presets*, c'est-à-dire des instruments qui font une utilisation raisonnée des contraintes, qui construisent sciemment leurs affordances, et qui accordent autant d'importance à l'interaction qu'au résultat produit. Ces outils bénéficieraient d'une meilleure perception de la place des instruments dans les processus créatif, ainsi que d'une réflexion approfondie sur les comportements qu'induisent les choix de conception chez l'utilisateur·rice·s et les contraintes musicales qu'ils imposent, qu'elles soient bienvenues ou non.

Afin de déterminer où et comment nous pourrions amorcer une évolution vers ces instruments, je détaillerai dans une première partie le fonctionnement des stations de travail audionumériques. J'étudierai les trois composantes d'un synthétiseur logiciel : interface, *mapping* et traitement du signal, puis l'historique et l'utilisation actuelle des *presets*.

Dans une deuxième partie, j'aborderai les limites de ce système en prenant en compte l'ensemble des acteurs impliqués dans la conception et l'utilisation des instruments numériques. Afin que les concepteur·rice·s et des artistes s'approprient plus facilement de nouveaux types d'interaction, le plus simple est de construire à partir de ce qui existe déjà. Je proposerai alors de complexifier le *mapping* des macro-paramètres des synthétiseurs logiciel. Cela pourrait permettre aux sound-designeur·e·s de concevoir des instruments plus complets, et de proposer aux

artistes une interaction plus engageante - voir de les inviter, s'ils le souhaitent, à concevoir eux même leurs instruments. Les pré-configurations étant présentes dans la quasi-totalité des outils logiciels utilisés lors de la réalisation d'une création musicale, leur amélioration pourrait s'étendre au-delà des synthétiseurs ; aux effets, notamment.

Dans une troisième partie, je ferai une revue de littérature du *mapping* pour explorer les différentes options envisageables.

Enfin, dans une dernière partie, je sélectionnerai quelques *mappings* identifiés comme pertinents lors de la partie précédente et proposerai une expérience afin de les comparer.

# **Musique Assistée par Ordinateur : outils et processus**

Je définirais ici la musique comme un “son organisé” (*organized sound*) (Goldman, 1961), et la MAO comme un son organisé par le biais de l’utilisation partielle ou exclusive de l’ordinateur.

L’ordinateur a aujourd’hui une place centrale dans la composition musicale de nombreux artistes. La multiplicité des envies et des processus est reflétée dans la variété d’outils à disposition, plus ou moins accessibles, amenant chacun·e à des logiques de travail différentes.

Bien que conçus pour un contexte donné, les outils évoluent au fil du temps en fonction des désirs des utilisateurs. Par exemple, certains logiciels à l’origine conçus pour la performance en direct se sont vus de plus en plus utilisés en studio. Les outils influencent la réflexion des artistes et la réflexion des artistes influence les outils.

De tous les types de logiciels disponibles, les plus utilisés restent largement les STANs (ou DAWs<sup>1</sup>), qui cherchaient à l’origine à émuler les méthodes de travail des studios de musique et de cinéma. Les DAWs sont les outils fondamentaux de la musique électronique moderne, qu’ils soient au centre de la démarche créative ou qu’ils servent simplement d’enregistreurs pour du matériel physique.

Je détaillerai d’abord les composantes du DAW avant d’apporter plus de précisions sur le cas particulier des *plug-ins* et de leurs *presets*.

## 1.1 Station de travail audio-numérique

Dans un DAW, on retrouve en général :

- des sources sonores, synthétiseurs et banques de sons
- un ensemble d’effets
- une section mixeur permettant de gérer le volume et la position spatiale des événements sonores

---

<sup>1</sup> Station de Travail Audio Numérique ou Digital Audio Workstation (plus répandu), par exemple : Ableton Live, Protools, Reaper, Logic, etc...

- une *time-line* : représentation horizontale du défilement du temps sur laquelle on vient déposer et éditer des évènements sonores

L'utilisation d'un DAW ne suppose pas nécessairement l'utilisation de toutes ses sous-parties. Pour une performance en *live*, certaines personnes peuvent par exemple se passer de *time-line*.

## 1.2 Évènement sonore

Une musique est composée d'un ensemble d'évènements sonores organisés dans le temps, ces évènements peuvent prendre n'importe quelle forme (le silence peut être considéré comme un évènement s'il est précédé par du son). La distinction entre deux évènements étant déterminée par notre perception, elle n'est pas toujours évidente. Elle se fera fréquemment, spatialement ou temporellement. Deux évènements distincts peuvent donc avoir deux sources distinctes, deux sources peuvent fusionner en un seul évènement, et deux évènements peuvent provenir de la même source.

Les évènements sonores peuvent avoir 4 sources :

- les fichiers audios
- les synthétiseurs
- les effets
- la modulation

### 1.2.1 Audio

Les fichiers audio, ou *samples* (échantillons) peuvent être de tout type (percussif, voix, morceau entier, phrase au piano) et de toute origine (acoustique, électronique, numérique). Certaines entreprises se sont spécialisées dans la conception et la vente de ces *samples*, souvent sous forme de boucles à un tempo donné afin faciliter leur utilisation. Les utilisations du *sample* sont très variées : certaines personnes les laissent bruts, d'autres les modifient à un point où la source n'est plus reconnaissable. Le *sample* devient un matériau permettant de faire vivre les outils

de modifications. Ces outils sont sur le spectre des synthétiseurs, parfois plus proches de l'audio, parfois plus proches de la synthèse.

## 1.2.2 Synthétiseurs

Un synthétiseur est un instrument générant des signaux audio : il en existe de nombreux types, utilisant des formes variées d'algorithme pour modéliser leur timbre. Les synthétiseurs physiques embarquent communément des claviers de piano en gamme tempérée (une interface familière pour les compositeur·rice·s). Une spécification s'est développée autour de ce cadre afin de faciliter le contrôle des synthétiseurs. Elle comprend un ensemble de normes précisant la forme des messages à envoyer au moteur de synthèse pour déclencher une note et modifier ses variables. La norme MIDI (apparue en 1983) est devenue omniprésente dans les logiciels de composition, les synthétiseurs logiciel utilisent toujours cette même spécification.

Un message MIDI a trois fonctions principales :

- jouer une note, en envoyant son index<sup>2</sup> et la manière dont elle est jouée (le paramètre de vitesse, par exemple, nous informe sur la vitesse à laquelle la touche est appuyée)
- rappeler des pré-configurations du synthétiseur, à l'aide de messages “*Program Change*”
- contrôler un paramètre de synthèse en utilisant des messages “*Continuous Controller*” (CC). Ces contrôles ont une valeur allant de 0 à 127 (en MIDI 1.0, qui utilise 8 bits pour les CC). Si le synthétiseur respecte l'ensemble des recommandations de la spécification, certains CC seront attribués à certains paramètres communs, comme le CC 7, destiné au volume (The MIDI Manufacturers Association. 1996)

## 1.2.3 Effets

---

<sup>2</sup> À titre d'exemple, le La de la quatrième octave MIDI (à 440Hz) porte la valeur 69, le Si 4 la valeur 70, etc.

Lorsque le fichier audio est lu, ou le message MIDI envoyé au synthétiseur, on peut modeler le son à l'aide d'une multitude d'effets. Ils ont une place très importante dans la musique électronique et ont une grande influence sur l'identité sonore d'un artiste (Sramek *et al.* 2023).

À l'origine, lorsque les machines étaient physiques, les ingénieur.e.s du son utilisaient un *patch* permettant de construire le chemin du signal à travers les unités de traitement désirées. Les outils de traitements analogiques fondamentaux sont la correction fréquentielle (égaliseur), la correction dynamique (compresseur / expandeur) et les simulation d'acoustique (réverbération acoustique, écho à bande). Les débuts du numérique ont ouvert la voie à de plus amples possibilités de génération du son et à la multiplication des effets sur une même voie. La numérisation du son permet de rendre bien plus simples des procédés auparavant laborieux.

À titre d'exemple les *buffers*<sup>3</sup> simplifient largement les procédés granulaires<sup>4</sup> ; ils remplacent un fastidieux travail de découpe de bande analogique par un simple geste sur un potentiomètre.

Ce mémoire traitera essentiellement des synthétiseurs pour des raisons de concision, mais la majorité des points abordés sont transposables aux effets.

## 1.2.4 Modulations

La modulation des paramètres, c'est-à-dire leur modification à travers le temps, est un des procédés fondamentaux de la musique électronique moderne (Smith, 2021). Ces mouvements permettent de rendre plus vivant les algorithmes de synthèse en faisant évoluer leur timbre. Les synthétiseurs ont toujours embarqué des sources de modulation. Ce sont au plus simple sont des LFOs (*Low Frequency Oscillator* - oscillateur basse fréquence) qui génèrent des mouvements répétitifs, ou des enveloppes, qui déclenchent une trajectoire déterminée à l'activation d'une note. Mais les DAWs permettent aujourd'hui de concevoir une trajectoire pour chaque

---

<sup>3</sup> un *buffer* est un conteneur de données, dans ce cadre là, d'échantillons audio

<sup>4</sup> On parle de procédé granulaire lorsque le son est découpé en grain : de petits échantillons allant de quelques millisecondes à plusieurs secondes.

variable, tout au long du morceau, on parle alors d'automation : c'est une des formes de modulation les plus courantes pour les utilisateurs de DAW.

Plus qu'une transition entre deux états, la modulation peut être conçue et composée comme un évènement sonore en soi. L'interaction entre plusieurs paramètres se modifiant simultanément est un terrain d'exploration stimulant et sans borne. Certaines sonorités ne peuvent être obtenues que par la manipulation d'un paramètre à travers le temps. Un des exemples les plus anciens est la modulation du temps de délai d'un effet écho : cela entraîne une variation de la hauteur du son contenu dans la ligne à retard<sup>5</sup> ou la bande analogique. Il est possible d'utiliser cet effet afin d'obtenir un son plus grave ou plus aigu, mais dans le cas d'un écho avec une boucle de rétroaction, la variation induit un effet sonore tout particulier qu'il serait plus ardu de reproduire en utilisant de la synthèse.

## 1.3 Plug-ins



Figure 1. Plug-ins de marques variées<sup>6</sup>

<sup>5</sup> Dans le cas où celle ci effectue une interpolation des échantillons interstitiels.

<sup>6</sup> <https://routenote.com/blog/how-to-install-logic-pro-x-au-plugins-without-restarting-your-mac/> [consulté le 16.04.2024]

### 1.3.1 Contexte technique

Les effets et synthétiseurs intégrés dans un DAW permettent de réaliser la grande majorité des opérations sur le son. Cependant, ils sont en général complétés par des *plug-ins* tiers, qui ont un traitement particulier du son ou une ergonomie différente.

La multiplication des *plug-ins* et leur variété nécessitent la mise en place de standards pour qu'ils puissent être utilisés dans des logiciels différents. Il existe plus d'une dizaine de ces standards à ce jour (Goudard, & Muller, 2003). Un des plus commun est le VST (*Virtual Studio Technology*), développé en 1996 par *Steinberg* pour leur DAW *Cubase*. Le standard VST permet par exemple à tous les hôtes de déterminer la catégorie du *plug-in* (synthétiseur ou effet), le nom du constructeur, d'envoyer et de recevoir de l'audio du *plug-in*, d'automatiser ses paramètres et de sauvegarder et rappeler des pré-configurations.

Ces fonctionnalités peuvent sembler triviales, mais il est toujours complexe d'accorder plusieurs entreprises sur un même fonctionnement. De plus, dans le domaine de l'audio en temps réel, une mauvaise gestion de la mémoire et des ressources processeurs peut rapidement impacter le comportement des autres logiciels. Une attention toute particulière doit donc être portée sur le développement de ces *frameworks* afin que tout les outils qui en découlent aient une gestion appropriée de ces problématiques. Les fonctions au cœur du traitement doivent être conçues solidement afin d'éviter tout artefact indésirable.

Ces standards s'accompagnent de SDK (*Software Development Kit* - kit de développement) offrant la possibilité à n'importe quel·le développeur·se<sup>7</sup> de créer de nouveaux outils. Aujourd'hui, le *framework*<sup>8</sup> JUCE<sup>9</sup> est le plus populaire, il permet d'exporter un même code vers la plupart des standards. En plus de ses capacités d'export, JUCE dispose d'un ensemble de code<sup>10</sup> pré-écrit et optimisé appelé librairies. Elles contiennent des fonctions préfabriquées, autant pour le traitement du

<sup>7</sup> À l'exception des SDK sous licence -comme le format AAX de Protools- qui ne peut-être utilisé que par les développeurs ayant été préalablement validé par l'entreprise.

<sup>8</sup> Abstraction permettant un accès plus simple à des fonctionnalités génériques, une base de travail simplifié.

<sup>9</sup> <https://juce.com/> [consulté le 21.03.2024]

<sup>10</sup> En général sous forme de "classe" contenant à la fois la structure de donnée et les fonctions qui les affectent, pour plus d'informations se renseigner sur la "Programmation Orientée Objet" (OOP)

signal (*buffer*, oscillateur, etc.) que pour la construction de l'interface (interrupteur, potentiomètre, etc.).

### 1.3.2 Conception

La manière dont sont conçus les instruments a fortement évolué avec les révolutions successives de l'électronique et du numérique.

L'interface des instruments acoustiques est intimement liée à la manière dont le son est généré (Hunt, & Wanderley, 2002). Un violon, par exemple, a des contraintes de construction strictes pour pouvoir résonner en sympathie avec la vibration des cordes qui doivent, elles, être suffisamment tendues et de la bonne longueur. Le lien entre l'interface et la génération du son se fait donc de manière implicite et engendre des réseaux de correspondances complexes. La question type pour mieux appréhender cette complexité pourrait-être : “où est le paramètre de volume sur le violon ?” (Hunt, & Kirk, 2000). Cette variable est déterminée par de nombreux paramètres : la pression de l'archet, sa rapidité, l'endroit où il frotte, la manière dont l'instrument est tenu, etc. À l'inverse, on serait bien en peine de définir exactement l'influence de la seule pression de l'archet sur le son produit. Tous les éléments d'interaction sont difficilement discrétilsables et ont une influencent les uns sur les autres.

Dans le cadre des instruments numériques, le son est généré par un ordinateur. La physique de l'oscillateur, qui était déterminante pour la forme des instruments acoustique, n'est donc plus une contrainte (Hunt, & Wanderley, 2002). Cette séparation ouvre de nombreuses opportunités de conception et une exploration indépendante de ce que peuvent offrir l'interface et le traitement du signal (Sramek *et al.* 2023). Néanmoins, cette liberté suppose une plus grande responsabilité dans la création des liens entre l'interface et le système de traitement, chaque lien entre l'interprète et l'ordinateur devant être conçu sciemment, avant que quoi que ce soit puisse être joué (Ryan, 1991). Ces liens, qui étaient jusqu'alors invisibles, se fondant dans le corps même de l'instrument, deviennent sujet d'analyse et de discussion. De même, le contrôleur (l'élément d'interface physique avec lequel l'instrumentiste interagit) doit être choisi. Il faut déterminer sa forme et les données qu'il récolte.

En résumé :

- l'**interface** est composée d'éléments visuels représentant une valeur, on les appelle *paramètres* (ou *widgets* dans le contexte d'une interface graphique).
- ces valeurs sont ensuite mises en forme par les *fonctions* du **mapping**<sup>11</sup>
- puis attribuées à une ou plusieurs *variables*<sup>12</sup> du **système de traitement**<sup>13</sup>.

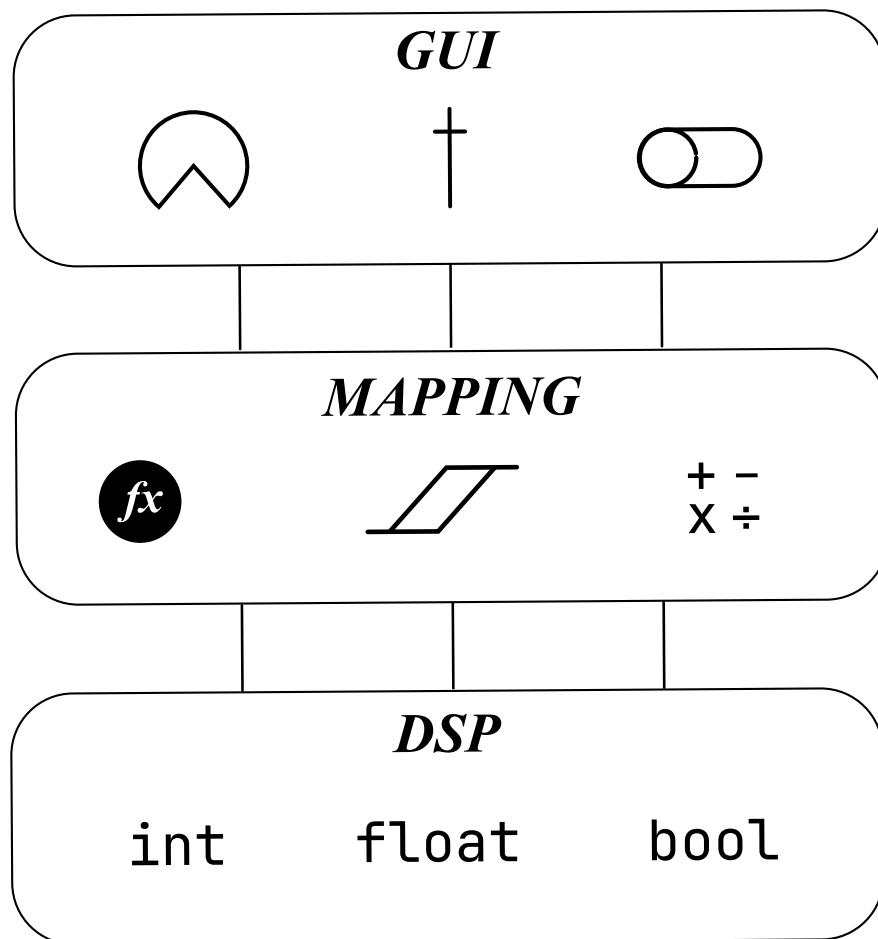


Figure 2. Architecture d'un plug-in

Bien que le DSP semble être le cœur du *plug-in*, son *mapping* et son interface sont tout aussi importants. La réflexion sur la manière dont ces trois pôles interagissent

<sup>11</sup> On l'appelle *mapping* en référence au terme mathématique définissant une fonction au sens général : qui assigne chaque élément d'un ensemble A aux éléments d'un ensemble B

<sup>12</sup> Terme issu de la programmation, une variable est une valeur pouvant varier pendant l'exécution du programme, à l'inverse des constantes qui elles ne varient pas

<sup>13</sup> Les dénominations paramètre / widget / fonctions / variables ne sont pas la norme, je les définis comme tel dans le cadre de ce mémoire pour rendre le propos plus clair.

est déterminante dans la conception de l'instrument. Établir l'impact de chaque partie séparément est compliqué.

Je proposerais ici un inventaire des problématiques inhérentes à chaque partie.

### 1.3.3 DSP

Le *DSP* détermine le résultat en sortie en fonction de la valeur de ses variables. Dans le cadre du développement, le *DSP* travaille dans une bulle fermée afin d'éviter tout problème lors du traitement de l'audio en temps réel, l'objectif fondamental d'un *plug-in* étant d'émettre du son (et le bon !). Il est donc par nature éloigné du *GUI* dont la mise à jour se fait en général sur un autre *thread*<sup>14</sup> du processeur. Les échanges de données entre les deux sont tout à fait possibles, mais nécessitent une bonne connaissance du fonctionnement interne de la machine pour éviter, par exemple, que l'un écrive à une adresse mémoire pendant que l'autre lit le contenu de cette même adresse. Des données corrompues peuvent, au mieux couper le son, au pire générer un son extrêmement fort.

### 1.3.4 Interface

L'interface, au sens large, inclut l'ensemble des “dispositif[s] intermédiaire[s] facilitant l'usage d'un système” (Vinet & Delalande, 1999).

La MAO utilise deux interfaces distinctes, l'interface logiciel (*GUI*), prenant la forme d'une fenêtre graphique permettant d'interagir avec des *widgets*, et l'interface physique, au minimum un clavier et une souris. On trouvera le plus souvent des contrôleurs supplémentaires, tel que le clavier MIDI : un clavier de piano générant des messages MIDI pouvant contrôler un synthétiseur physique ou virtuel. Il s'accompagne en général de quelques potentiomètres pouvant être assigné à des messages CC. Une littérature conséquente existe sur les contrôleurs expérimentaux,

---

<sup>14</sup> Un *thread* processeur est un cœur virtuel. Les programmes exécutant leurs commandes de manière séquentielle, il est nécessaire d'avoir plusieurs cœur de processeur pour pouvoir effectuer des tâches en parallèle et ainsi éviter qu'une tâche en bloque une autre.

dont le NIME<sup>15</sup> regorge d'exemples. Bien qu'intéressants, ils ne sont utilisés que par une poignée d'experts, car en général coûteux et complexes à prendre en main<sup>16</sup>.

J'étudierai en particulier le *GUI*, le clavier et la souris, afin de proposer des améliorations ayant un champ d'application le plus large possible. Les réflexions développées dans ce cadre pouvant ensuite être réutilisée avec des interfaces physiques plus complexes.

### Affordance

L'interface définira en grande partie les affordances de l'instrument, c'est-à-dire ce que l'utilisateur perçoit des possibilités offertes par l'objet, ce que l'objet invite à faire. Par exemple, une poignée de porte invite à appliquer une rotation du poignet, alors que les barres de sortie de secours invitent à être poussées (Gaver, 1991). L'affordance est à double tranchant : si elle est mal pensée, on tentera sans succès d'ouvrir la porte dans le mauvais sens. Un design soigneusement pensé dès la conception peut permettre d'éviter les mésusages et l'ajout d'informations textuelles surchargeant inutilement l'interface (tirer / pousser) (Norman, 1988). "Un design se basant principalement sur les caractéristiques d'une nouvelle technologie sera techniquement intéressant, mais fonctionnellement peu commode. À l'inverse, un design qui ne se base que sur les besoins actuellement exprimés par les utilisateur·rice·s passera peut-être à côté de nouvelles opportunités intéressantes"<sup>17</sup> (Gaver, 1991) [Traduction libre]. Le concept d'affordance est un outil de réflexion permettant de guider le processus de conception. Il encourage à considérer l'instrument en termes d'actions rendues possibles, évidentes. Il permet de ne pas se concentrer uniquement sur la technologie ou l'utilisateur·rice·, mais aussi sur l'interaction entre les deux (Gaver, 1991).

---

<sup>15</sup> New Interfaces for Musical Expression, <https://www.nime.org/> [consulté le 22.03.2024]

<sup>16</sup> La complexité n'est pas une chose à éviter en soi et amène son lot d'avantages. Néanmoins les interfaces NIME supposent souvent une virtuosité nécessitant plusieurs années d'apprentissage, et je cherche ici quelque chose de plus abordable.

<sup>17</sup> Texte original : “*Designs based primarily on the features of a new technology are often technically aesthetic but functionally awkward. But equally, designs based primarily on users' current articulated needs and tasks can overlook potential innovations suggested by new technologies.*”

## Skeuomorphisme

Les *GUIs* des *plug-ins* s'inspirent des interfaces physiques de leurs ancêtres. Les effets analogiques sont limités par des nécessités techniques et matérielles. Les unités sont souvent sous forme de *rack*<sup>18</sup> pour des questions pratiques, contraignant ainsi fortement les formes possibles de l'interface. La forme et la position des éléments d'interface sont en général dictées par l'électronique se trouvant derrière. Ainsi, on retrouve principalement l'interrupteur (*switch*) pour choisir entre plusieurs possibilités, le bouton pour déclencher des événements et faire des sélections, et le potentiomètre rotatif (*knob*) ou linéaire (*slider*) pour piloter une variable dans un circuit électronique. Dans la suite de ce mémoire, j'utiliserai les dénominations anglaises vu ci-dessus. Ces éléments d'interfaces, appelés *widgets* dans le contexte du développement logiciel, sont retrouvés tels quels dans les *plug-ins*. À ces quatre *widgets*, s'ajoutent les menus, hérités du *WIMP*<sup>19</sup>, et quelques éléments plus rares, comme les pads XY, qui permettent dans un seul mouvement d'interagir avec deux paramètres.



Figure 3. Roland JUPITER-8<sup>20</sup> (gauche), widgets par défaut de JUCE<sup>21</sup> (droite)

<sup>18</sup> Format de taille (largeur / hauteur / profondeur) et parfois d'alimentation électrique permettant de mettre facilement toutes les machines dans un même meuble, ne laissant voir que la façade sur laquelle sont les paramètres.

<sup>19</sup> Window Icon Menu Pointer : Fenêtre, Icône, Menu, Pointeur, c'est le paradigme d'interaction homme/machine dominant depuis sa popularisation par les premiers macintosh.

<sup>20</sup> <https://greatsynthesizers.com/en/review/roland-jupiter-8-the-great-master-of-synth-pop/> [consulté le 16.04.2024]

<sup>21</sup> capture d'écran issue d'un fichier de démonstration : <https://github.com/juce-framework/JUCE/blob/master/examples/GUI/WidgetsDemo.h> [consulté le 16.04.2024]

Une interface digitale est dite skeuomorphique quand elle émule un objet du monde réel en apparence ou en interaction (Interaction Design Foundation - 2016). Elles ont particulièrement été utilisées lors du début des ordinateurs de bureau afin de permettre aux utilisateur·rice·s non expert·e·s de retrouver leurs marques dans un monde dont ils ne connaissaient pas le fonctionnement technique. Les designer·e·s ont alors proposé des métaphores comme le bureau de l'ordinateur ou la corbeille, formes familières des environnements de travail matériel.

Le développement de l'audio numérique a suivi le même chemin. L'objectif était de rendre le studio plus efficace en le reproduisant numériquement. Le skeuomorphisme permettait aux ingénieur·e·s du son de réduire leur charge mentale et le temps d'apprentissage de ces nouvelles technologies (Kolb, & Oswald, 2014). Néanmoins, le skeuomorphisme n'a de valeur dans l'ergonomie et la vitesse d'apprentissage que si l'utilisateur a une connaissance de l'objet émulé. Les nouvelles générations ont découvert les environnements numériques en même temps que le monde réel. Pour ces "natifs numériques", l'idée que l'interface est apprise en transposant des connaissances du réel tend à perdre de son sens (Kolb, & Oswald, 2014) ; on est alors en droit de questionner la pertinence d'un tel choix pour l'interface (McGregor, 2019). "Même les personnes ayant dû apprendre ces nouvelles technologies ("immigrants numériques") sont aujourd'hui expertes et transposent leur connaissance des interfaces apprises précédemment plutôt que du monde physique"<sup>22</sup>(Kolb, & Oswald 2014) [traduction libre]. Certains chercheurs évoquent des raisons plus commerciales au maintien des interfaces skeuomorphiques : le visuel fait vendre et ces interfaces profitent aujourd'hui du prestige associé aux machines du passé et aux morceaux de musique qui ont été composés en les utilisant (Williams, 2015).

Il ne serait néanmoins pas judicieux de vouloir faire disparaître entièrement ces interfaces sous prétexte qu'elles brident les libertés créatives ou qu'elles sont "dépassées". Chaque type d'interface à sa pertinence, il existe autant d'instruments

---

<sup>22</sup> Texte original : *"For both types of users, the second generation of Digital Natives and the assimilated Digital Immigrants, the idea that interfaces are being learned by transferring knowledge from the 'real' (i.e. analogue) world to the digital world may loose its dominance. Experienced Digital Immigrants rather transfer knowledge they previously acquired using other interfaces, as opposed to employing knowledge they acquired interacting with the physical world."*

que d'artistes. Pouvoir essayer ces formes d'interactions, ne serait-ce que sous forme d'émulation logicielles, reste une expérience riche d'apprentissage. Néanmoins, s'en libérer semble nécessaire pour explorer certaines alternatives (Vinet & Delalande, 1999).

Il est important d'évoquer les liens de parenté des *GUIs*, car les origines des interfaces expliquent aussi en grande partie les limitations trouvées dans le *mapping*, qui sera le sujet central de ce mémoire.

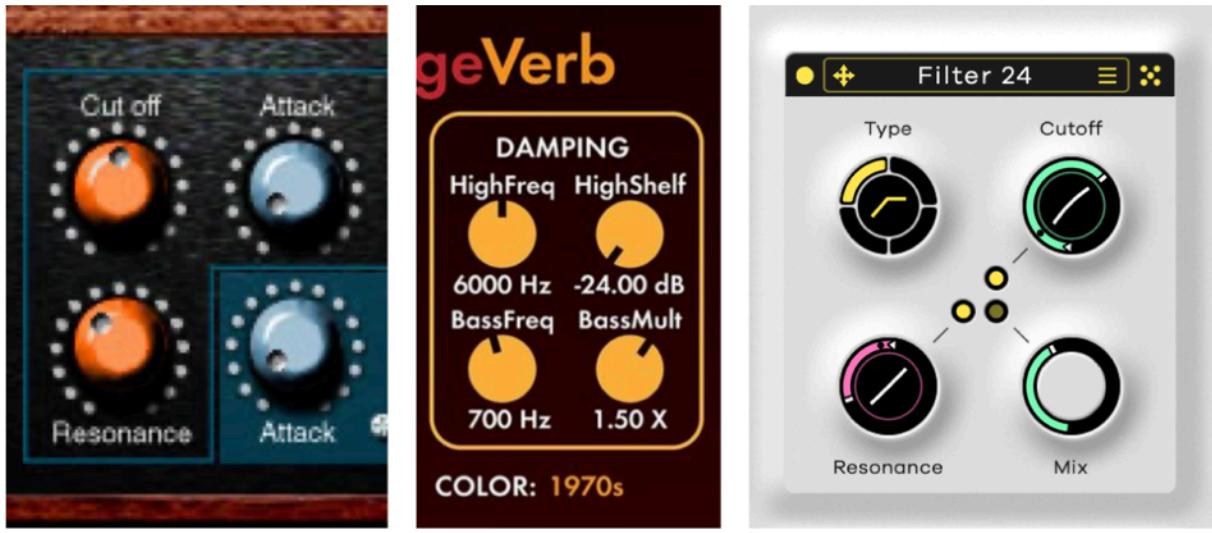


Figure 4. De gauche à droite :  
 Neon - Steinberg (1er instrument vst) 1999 - Skeuomorphism<sup>23</sup>  
 VintageVerb - Valhalla DSP - 2012 - Flat Design<sup>24</sup>  
 Transit - Baby Audio - 2023 - neumorphism<sup>25</sup>

Une interface skeuomorphique peut vouloir dire photo-réaliste, mais le DAW *Logic Pro* d'*Apple*, bien qu'il propose un GUI *flat-design*<sup>26</sup>, opère néanmoins selon des logiques skeuomorphiques, puisque le logiciel émule une console de travail analogique (envoi, tranche, insert, *fader*, etc.). Les outils passés influencent non seulement l'interface, mais aussi les logiques internes des instruments modernes. Les designs graphiques évoluent : avec le temps, on a vu apparaître les influences

<sup>23</sup> <https://www.kvraudio.com/product/neon-by-steinberg> [consulté le 16.04.24]

<sup>24</sup> <https://valhalladsp.com/shop/reverb/valhalla-vintage-verb/> [consulté le 16.04.2024]

<sup>25</sup> <https://babyaud.io/transit> [consulté le 16.04.2024]

<sup>26</sup> Design minimaliste et sans relief

du design web, comme le *flat design* ou le néomorphisme<sup>27</sup>, mais les fonctionnalités et les concepts d'interaction restent les mêmes.

### *Retour visuel et interface expérimentale*

On voit de plus en plus de *plug-ins* disposant d'un retour visuel de leur état de fonctionnement, en particulier lorsque leurs *mappings* utilisent un modèle physique (voir 3.6). Les *waveforms* de modulation éditables sont devenues répandues et se trouvent au centre de l'interface de plusieurs *plug-ins* à succès, par exemple *Shaperbox 3*<sup>28</sup> chez *CableGuys* ou *Infiltrator 2*<sup>29</sup> de *Devious Machine*.

Les visuels peuvent devenir plus évocateurs dans le cas de systèmes physiques, comme *Forest Grain* de Dillon Bastan, qui simule la croissance d'une forêt et l'utilise comme déclencheur pour la lecture de grains audio. Je citerais aussi *Codec* de *Lese*<sup>30</sup>, dont le visuel central ne modélise pas une variable particulière du moteur de traitement, mais évolue en fonction des conséquences du traitement sur l'audio.

Aujourd'hui, les objectifs de l'interface évoluent. Jesper Kouthoofd explique que certains instruments utilisent l'affichage comme source d'inspiration plus que comme source d'information (Bjørn, 2017, p.121). On trouve quelques marques qui explorent les formes que peuvent prendre les synthétiseurs physiques, comme chez *Ciat-Lonbarde*<sup>31</sup> ou *Destiny+*<sup>32</sup>. Plus anciennement, Rich Gold, le designer de la série “*Paper Face*” de *Serge Modular* (1972 - 1973) exprime explicitement son envie de se rapprocher de l'art, déplorant le fait que les synthétiseurs ressemblent à du matériel médical. Il relève aussi que le nom des notes n'est pas écrit sur les touches du piano. Il trouve donc curieux qu'autant d'explications figurent sur les synthétiseurs (textes, flèches, schéma, etc.) (Serge - Modulisme).

<sup>27</sup> Le néomorphisme est entre le flat-design et le skeuomorphism, il ajoute des relief par des jeux d'ombre sans adopter une esthétique photo-réaliste. Les coins sont arrondis et les objets semblent “collés” au fond.

<sup>28</sup> <https://www.cableguys.com/shaperbox> [consulté le 24.03.2024]

<sup>29</sup> <https://deviousmachines.com/product/infiltrator/> [consulté le 24.03.2024]

<sup>30</sup> <https://lese.io/plugin/codec/> [consulté le 24.03.2024]

<sup>31</sup> <https://www.ciat-lonbarde.net/> [consulté le 24.03.2024]

<sup>32</sup> <https://www.destiny-plus.com/> [consulté le 24.03.2024]



Figure 5. Un “Paperface” restauré<sup>33</sup>

Carla Scaletti, sur son logiciel de sound-design *Kyma*, fait elle aussi le choix conscient d'une interface abstraite, mais néanmoins plaisante à regarder, qui ne ressemble pas à un autre logiciel ou objet physique, afin d'éviter les associations d'idées qui pourraient limiter l'imagination des utilisateur·rice·s sur ce qu'il est possible d'accomplir avec ses outils. (Bjørn, 2017, p.296)

Il existe des logiciels qui explorent les interfaces graphiques (certaines listes répertorient les logiciels qui ont transformé la musique électronique dans les années 2000<sup>34</sup>) mais peu, voir aucun, ne sont passés à la postérité. Peut-être est-ce lié à l'obsolescence des systèmes d'exploitation, qui rend plus compliquée l'installation de logiciels “anciens” (à partie d'une dizaine d'années) (Goudard, 2020). En effet, sans le code source, il est très compliqué de “restaurer” un logiciel.

Je citerais néanmoins le travail d'Akira Rabelais sur *Argeiphontes Lyre*. Ce logiciel est une suite de morceaux de DSP à l'interface cryptique qui ne peuvent pas traiter les fichiers en temps réel. Ils imposent donc une forme d'itération ; on essaye un ensemble de paramètres, on traite le fichier audio, on écoute le résultat puis on recommence. Le logiciel va jusqu'à exploiter les fichiers d'installation comme support narratif : on y trouve un ensemble de textes, d'images, de sons, de PDFs, nommés avec des caractères empruntés à plusieurs alphabets, le tout participant à l'expérience étrange qu'est l'utilisation de ce logiciel.

<sup>33</sup> <https://djjondent.blogspot.com/2017/07/serge-restoration-paperface-21.html> [consulté le 24.04.2024]

<sup>34</sup> <https://dsparchive.neocities.org/>

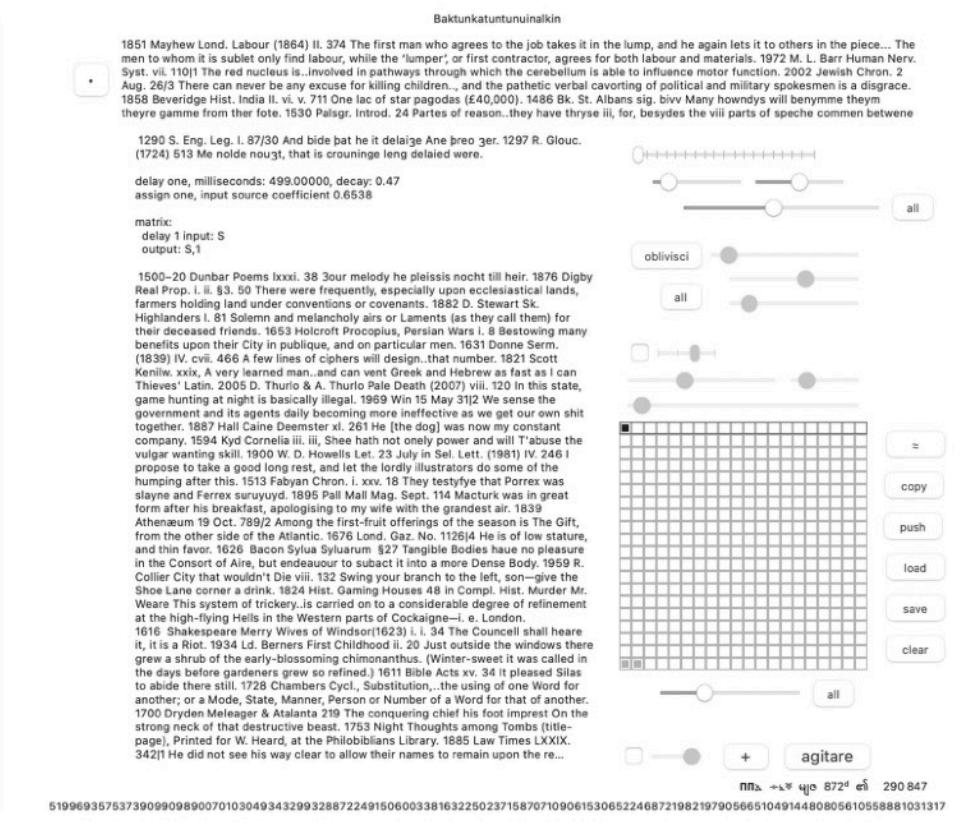


Figure 6. Un des Filtres de Argeiphontes Lyre (capture d'écran du logiciel)

### 1.3.5 Mapping

Le *mapping* se trouve entre le DSP et l'interface. C'est un ensemble de fonctions, dont la forme la plus simple serait  $f(x) = x$ . Ces fonctions prennent en entrée la valeur des *widgets* de l'interface et attribue leurs valeurs de sortie aux variables du DSP. Les limites entre paramètre, fonction et variable sont débatables : on peut considérer que chaque *widget* a une valeur normalisée entre 0 et 1 qu'il transmet au *mapping*, mais les fonctions élémentaires de celui-ci (voir 3.1) sont souvent implémentée à même le paramètre, comme la valeur minimale et maximale qu'il peut prendre. De même, la limite avec les variables de traitement du signal est complexe. Dans le cas d'un filtre biquadratique<sup>35</sup> par exemple, l'objectif du *mapping* est de déterminer la

<sup>35</sup> Une filtre biquadratique est un filtre numérique qui modifie le contenu en fréquence du signal par une sommation avec des versions retardés de lui-même.

fréquence de coupure, mais celle-ci est elle-même une variable permettant de déterminer plusieurs autres coefficients de calculs. On pourrait alors considérer que la fonction permettant de déterminer ces coefficients fait aussi partie du *mapping*, que la fréquence n'est qu'une valeur intermédiaire (Goudard, & Muller, 2003) (on pourrait par ailleurs questionner la légitimité d'interagir directement avec ces coefficients plutôt qu'avec la fréquence de coupure).

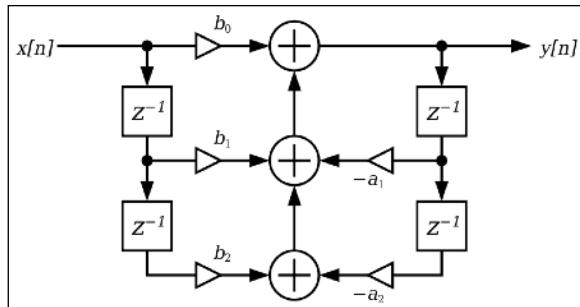


Figure 7. Schéma de filtre biquadratique, les z sont des retards, les a et b des coefficients<sup>36</sup>

Les différents contextes techniques ont amené de nouvelles manières de penser le *mapping* : en analogique, pour qu'une variable puisse être modifiée, il est nécessaire qu'un paramètre existe physiquement (même dans le cas du réglage de l'instrument, il n'est pas rare que les concepteurs interagissent avec des *trimpots*<sup>37</sup> directement soudés sur le circuit et non-accessibles depuis l'interface).

Le numérique a apporté un découplage encore plus grand, car le *mapping* peut être modifié pendant le jeu. Ceci permet à certains synthétiseurs numériques d'avoir des systèmes de traitement très complexes et des interfaces très simples, puisqu'un seul *slider* peut contrôler toutes les variables du synthétiseur en étant réaffecté successivement à chacune d'entre elles. Une interface minimale peut néanmoins être laborieuse à utiliser : devoir sans cesse choisir quel paramètre on désire manipuler avant de pouvoir effectivement le modifier peut être désagréable si l'interface n'est pas très bien pensée.

Une analogie revient régulièrement dans les articles sur le *mapping* : les possibilités sonores peuvent être vues comme un espace (Wessel, 1979)(Garnet, & Goudeseune, 1999)(Mudd, 2023). Le système de traitement du signal définit l'aire de

<sup>36</sup> [https://en.wikipedia.org/wiki/Digital\\_biquad\\_filter](https://en.wikipedia.org/wiki/Digital_biquad_filter) [consulté le 16.04.2024]

<sup>37</sup> Petit potentiomètre, manipulé avec un tournevis

cet espace, l'ensemble des possibles. L'interface et le *mapping* permettent d'effectuer une curation de ces possibles, de sélectionner les lieux intéressants et de définir la manière de se déplacer d'un endroit à un autre, c'est-à-dire définir l'ensemble des états intermédiaires du timbre et modeler le son que va produire le mouvement. Dans la MAO, comme précisé plus haut, la manière de se déplacer est tout aussi importante que le lieu de départ et la destination. Un *mapping* bien pensé peut faciliter ou guider vers des mouvements complexes auxquels on n'aurait potentiellement pas pensé.

On ne peut pas se reposer sur la technologie comme seule base pour un instrument (Daniel Trodberg dans Bjørn, 2017, p.74). L'interface doit être considérée comme au moins aussi importante que le DSP, afin de sélectionner avec soin les affordances de l'instrument. Le *mapping*, en particulier, peut vraiment faire la différence dans la façon dont les gens perçoivent le "son" du *plug-in* (Goudard, & Muller, 2003).

## 1.4 Presets

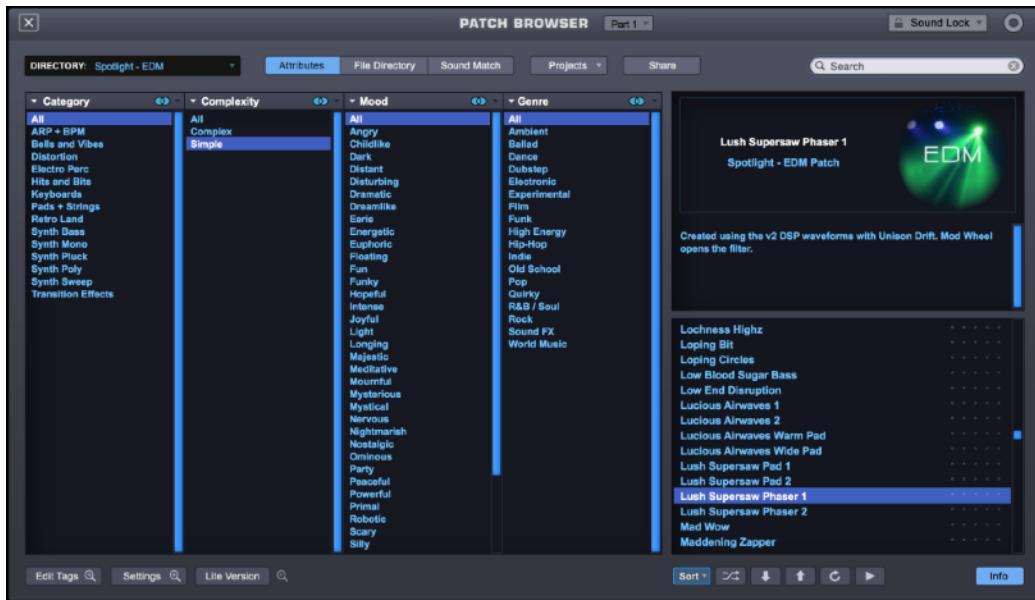


Figure 8. Le “patch browser” de Omnisphere, synthétiseur réputé pour ses banques de preset<sup>38</sup>

Lors de la composition musicale, plonger dans les dizaines, voire centaines de paramètres disponibles sur un synthétiseur peut se révéler extrêmement chronophage. Les paramètres sont techniques, parfois opaques et nombre de combinaisons ne produisent pas de son. Certain·e·s compositeur·rice·s iront alors chercher une pré-configuration ressemblant à l’idée qu’elles.ils ont en tête. Pour la sélectionner, l’interaction passe par des menus déroulants contenant des listes de noms plus ou moins évocateurs. Il est en général possible de filtrer selon certains critères comme la fonction musicale recommandée : /lead, basse, nappe, etc. Les compositeur·rice·s itéreront sur plusieurs possibilités jusqu’à trouver ce qui leur convient. Conçus par des sound-designeur·e·s, les *presets* font partie des arguments de vente lors de l’achat d’un synthétiseur. En effet, pour une personne n’ayant pas les connaissances nécessaires pour construire elle-même un son, les possibilités offertes par l’instrument dépendent uniquement du nombre de *presets* disponibles. Qu’ils soient utilisés ou non, ils sont omniprésents dans l’utilisation d’un DAW. Les standards prennent en compte leur utilisation : ils sont notamment implémentés dans le MIDI et le VST.

<sup>38</sup> <https://support.spectrasonics.net/manual/Omnisphere2/25/en/topic/browser-index>

### 1.4.1 Histoire des presets

#### Synthétiseur acoustique

Les *presets* sont aussi vieux que les synthétiseurs : sur les orgues (qu'on peut considérer comme des synthétiseurs additifs acoustiques), on trouve déjà des “registres”. Ce sont des poignées qui, lorsqu'on les tire, vont ouvrir et fermer plusieurs tuyaux simultanément afin que la somme des harmoniques imite l'instrument correspondant.



Figure 9. Registres de l'orgue Yves Fossaert<sup>39</sup>

#### Synthétiseurs analogiques

Aux débuts de la synthèse analogique, la technique ne permet pas d'avoir des *presets* inclus dans les instruments. Ils sont en général accompagnés de feuilles de *patch*<sup>40</sup>, indiquant la valeur que doit prendre chaque paramètre pour reproduire un son donné. On trouve des exemples intéressants chez ARP, constructeurs d'instruments électroniques, qui ont toujours eu une volonté de pédagogie. Le manuel du *ARP 2600* (synthétiseur sorti en 1971) précise que les schémas ne sont que des guides et nous invitent à rester flexibles (*Arp 2600 patch book*).

<sup>39</sup> <https://orgueetmusiqueavouvant.com/l-orgue-fossaert/la-composition-de-l-orgue/> [consulté le 16.04.2024]

<sup>40</sup> Dans le contexte de la synthèse, un *patch* est un synonyme de *preset* (on trouve aussi “*program*”). Les synthétiseurs devaient historiquement être “patché”, c'est-à-dire brancher des modules de fonctionnalité élémentaire les uns avec les autres pour pouvoir générer du son. Un “*patch*”, est donc un ensemble de branchements résultant en un son reproductible.

Peu à peu, on commence à trouver des “synthétiseurs à *preset*”, c'est-à-dire des synthétiseurs dont les paramètres sont réduits au minimum, disposant d'interfaces similaires aux orgues : on pousse un bouton pour enclencher un son pré-conçu.

Avec le développement du numérique, les synthétiseurs analogiques vont pouvoir inclure une couche de contrôle permettant de sauvegarder et de rappeler des configurations. On peut par exemple citer le *Prophet 5* de *Sequential*, ou le *OB-8* de *Oberheim*. Ces synthétiseurs sont prisés pour la performance : la capacité d'enregistrer les configurations de chaque morceau et de les rappeler à la volée étant particulièrement importante sur scène où le temps est limité.



Figure 10. Détail du ARP pro Soloist, un des premiers synthétiseurs à *preset*, sorti en 1972<sup>41</sup>

### Synthétiseurs numériques

Lorsque les systèmes de traitements commencent à se numériser aussi, il devient techniquement compliqué d'afficher tous les paramètres en façade.

Sorti en 1983, le *DX7* de *Yamaha* est un des synthétiseurs les plus emblématiques de cette époque. Il est le premier à implémenter la synthèse par modulation de fréquence (Chowning, 1977). Ce nouveau type de synthèse, réputé rigoureux et complexe, couplé à une interface très minimalistique, rend le synthétiseur compliqué à patcher. Beaucoup d'utilisateur·rice·s se contentent des banques de *presets* conçues par les sound-designeur·e·s engagé·es par Yamaha. Ces *presets* seront entendus dans des centaines de titres durant les années 80 et 90. Le *DX7* débutera une vague de synthétiseurs monolithiques qui, malgré des moteurs de synthèses très différents et souvent novateurs, sont pratiquement identiques du point de vue de l'interface.

<sup>41</sup> <https://syntaur.com/gallery.php?keyboard=2236> [consulté le 16.04.24]

Quelques grands noms : *Korg M1*, *Roland D50* ou encore *Kawai K1* (voir images en annexe p.85).

Avec les *presets* stockés sous forme numérique (sur cartouche), il devient plus facile de partager et de vendre des *presets*. Sound-designer·e<sup>42</sup> devient un métier à part. Elles.ils produisent des configurations pour que les musicien·ennes puisse jouer tout de suite, sans perdre le temps de configuration du synthétiseur qu'imposait l'analogique.

Le *preset* n'est plus une direction, une estimation en degré de la position d'un potentiomètre rotatif, mais bien un instrument figé, reproductible à l'identique à la seule condition de disposer du même synthétiseur.



Figure 11. Cartouches de presets pour le DX7<sup>43</sup>

## *Synthétiseurs logiciels*

À ses débuts, le processus de synthèse par ordinateur est asynchrone car les machines de l'époque ne sont pas assez puissantes pour générer du son en temps réel. *MUSIC* de Max Mathews est un des premiers langages de programmation permettant d'interagir musicalement avec les ordinateurs. On pourrait considérer les premières librairies de DSP comme des formes de *presets*, permettant d'obtenir plus vite un résultat lors de la programmation.

<sup>42</sup> Au sens de concepteur de preset, le sound-designer concevant les sons (par exemple pour le cinéma) étant plus ancien. Bien qu'une personne faisant l'un fait souvent l'autre, ce sont deux tâches avec des objectifs et des méthodes différentes.

<sup>43</sup> <https://yamahablackboxes.com/collection/yamaha-dx7-synthesizer/patches/> [consulté le 16.04.2024]

Le premier instrument VST ayant vu le jour est *Neon*, de *Steinberg*, sorti en 1999, il contient déjà des *presets* malgré ses 14 paramètres (très peu à l'échelle des synthétiseurs).

Le concept de *preset* est intégré dans le *SDK* des VSTs. L'hôte est censé offrir la possibilité de les enregistrer et de les rappeler (Goudard, & Muller, 2003). La plupart des développeur·se·s de *plug-ins* fournissent leurs propres menus, ce qui a deux avantages : une meilleure flexibilité dans l'apparence et les fonctionnalités, ainsi qu'une recherche directement intégrée dans l'outil, ce qui permet un processus de travail plus fluide.

#### **1.4.2 *Plug-ins* hôtes et macro-contrôles**

Une marque pouvant commercialiser plusieurs dizaines de synthétiseurs, chacun contenant plusieurs milliers de *presets*, on trouve aujourd'hui des " *plug-ins hôtes*" compilant tout ces *presets* au sein d'un même logiciel. *Analog Lab* de *Arturia* et *Komplete Kontrol* de *Native Instrument* en sont les deux principaux exemples. Ces logiciels sont pensés pour être utilisés avec les claviers maîtres associés. La série des *Keylabs* chez *Arturia* et des *Komplete Kontrol* chez *Native Instrument*.

Le centre du *GUI* de ces *plug-ins* est une fenêtre de recherche dans laquelle on peut filtrer de diverses manières : type de son, instruments, sound-designer·e ou encore style musical. Afin d'améliorer la fluidité d'utilisation, charger un *preset* de synthétiseur n'affiche pas l'interface graphique de celui-ci, seuls les sons changent. Pour adapter l'interaction en fonction des instruments, on trouve un ensemble de potentiomètres et de *sliders* appelés " *macro-knobs*" ou " *performance controls*" (*Analog Lab V User Manual*, 2021). Ils sont pré-assignés aux potentiomètres physiques des claviers maîtres. Chaque *preset* utilise ces *macro-knobs* pour mettre en avant les paramètres du synthétiseur qui semblent les plus pertinents. On y retrouve souvent la fréquence de coupure du filtre qui a un impact drastique, ainsi que certains paramètres très pratiques pour façonner rapidement un son et l'adapter à nos besoins comme le volume, le temps d'attaque et de *release*<sup>44</sup>.

---

<sup>44</sup> Durée du son après avoir relâché la note.

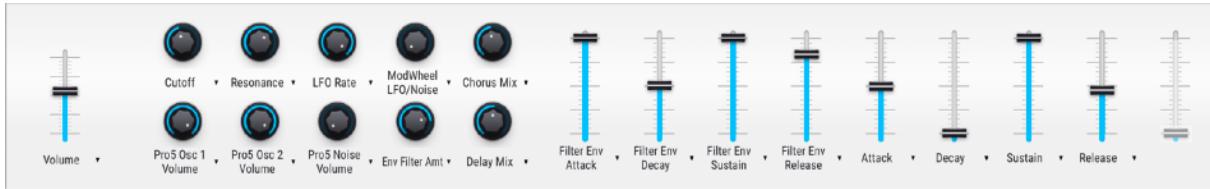


Figure 12. Macro contrôles d'un preset de Analog Lab (capture d'écran)

Dans le manuel de *Komplete Kontrol*, on peut lire que le choix des paramètres macros des instruments “est réalisé par ceux qui les connaissent le mieux – les concepteurs des instruments eux-mêmes.” (Komplete Kontrol User Manual, 2023).

Malgré le fait qu'ils soient disponibles sur la majorité des grands synthétiseurs *software* moderne comme *Vital*, *Pigment*, *Serum*, *Massive* ou *Phaseplant*, les *macro-knobs* restent peu exploités, et rarement pour inviter à l'exploration et à la découverte. Souvent, ils répliquent seulement l'un des paramètres du synthétiseur, permettant d'y accéder depuis le *plug-in* hôte, et simplifient les termes techniques en les remplaçant par des termes perceptifs (la fréquence de coupure du filtre est parfois retrouvée sous le terme de “brillance”). Le minimalisme de ces interactions pose des questions d'expressivité timbrale pour les artistes qui n'ont pas l'envie ou le temps de construire eux-mêmes leurs paramètres de contrôle.

Pourtant, les macro-paramètres sont une opportunité pour les sound-designeur·e·s de proposer de véritables “micro-instruments”, qui ne seraient pas qu'un point dans l'espace des possibles, mais bel et bien une zone d'exploration, avec des mouvements et des déplacements plus abouti. En effet, le *macro-knob* est aussi une proposition de modulation de la part des sound-designeur·e·s, qui connaissent l'instrument qu'elles. Ils viennent de concevoir et peuvent nous guider vers de potentiels événements et mouvements sonores.

Aujourd’hui, le *preset* est un marché florissant évoluant parallèlement aux *plug-ins*. Ce qui était avant une cartouche est maintenant un dossier de fichiers, d’autant plus simple à télécharger et à échanger. Les boutiques en ligne vendent des packs de *presets* de la même manière qu’ils vendent des synthétiseurs et des effets. *Phaseplant* et *Vital* proposent des abonnements permettant de recevoir les nouveaux packs dès leur sortie. On peut vite se perdre dans l’infinité de *presets* à disposition. À terme, la diversité de l’offre est telle qu’on peut se demander si les *presets* ne déplacent pas le problème d’une difficulté à construire un son vers une difficulté à choisir un son.

De plus, alors que les synthétiseurs logiciels ont largement évolué, l’utilisation de *presets* a, de son côté, très peu changé depuis *Neon*. On a ajouté des filtres de sélection ainsi qu’une couche d’abstraction de l’interface par la mise en place de macro-paramètres facilitant l’utilisation. Les *presets* conservent de nombreuses limites que je vais détailler afin de pouvoir établir les potentiels points d’amélioration.

# **Limites et réflexions**

## 2.1 Les *presets* et leurs limites

*The only limiting factor is your imagination, but that is a tremendous limitation*  
-mylarmelodies

Une interface contenant de nombreux paramètres n'est pas toujours bénéfique pour les processus créatifs et peut créer de la frustration. Même pour un.e utilisateur·rice expert·e, de trop grandes possibilités peuvent entraver leur capacité à explorer librement (Sramek *et al.* 2023). La loi de Hick établit par exemple que plus nous avons d'options à disposition, plus le temps de décision est long, ce qui génère une « fatigue décisionnelle » (Bjørn, 2017, p.149). Néanmoins, une interface simple ne devrait pas impliquer des fonctionnalités pauvres (Spagnolli *et al.*, 2020). Levitin D. J. *et al.* (2002) ajoutent qu'un instrument devrait chercher l'équilibre entre le challenge, la frustration et l'ennui, entre une facilité d'utilisation initiale et une complexité permettant un développement de l'interaction dans le temps. Bien qu'une prise en main simple soit clairement un avantage pour les systèmes interactifs, une intuitivité absolue semble être une illusion : aucun système n'est fondamentalement plus naturel que les autres.

Malgré leur aspect pratique, on pourrait objecter que les *presets* limitent la créativité de leurs utilisateur·rice·s. Ils permettent effectivement une prise en main extrêmement simple, mais peu de développements une fois le son sélectionné. La place des *presets* est très importante pour les entreprises qui conçoivent les *plug-ins* car ils permettent de cibler le public. Certains synthétiseurs se retrouvent associés pour un style musical spécifique malgré le fait qu'ils offrent des possibilités allant bien au-delà (Goldmann, 2019). “*Shaping the character of an instrument is a task that has moved to the sound-design department*”<sup>45</sup> (Goldmann, 2019). Reprenons l'exemple du DX7 : son fonctionnement permet un nombre impressionnant de possibilités, explorées aujourd'hui encore par des artistes. Pourtant, les *presets* fournis à la sortie de l'usine sont principalement des émulations d'instruments acoustiques. Dans la

---

<sup>45</sup> [traduction libre] : “Modeler le caractère de l'instrument est une tâche qui s'est déplacée à l'étape du sound-design”.

banque 1, pour 32 *presets* on trouve 2 “FX”, qui sont des reproductions d'un son de train et d'avion, les 30 autres sont des reproductions d'instruments acoustiques<sup>46</sup>.

De plus, le processus, qui est en lui-même une source de créativité, est ici très limité : le modèle d'interaction étant le clic dans des menus déroulant. La sélection de *presets* laisse peu de place à l'erreur ou au détournement qui pourrait venir nourrir la sérendipité (“acte de trouver sans chercher en usant de sagacité”, Nirchio, 2018), et permettre à l'utilisateur de découvrir des idées musicales qu'il n'aurait pas imaginées avant (Kirkegaard, *et al.* 2020).

De la même manière, pour quelqu'un n'ayant pas la connaissance de l'effet de chaque paramètre, les possibilités de modulation d'un *preset* sont limitées à ce que la.le sound-designeur·e aura choisi de rendre accessible par les *macro-knobs*.

On pourrait conclure qu'il serait bon de trouver une manière de rendre les *presets* plus exploratoires, afin qu'ils laissent plus d'opportunités de jeu et d'interactions, qu'ils offrent plus de surprises et de résultats inattendus dans les timbres et les modulations proposées.

Dans cette réflexion, il est néanmoins important de séparer plusieurs phases dans l'élaboration d'une composition électronique. Une première phase d'exploration dans laquelle la surprise est bienvenue à laquelle succède une phase dans laquelle on cherche une reproductibilité plus efficace (Gelineck, & Serafin, 2009). J'étudie principalement l'exploration des timbres, et on pourrait relever que certains genres musicaux se concentrent sur le rythme et l'harmonie et ne cherchent pas nécessairement une complexité dans l'évolution des timbres. De manière générale, un même outil peut être vital pour un.e utilisateur·rice et inutile, voir contre-productif, pour un.e autre (Bjørn, 2017 p.23). Même dans le contexte des musiques électroniques, les processus de travail sont très variables selon les compositeur·rice·s. Un *preset* n'est pas toujours choisi comme source de sons à explorer, il est parfois sélectionné avec un objectif précis en tête. Dans ce cas, les fonctionnalités permettant de rendre le *preset* exploratoire ne seraient pas nécessaires. Ils pourraient même venir entraver le travail des compositeur·rice·s s'ils

---

<sup>46</sup> L'ensemble des banques du DX7 sont compilées sur ce site : <https://yamahablackboxes.com/collection/yamaha-dx7-synthesizer/patches/>

sont la seule méthode d'interaction et ne permettent pas la précision et la reproductibilité requise.

J'émets l'hypothèse qu'une amélioration des *presets* permettrait de bénéficier d'une meilleure phase d'exploration des timbres dans la composition de musique électronique et de mieux appréhender ce que des modulations complexes du timbre peuvent offrir.

Une prise de conscience sur ces possibles pourrait permettre l'émergence de "micro-instruments", ayant une utilisation simple, une diversité de timbre limitée, mais dans lesquels on retrouverait une part des possibilités d'explorations et d'interactions propres aux instruments acoustiques.

## 2.2 Agentivité : contextes et contraintes

L'agentivité (dont la forme anglaise "agency" est plus répandue) désigne la capacité d'agir sur soi-même, sur l'autre et sur son environnement (Jézégou, 2022). D'abord centré sur l'humain, le concept s'ouvre peu à peu pour englober plus généralement n'importe quel "objet", qu'il soit humain, immatériel, durable ou éphémère (Harman, 2015). Ce concept est un outil de réflexion né de la philosophie. Il est utilisé dans la recherche d'interfaces homme/machine (Frauenberger, 2019) et apparaît plus récemment dans la recherche sur les outils de création sonore (Sramek *et al.* 2023). Il vise à mieux comprendre notre lien avec ce qui nous entoure et à prendre conscience de l'ensemble des acteur·rice·s impliqué·e·s dans un phénomène donné. "L'idée d'un.e musicien.ne "utilisant" un "outil" ne tient pas intrinsèquement compte de la bidirectionnalité de cette interaction, qui façonne et guide la création du produit final au-delà de ce qui peut être perçu par les descriptions fonctionnelles de l'un ou de l'autre"<sup>47</sup> (Rodger *et al.*, 2020) [traduction libre]. On pense au concept de *Gestalt* : l'ensemble est plus que la somme de ses parties (Bjørn, 2017, p.12).

L'instrument de musique suppose un contexte socio-culturel qui va en grande partie définir son sens dans la composition et ses méthodes d'utilisations (Rodger *et al.*,

---

<sup>47</sup> Texte original : "*The idea of a musician 'using' a 'device' does not inherently accommodate the bidirectional influence of musicians with their musical and interpersonal environments, which shape and guide the interactions between musician and instrument beyond what can be captured by functional descriptions of either alone.*"

2020). On ne joue pas d'un instrument de la même manière selon le genre musical, bien que l'instrument reste le même.

J'évoque ici les *presets* de synthétiseur mais on pourrait étendre l'usage du mot à toutes les décisions pour lesquelles on fait usage d'un concept pré-fabriqué, comme une gamme ou une rythmique. Faire usage ou non d'une pré-configuration est un geste créatif en soi qui vient modeler le résultat. Dans la MAO, le travail n'est plus un prérequis, mais un choix créatif (Goldmann, 2019). En facilitant certains processus, comme le *sampling*<sup>48</sup> ou l'utilisation de fichiers MIDI issus d'une musique préexistante, les DAWs rendent plus évident un processus qui a toujours existé : une musique est construite sur des bases culturelles. On ne peut pas attendre d'un.e compositeur·rice qu'elle·il soit créatif sur tous les pans de sa musique (la créativité, le fait de créer de la nouveauté, n'étant pas non plus une nécessité).

L'agentivité est particulièrement intéressante à étudier sur les instruments de musique : dans le contexte de l'art où la parenté d'une idée peut-être un sujet sensible, quel sont véritablement les acteur·rice·s qui déterminent le résultat sonore ?

- l'artiste guidé par l'utilisation du preset ?
- les sound-designer·e·s guidé·es par ce qu'il est possible de faire avec le synthétiseur ?
- les développeur·se·s guidé·es par ce qu'il est possible de faire avec les librairies de programmation ?
- les développeur·se·s programmant les librairies de programmations guidé·es par les paradigmes d'interaction actuels et par le fonctionnement des ordinateurs tel qu'il est aujourd'hui ?

Cette mise en abîme ne semble pas avoir de fin. La suite de décisions menant au sonore est un maillage d'interactions complexes.

Chacun·e crée à l'étage où elle·il est le plus à l'aise. Chaque niveau d'abstraction offre des opportunités d'interaction et des processus intellectuels différents dans ce millefeuille d'interfaces où chacun·e crée les *presets* de l'utilisateur·rice suivant.e.

---

<sup>48</sup> Utilisation d'un extrait sonore issu d'un enregistrement préexistant. Plus largement, utilisation et manipulation de fichier audio.

À toute ces *affordances* techniques viennent s'ajouter les *affordances* culturelles : ce que nous choisissons de faire et ce que nous faisons est largement conditionné par ce que nous avons appris, qui est ensuite filtré et façonné par ce que nous avons observé comme faisable et “autorisé” par nos pairs (Rodger *et al.*, 2020).

La démocratisation des “outils pour développer des outils” et la mise en place de ce spectre de métiers aux limites mouvantes a participé à rendre floue la frontière entre les compositeur·rice·s et les technicien·ne·s, entre les artistes et les concepteur·rice·s, en particulier dans la musique électronique (Bjørn, 2017, p.18). Au même titre qu’un ou une artiste qui publie une musique, la personne qui conçoit un outil émet une proposition qui sera reçue ou non par le public. Pour certains, concevoir les instruments est aussi stimulant que de les utiliser. Il est important de prendre en compte l’ensemble de ce système et ses complexités lors de la proposition d’une nouvelle manière de concevoir des outils : on ne pourrait se contenter de centrer notre réflexion sur les utilisateur·rice·s sans penser aux implications pour les personnes en amont.

Avec un prisme technique, on peut voir l’agentivité comme une manière de penser les interactions entre les trois principales composantes du *plug-in*. L’interface est en général subordonnée aux décisions ayant été prises lors de l’élaboration du DSP, l’objectif étant de contrôler le traitement en ayant le moins de friction possible dans l’interaction. Subordonner le traitement à l’interface pourrait nous ouvrir à une manière antithétique de construire l’outil, nous amènerait peut-être vers des résultats nouveaux, à la fois pour l’interface et le traitement. Chaque composante peut être un point de départ. *Mapping*, interface et DSP peuvent tous les trois être sujet d’innovations et d’expérimentations.

## 2.3 Outil ou instrument ?

Je placerais la frontière entre outil et instrument dans la nécessité ou non de contrôle sur le résultat final. On attend d’un outil qu’il se plie à un objectif déterminé, qu’il remplisse sa tâche efficacement ; on l’utilise avec une fin précise en tête. Il est par contre impossible de définir clairement l’objectif d’un instrument, de même qu’il est impossible de définir clairement l’objectif d’un acte artistique. Du fait cette

indéfinition, l'utilisation d'un instrument ne répond pas uniquement à la volonté d'obtenir un produit précis, elle vise aussi à faire l'expérience d'une interaction : on se laisse plus volontiers influencer par les affordances d'un instrument.

Un outil contenant plus de fonctionnalités que nécessaire sera un outil moins efficace, là où les fonctionnalités inattendues d'un instrument sont acceptées avec curiosité.

Cette frontière, comme toute limite entre deux définitions, est mouvante. Un objet est rarement seulement un outil ou seulement un instrument, sa véritable nature ne lui est attribuée que par l'interaction avec une volonté, qui sera en retour influencée par les affordances de l'objet.

Reprendons l'exemple de *Argeiphontes Lyre* mentionné plus haut (voir figure 6). Ce type de projet, qui se rapproche de l'œuvre interactive, nous pousse à nous interroger sur les raisons pour lesquelles nous interagissons avec les instruments de musique : en quoi est-ce que l'ajout d'une narration, la construction d'une sorte de récit, de mythe, influence la manière dont on va utiliser l'instrument, et quelle est la part "d'objectif à remplir" et la part de "fin en soi" dans la création musicale ? On ne compose pas uniquement pour obtenir un produit sonore, un fichier audio, mais aussi pour l'épanouissement intellectuel et émotionnel que peut apporter l'acte de composition. Chacun·e a son propre rapport et son propre équilibre entre le désir de composer et le désir de produire un objet sonore.

Les *presets* sont-ils des outils ou des instruments ?

La question est d'autant plus épineuse dans le contexte particulier de l'utilisation de l'ordinateur. À l'origine, l'ordinateur personnel est surtout un ordinateur de bureau dont l'objectif principal est l'augmentation de la productivité des salariés. Une autre étude sur l'influence de ce passé sur la perception que les artistes ont de cet outil-instrument permettrait de faire évoluer cette réflexion.

Bien qu'il soit nécessaire que les *presets* puissent conserver leur statut d'outil pour une partie des utilisateur·rice·s, il pourrait être bénéfique pour certain·e·s artistes de leur ajouter une dimension instrumentale.

## 2.4 Sources d'amélioration potentielles

Des trois parties d'un *plug-in* : traitement, interface et *mapping*, quelles sont celles qui pourraient être modifiées pour permettre de rapprocher l'outil de l'instrument ?

Il semble important de mener cette réflexion avec le contexte de création bien en tête : on ne saurait proposer une solution dont les prémisses seraient la refonte de l'entièreté du paradigme de production actuel.

### 2.4.1 DSP

De nombreuses recherches ont été (et sont) effectuées sur les algorithmes de synthèse. On peut citer par exemple la synthèse additive, soustractive, à modulation de fréquence, par distorsion de phase, granulaire, par concaténation, etc. Ces recherches sont le plus souvent tournées vers les détails techniques d'implémentation et l'exploration des possibilités de l'algorithme. Elles laissent en général le soin aux autres de développer les interfaces nécessaires aux interactions. On trouve pléthore de données et d'exemples sur l'implémentation des divers systèmes de traitement. L'enjeu, aujourd'hui, porte plus sur la recombinaison astucieuse de systèmes déjà connus que sur la découverte d'un nouveau type de synthèse ou d'effet.

Le problème ne se situe pas non plus au niveau de la puissance des ordinateurs : les limites de temps réel sont loin derrière nous et le son numérique est largement maîtrisé par les processeurs actuels (qui trouvent leurs limites dans le traitement de l'image et le jeu vidéo). Les marges d'amélioration se trouvent plutôt vers les pilotes et la gestion audio des systèmes d'exploitation qui n'ont pas toujours le son comme priorité : ce sont des solutions extrêmement générales, conçues pour le plus grand nombre.

Le *DSP* est, de plus, largement découplé de la gestion des *presets*, qui vient souvent se greffer à n'importe quel type de synthétiseur et d'effet de la même manière.

## 2.4.2 Interface

L'interface semble être un terrain fertile pour la future recherche. Sans même évoquer les interfaces physiques NIME, de nombreuses possibilités sont à expérimenter avec le matériel déjà accessible. On peut par exemple imaginer des *plug-ins* qui sont contrôlés au clavier plutôt qu'à la souris, et envisager les modes d'interaction qui en découleraient. Un fil de discussion<sup>49</sup> (*thread*) sur *Reddit* énumère les pires manières possibles d'interagir avec un paramètre de volume. Bien qu'à visée humoristique, ce *thread* est fascinant par la quantité faramineuse d'idées proposées, et de possibilités d'interactions alternatives.

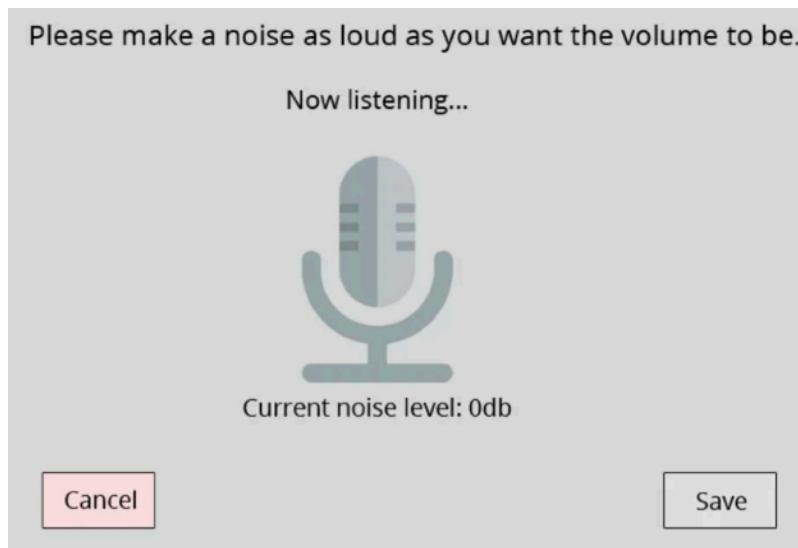


Figure 13. Un des exemples du *thread*<sup>50</sup>

Toutes ces innovations supposent un bouleversement conséquent des habitudes des utilisateur·rice·s et des concepteur·rice·s. Vinet et Delalande (1999) précisent que l'un des obstacles se trouve dans les librairies de programmation utilisées pour concevoir les interfaces : on y trouve de nombreuses fonctions pour les objets d'interaction habituels (menus, boîtes de dialogue, etc.) mais elles sont difficilement extensibles à des interfaces post-WIMP. Il serait donc nécessaire de concevoir de nouveaux environnements de développement qui nous permettraient d'explorer plus aisément de nouvelles méthodes d'interactions. C'est d'ailleurs ce qu'a entrepris de

<sup>49</sup> <https://uxdesign.cc/the-worst-volume-control-ui-in-the-world-60713dc86950> [consulté le 17.04.2024]

<sup>50</sup> "veillez faire un son aussi fort que le volume que vous désirez. En écoute..."

faire l'équipe de *Fors*<sup>51</sup> qui développent son propre *framework* lui permettant de concevoir ses interfaces comme elle l'entend, tout en bénéficiant d'une meilleure efficacité technique.

Cette constatation explique peut-être en partie la différence évoquée plus haut entre le développement *hardware* et *software*. Utiliser une interface rudimentaire pour un synthétiseur physique est un choix en soi : un synthétiseur *D/Y*<sup>52</sup> dans une brique de lait aura du caractère. Un synthétiseur logiciel *D/Y* en revanche, utilisera probablement les *widgets* par défaut de *JUCE*, le choix aura donc été fait par autrui. Faire du minimalisme en *hardware*, c'est faire moins ; faire du minimalisme en *software*, c'est devoir réécrire l'ensemble des fonctions par défaut fournies par les librairies.

Une amélioration des *presets* par leur interface semble donc compliquée à mettre en place : bien que largement améliorable, cette modification demanderait des changements profonds, techniquement et dans les habitudes des utilisateur·rice·s.

L'évolution des habitudes de travail est un processus long : le nombre d'acteur·rice·s et le millefeuille d'influences culturelles s'entrelaçant avec celui des influences techniques rendent l'inertie d'autant plus importante. Un changement nécessiterait la synchronisation de nombreuses personnes agissant à plusieurs niveaux d'abstractions, ainsi que des artiste-luthier·ère·s croisant leurs réflexions de conception et d'utilisation afin de démocratiser des paradigmes d'interaction qui sont encore aujourd'hui peu répandus. À titre d'exemple, le clavier AZERTY, à l'origine conçu en partie pour éviter le blocage de machine à écrire, persiste encore aujourd'hui malgré une efficacité sujette à débat. L'universalité de l'interface et le temps nécessaire pour apprendre une de ses variations (comme le clavier Bépo) écrase largement le potentiel bénéfice que cette variation pourrait apporter.

Cet équilibre entre universalité et temps d'apprentissage prend une autre forme pour des instruments, dont l'objectif n'est pas seulement l'efficacité. Sommes-nous verrouillés dans nos habitudes, là où des méthodes d'interactions alternatives pourraient nous ouvrir de nouveaux processus créatifs ?

---

<sup>51</sup> <https://fors.fm/> [consulté le 10.05.2024]

<sup>52</sup> Do It Yourself - fait le toi même

Pour garder les utilisateur·rice·s intéressés et dans un “état de flow”<sup>53</sup>, il faut idéalement proposer une interface excédant légèrement les compétences de l’utilisateur (Bjørn, 2017, p25), un changement d’UI provoquerait peut-être une rupture trop forte : l’outil pourrait sembler trop complexe ou expérimental, ce qui risquerait de rebuter culturellement certain·e·s utilisateur·rice·s.

Une approche plus subtile est mesurée pourrait se trouver dans le *mapping*.

### 2.4.3 Mapping

Les *macro-knobs* déjà en place dans la plupart des synthétiseurs pourraient être améliorés assez simplement : pour le moment, un *macro-knob* permet en général d'accéder à *un* paramètre du synthétiseur. L'interface n'aurait donc pas à être modifiée puisque l'élément graphique et fonctionnel est déjà présent. Le système de traitement ne nécessiterait qu'un ajout mineur pour permettre de nouveaux modes de *mapping*. Il suffirait alors de donner quelques outils supplémentaires aux sound-designeur·e·s pour qu'ils puissent augmenter les capacités des *macro-knobs* déjà existants afin de les rendre plus attrayants.

De nombreux articles explorent des *mappings* potentiels, en particulier pour augmenter des instruments acoustiques ou analyser des gestes dans l'espace, mais peu se sont penchés sur ce qui est aujourd'hui un des contextes de travail les plus répandu : le DAW et la musique "*in the box*". Il convient alors d'étudier ce que la recherche propose comme possibilités pertinentes dans notre contexte, que je pourrais ensuite adapter à nos *macro-knobs*. Nous verrons quelles possibilités proposées par la recherche pourraient être pertinentes dans notre contexte.

---

<sup>53</sup> L'état de flow est un état de mental de concentration dans l'accomplissement d'une tâche, l'utilisateur est alors complètement absorbé par son occupation.

# **Mapping, revue de littérature**

Le *mapping* est la mise en correspondances de deux ensembles de valeurs. Dans notre cas, ces ensembles sont les paramètres de contrôle d'un GUI et les variables d'un synthétiseur logiciel.

Hunt, & Wanderley, (2002) mentionnent deux catégories de mapping, les *mappings* utilisant un mécanisme génératif et ceux définis explicitement. Dans les *mappings* génératifs, on trouvera principalement l'utilisation de réseaux de neurones comme chez Lee, & Wessel, (1992).

En gardant en tête l'objectif d'un *mapping* abordable, facilement utilisable par un·e sound-designeur·e et implémentable par un·e développeur·se, je me concentrerais sur la deuxième possibilité : les *mappings* explicites.

Ils ont par ailleurs l'avantage d'être plus facilement adaptables à des effets. Les *mappings* génératifs se basent parfois sur l'analyse du son produit par le synthétiseur, ce qui peut être compliqué à transposer à des effets dont la sortie dépend fortement de l'audio en entrée.

Le lien le plus simple entre le contrôleur et le DSP est nommé *one-to-one* (Ryan, 1991), il associe *un* paramètre du contrôleur à *une* variable du moteur de synthèse. Mais le *one-to-one* suppose tout de même une fonction de mise en forme (*shaping*) pour faire correspondre les plages de valeurs, et potentiellement ajouter une courbe de réponse permettant une interaction plus pertinente.

## 3.1 Shaping

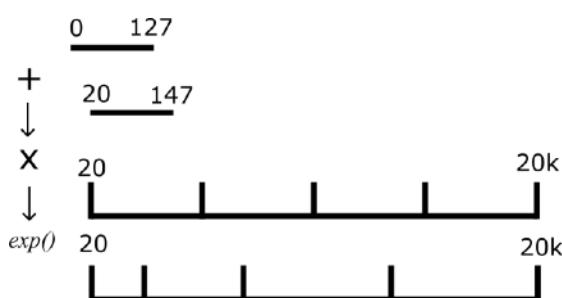


Figure 14. Évolution de la plage de valeur

Un exemple concret : si l'on cherche à utiliser le potentiomètre d'un contrôleur MIDI pour manipuler la fréquence de coupure d'un filtre, il faut faire correspondre une gamme de valeur linéaire allant de 0 à 127 (valeur MIDI standard) à une gamme de valeur logarithmique allant de 20 Hz à 20 kHz (domaine de fréquence audible).

Il faut donc ajouter un *offset* (décalage), puisque nos valeurs MIDI commencent à 0 et non à 20, puis un *scaling* (modification de l'échelle) afin que nos valeurs couvrent bien toute la plage de la variable. Ceci fait, le filtre sera peu pratique à utiliser car nous avons une perception logarithmique des fréquences. Le centre du potentiomètre donnerait une valeur de 10 kHz, nous manquerions donc fortement de précision dans la zone des graves et des mediums, qui représentent une grande quantité d'informations sonores. Il est donc nécessaire de passer notre valeur de paramètre dans une fonction exponentielle (on parle de *skew*). Ici encore, l'usage en temps réel de notre potentiomètre risquerait de poser des problèmes de clics et de ruptures dans le son, car 127 valeurs étalées sur 20000 fréquences créent des sauts de plusieurs centaines de Hertz dans les aigus. Il faut donc aussi lisser nos valeurs (*slew* en synthèse) en interpolant un certain nombre de valeurs intermédiaires lors de la transition entre deux pas.

Ryan, J. (1991) propose une liste des fonctions fondamentales applicables à un flux de données de contrôle :

- décaler ou inverser (**addition**)
- compresser ou étendre (**multiplication**)
- limiter, segmenter ou quantifier (**thresholding** - seuillage en français)
- en conservant en mémoire les états précédents du flux, on peut lisser, mesurer la vitesse de changement, amplifier certaines caractéristiques, ajouter des délais ou de l'hystérésis (**differentiel, intégration, convolution**).
- le débit de données peut être réduit ou augmenté (**décimation, interpolation**).

Une fois les bases du *shaping* posées, on peut tisser des liens plus complexes entre les flux entrants et les variables à contrôler.

## 3.2 Topologies de Mapping

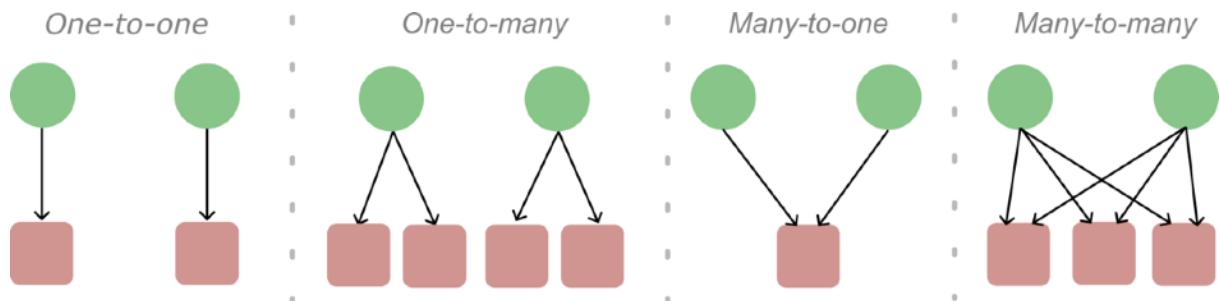


Figure 15. Topologies de mapping fondamentales

Après le *one-to-one*, on peut commencer à contrôler plusieurs variables avec un seul paramètre : il s'agit de *one-to-many* (Hunt, et al. 2002). On trouve aussi le terme *divergent* (Rovan et al. 1997). Cette méthode permet de rapidement obtenir des *mappings* plus intriqués, qui peuvent convertir un mouvement simple en un évènement sonore complexe. Elle ouvre une véritable réflexion et le *mapping* devient un outil créatif ayant un impact fort sur le résultat sonore. L'évolution simultanée de plusieurs variables pousse à faire des choix esthétiques, chaque groupe de variables permettant des modulations drastiquement différentes. Chaque *mapping* entre un paramètre et une variable peut avoir une fonction de *shaping* différente, ce qui ouvre d'autant plus de possibilités.

Le *one-to-many* nous invite à imaginer son contraire : le *many-to-one*. Il est rarement utilisé tel quel puisque de manière générale, le mapping des instruments tend à être *few-to-many*, c'est-à-dire qu'il existe plus de variables que de paramètres avec lesquels interagir (Hunt, et al. 2002). Mais dans le cadre d'un mapping *one-to-many*, faire en sorte que certains paramètres modulent la même variable peut permettre de se rapprocher du *mapping* des instruments acoustiques, comme décrit plutôt avec l'exemple du violon.

Hunt et Kirk (2000) concluent à l'issue de leurs expériences que le mapping *one-to-one* est moins engageant pour l'utilisateur·rice·s. Étonnamment, dans leur test - dont l'objectif était de reproduire un son entendu à l'aide de *sliders* - des *mappings* plus complexes permettaient une meilleure efficacité, notre perception du son ayant du mal à séparer les caractéristiques du signal et plus de facilité à travailler le son comme avec un instrument pour lequel les variables sont liées entre elles.

Un *mapping* bien pensé peut permettre une interaction temps réel plus plaisante avec l'instrument, ainsi que des modifications timbrales qui ne sont possibles que par la manipulation simultanée de plusieurs variables.

Ce type de *mapping* modifie l'affordance des modulations complexes, en les mettant sur le devant de la scène. En faisant des “propositions de modulation”, on invite l'artiste à profiter du plein potentiel de celles-ci, voir à en inventer d'autres, une fois que le fonctionnement est compris et qu'elle·il prend la pleine mesure des opportunités.

### 3.3 Exemple de mapping créatif



Figure 16. Serge Modular DUSG (gauche)<sup>54</sup> - Mutable Instrument Tides (droite)<sup>55</sup>

Afin de mieux comprendre les possibilités qu'offre le *mapping*, je propose ici l'exemple du module *Eurorack*<sup>56</sup> *Tides* de Émilie Gillet (*Gillet, s. d.*) qui est un cas pertinent d'outil plaçant le *mapping* en son centre.

*Tides* est inspiré de modules plus anciens, comme le *DUSG (DUal Slope Generator)* de *Serge Modular*. Leur concept fondamental est simple : ils génèrent une pente montante puis descendante dont on peut régler le temps de montée et le temps de

<sup>54</sup> <https://schneidersladen.de/en/serge-modular-dual-universal-slope-generator-mk2-dsg> [consulté le 16.24.2024]

<sup>55</sup> [https://pichenettes.github.io/mutable-instruments-documentation/modules/tides\\_2018/](https://pichenettes.github.io/mutable-instruments-documentation/modules/tides_2018/) [consulté le 16.24.2024]

<sup>56</sup> L'Eurorack est un format de synthèse modulaire. La synthèse modulaire donne accès à chaque bloc fonctionnel (module) individuellement et permet de les recombiner selon notre bon vouloir à l'aide de câble de patch

descente. Plutôt que de proposer ces deux variables comme paramètres, *Tides* a un potentiomètre permettant de définir le ratio entre le temps de montée et le temps de descente et un autre potentiomètre qui règle la vitesse globale de ces pentes. Ce choix propose une affordance différente et des opportunités de modulations nouvelles, qui auraient été plus compliquées à mettre en place avant et moins évidentes à imaginer.

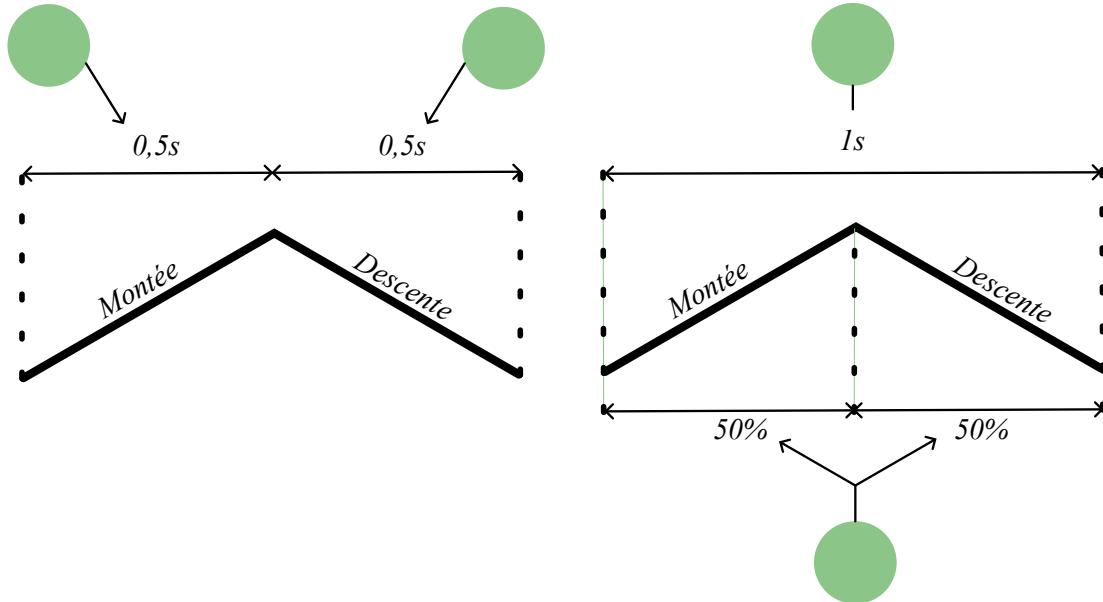


Figure 17. La même enveloppe sur le DUSG (gauche) et Tides (droite)

*Tides* génère 4 pentes, on pourrait alors penser qu'il offrirait 4 fois ces mêmes paramètres, mais là encore, Émilie Gillet prend une direction différente en utilisant un paramètre réglant plutôt *la relation* entre les pentes selon plusieurs modes.

*Tides* fait aussi un bon usage des contraintes : lorsque les pentes atteignent des fréquences hautes et deviennent audibles, le paramètre de relation limite ses valeurs à des ratios de fréquences donnant des accords. Ce choix limite les possibilités, mais facilite l'usage dans la composition musicale, c'est une décision qui influence fortement la pratique et les résultats sonores.

*Tides* est donc un bon exemple d'instrument formateur, qui permet de mieux imaginer ce qui est rendu possible par l'usage de ces techniques. Ces *mappings* ne sont pas "compliqués", mais sont bien pensés et au centre de l'outil. Ils permettent alors aux utilisateur·rice·s d'explorer des logiques de composition auxquelles elles·ils

n'auraient pas pensé avant, ou qui auraient été laborieuses à utiliser avec les outils préexistants. Le DUSG conserve malgré tout de nombreux avantages : il expose des briques de fondation d'apparences très simples, mais qui cachent une étonnante complexité. Il peut être source d'un travail plus approfondi et plus proche de la machine. Les deux modules ont chacun leur philosophie et permettent deux approches très différentes de la synthèse modulaire, qui peuvent tout à fait cohabiter dans un même *patch*.

Bien que nous ne disposions pas de la flexibilité disponible à l'élaboration d'un nouvel instrument en partant de zéro, *Tides* semble être une bonne référence dans son utilisation du *mapping one-to-many*. Il ouvre des pistes de réflexion sur ce qu'il est possible de faire avec des *macro-knobs*.

### 3.4 Couche de paramètres intermédiaires

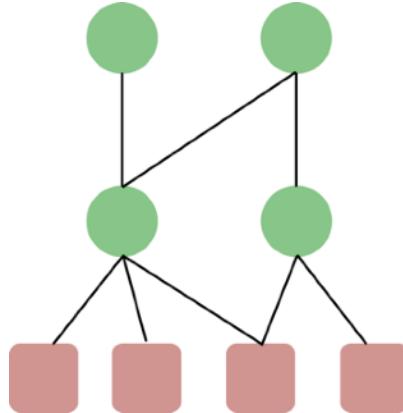


Figure 18. Utilisation de paramètres intermédiaires

Afin de faciliter l'élaboration de *mapping one-to-many* Garnett et Goudeneuse (1999) proposent l'utilisation de paramètres perceptifs. Ils permettent d'abstraire un certain nombre de variables derrière une perception sonore comme la "brillance". Wanderley *et al.* (1998) proposent eux l'utilisation de paramètres abstraits.

L'utilisation de cette couche intermédiaire est particulièrement utile dans le cas de synthèses avec beaucoup de variables comme la synthèse additive, dans laquelle la manipulation d'une variable (l'augmentation du gain d'une seule sinusoïde) a peu d'impact dans le son final.

On peut donc avoir un *mapping many-to-many* entre les paramètres de l'interface et les paramètres intermédiaires, mais aussi un mapping *many-to-many* entre les paramètres intermédiaires et les variables de synthèse.

## 3.5 Interpolation de preset

Plutôt que de choisir individuellement les paramètres et leurs évolutions, Bowler et al. (1990) proposent de choisir plusieurs pré-configurations et de faire des moyennes entre les états de chaque variable. Entre un *preset 1* et un *preset 2*, on trouverait donc un *preset 1,5* qui se trouverait à mi-chemin. Toutes les variables prenant de nouvelles valeurs, le timbre produit est plus qu'une simple moyenne des deux timbres, on peut parfois être surpris par le résultat.

On obtient une sorte de *mapping one-to-many*, un paramètre modifiant le coefficient d'interpolation qui va ensuite contrôler tous les paramètres de synthèse. On peut voir ce type de *mapping* comme un cas particulier de couche intermédiaire.

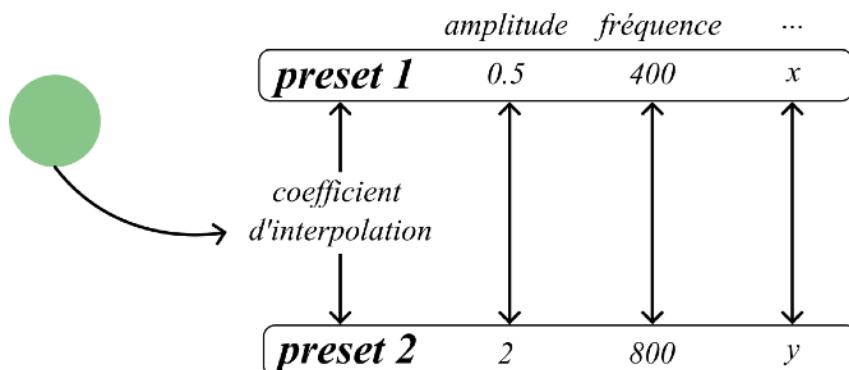


Figure 19. L'interpolation agit sur de multiples variables simultanément

Même au plus simple, avec deux *presets*, une interpolation peut ouvrir un nouvel espace d'exploration. Les sound-designer·e·s peuvent se contenter de faire leurs *presets* comme il en ont l'habitude et de choisir ensuite des paires à interpoler, ou laisser le choix à la·le utilisateur·rice de faire ses mélanges.

Néanmoins, une réflexion plus poussée des états intermédiaires et du mouvement qui sera généré, ainsi qu'une construction des *presets* pensée autour des possibilités de l'interpolation, permettra de mieux les exploiter et de concevoir des interactions plus intéressantes pour les utilisateur·rice·s.

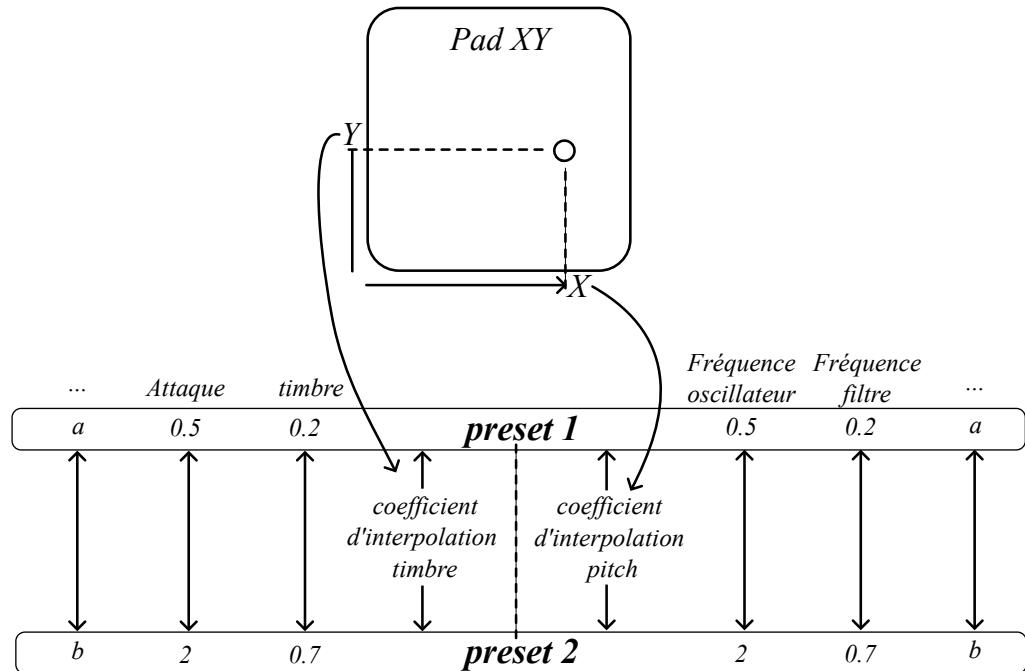


Figure 20. Fonctionnement du Polybrute de Arturia

Le *Polybrute* de *Arturia* est un synthétiseur qui utilise ce système d’interpolation de *presets*. Il le contrôle sur deux axes : le premier permet de régler les paramètres associés à la fréquence du son (note de l’oscillateur, fréquence de coupure du filtre) qui ont une importance toute particulière dans l’harmonie de la musique (de plus dans un synthétiseur polyphonique, donc destiné à faire des accords et des mélodies).

Le second axes est dédié au reste des paramètres, le tout permettant une interaction précise et nuancée avec le synthétiseur.

On peut aussi utiliser les deux axes comme les coordonnées d’un espace dans lequel on peut placer une multitude de *presets* qui seront interpolés selon plusieurs modes : les *presets* comme planètes dans le *SYTER* de l’IRCAM, comme lampes chez Spain, & Polfreman, (2001), sources de triangulation chez Drioli *et al.* (2009), ou en *nodes* dans l’outil Max/MSP conçu par Andrew Benson en 2009 (Gibson, & Polfreman, 2019).

## 3.6 Modèle physique

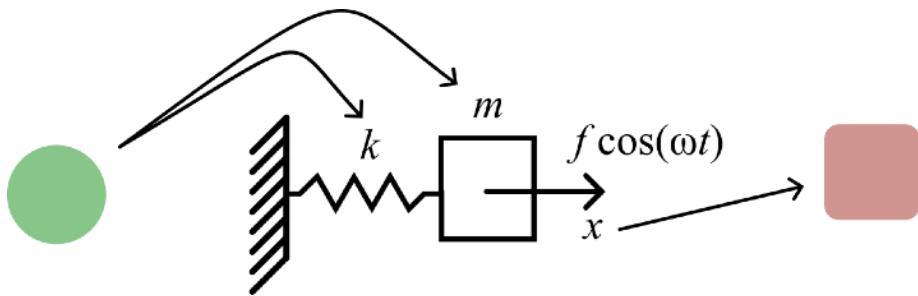


Figure 21. Modèle masse-ressort comme couche intermédiaire

Afin de concevoir des *mappings* bénéficiant d'une logique interne, qui seront donc plus facile à apprêhender puisque utilisant des modèles mentaux préexistant, on peut utiliser des modèles physiques comme couche intermédiaire. Le paramètre modifie certaines constantes de l'équation dont le statut peut ensuite être assigné à une variable.

Exemple : dans l'utilisation d'un modèle masse/ressort, le paramètre peut modifier la masse ou la raideur du ressort. De cette équation, on peut déduire la position de la masse qui est ensuite assignée à une variable.

On peut imaginer l'utilisation de tout type de modèles : murmuration d'étourneaux, modèle particulaire (Montgermont, 2005), croissance des plantes, gravitation, etc.

La suite “*inspired by nature*” de Dillon Bastan contient plusieurs exemples de ces procédés.

Momeni, A., et al. (2006) décrivent plus largement l'utilisation de ces simulations physiques pour le contrôle de synthétiseurs audio ou visuel en proposant des objets *Max* et *Pure Data*.

Ces *mappings* sont en général associés à un retour visuel approprié, sans cela il peut être compliqué de comprendre ce qu'il se passe et quelles sont les conséquences de l'utilisation de chaque paramètre. De plus, ces modèles ont des impacts visuels fort et on peut y voir une certaine beauté. Ce type de *mapping* est en général très ludique et donne envie d'expérimenter avec le système, ne serait-ce que pour voir les conséquences de nos mouvements se refléter sur l'interface.

## 3.7 Non-linéarité

*Surprise emerges from a system that is partially but not completely knowable*  
(Mudd, T. 2023)

En prenant en compte la volonté de ne pas trop bousculer l'UI, on peut partir d'interfaces ayant des *widgets* similaires mais des modes d'interaction différents.

Je pense ici au *no-input mixing* qui consiste à brancher les sorties d'une console de mixage sur ses propres entrées. Cette récursion comprenant des étages de gain et de traitement, on crée ainsi une boucle de rétroaction qui peut engendrer de l'oscillation.

On obtient un instrument dont les éléments d'interface sont similaires aux nôtres : des potentiomètres et des *sliders*, mais dont le comportement est très particulier. Todd Mudd (2023) compare les instruments basés sur le *feedback*<sup>57</sup> à un espace explorable dans lequel les trajectoires naviguent entre des états stables, et pour lequel atteindre certaines niches nécessite d'effectuer les bons mouvements au bon moment.

Les paramètres de la console n'ont plus rien à voir avec leur utilité d'origine, le gain n'est plus un simple contrôle du volume et se met à avoir une influence importante sur la fréquence du son produit (Mudd, D. 2023).

Un réseau de *mappings* complexes se met en place. En effet, le *feedback* engendre de nombreuses non-linéarités, c'est-à-dire qu'en replaçant les potentiomètres aux mêmes valeurs, on obtiendra un résultat sonore différent en fonction des états passés du système. Ces caractéristiques rendent l'instrument passionnant et permettent une exploration sans cesse renouvelée à travers le temps. Néanmoins, on ne perd pas la possibilité de diriger l'instrument avec une grande précision et on peut développer une véritable connaissance et maîtrise de ce qu'il est possible d'accomplir. Ce type d'interaction pousse les utilisateur·ice·s à se concentrer sur l'écoute et l'attention portée au comportement de la machine.

Mudd conclut que travailler avec des instruments compliqués, confus ou imprévisibles peut être enrichissant, et qu'un contrôle direct et complet n'est pas toujours désirable.

---

<sup>57</sup> Ou boucle de rétraction en français.

Sans nécessairement chercher à simuler le comportement du *no-input mixing*, tendre vers certaines des caractéristiques évoquées permettrait peut-être un engagement plus fort avec les *presets*.

# Comparaison expérimentale des *mappings*

## 4.1 Contexte d'expérimentation

Je propose de sélectionner 4 de ces *mappings* explicites présentés dans la partie précédente et les faire comparer par des compositeur·rice·s lors d'une expérience. J'ai choisi de laisser de côté le modèle physique qui pâtirait d'une interface standard, et de me concentrer sur les 4 *mappings* suivants :

- le *one-to-one*
- le *one-to-many*
- l'interpolation
- la non-linéarité

J'ai choisi le DAW *Ableton Live* car il donne accès à *Max For Live* (M4L). C'est une version de *Max/Msp*<sup>58</sup> utilisable dans *Ableton* sous forme de *devices* qui ont la même apparence que les *plug-ins* natifs. La possibilité de créer ou modifier ces *devices* me permettra de mieux contrôler les *mappings*, en particulier pour le non-linéaire.

Dans le but pouvoir généraliser les résultats, j'ai conçu 4 sessions *Ableton*. Chacune des 4 sessions utilise un synthétiseur différent, dont un *preset* est décliné en 4 instruments : un par *mapping* testé, chacun sur une piste. Les pistes de chaque session sont dans un ordre différent afin d'éviter un biais de l'effet d'ordre dans les résultats.

Afin de reproduire l'expérience d'utilisation d'un hôte de *plug-in*, je masque l'interface du synthétiseur et propose deux macros-paramètres pour chaque *mappings*. Cela permet par ailleurs de rendre les instruments en tout point semblables du point de vue de l'interface graphique. Disposer de deux potentiomètres plutôt qu'un seul permet d'explorer les interactions possibles entre deux contrôles. Deux potentiomètres plutôt que trois permet de rester concentré sur l'essence du *mapping* sans se perdre dans la complexité générée par un paramètre supplémentaire.

---

<sup>58</sup> Logiciel de programmation visuel permettant de traiter des flux de données et de l'audio en temps réel

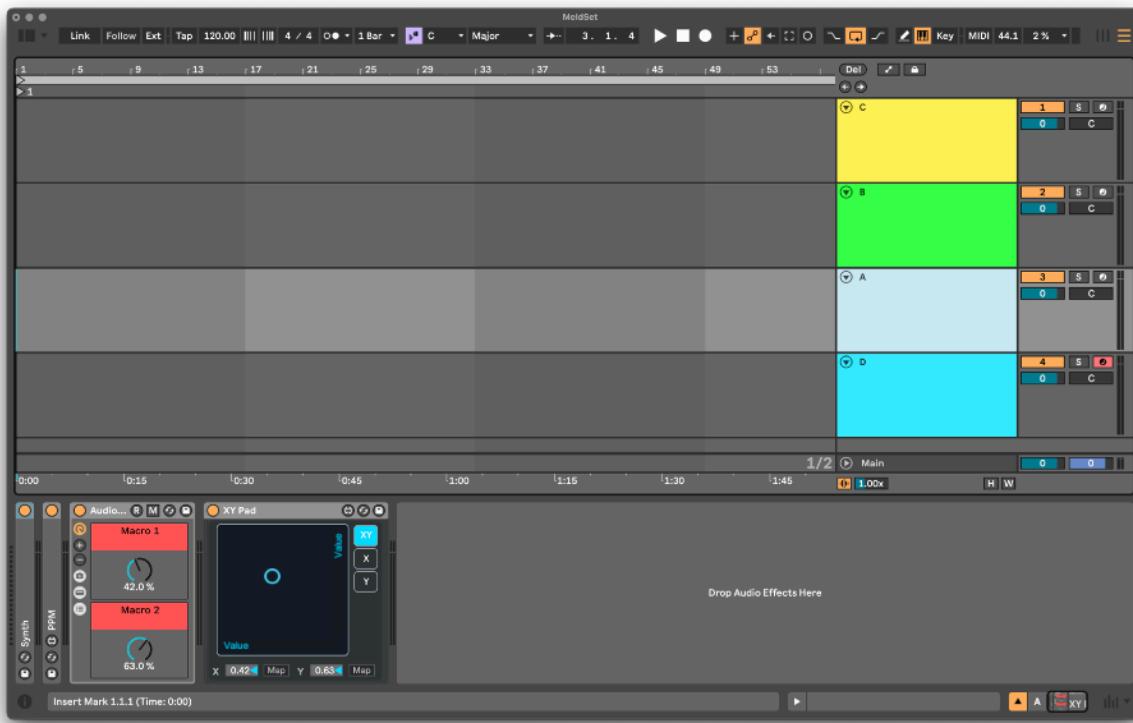


Figure 22. Une des sessions Ableton

La piste A contient le *preset* contrôlé en *one-to-one*, la piste en B en *one-to-many*, la piste C en interpolation et la piste D en non-linéaire. On voit en bas à gauche de la figure 21 les contrôles de la piste sélectionnée (*one-to-one*).

La surface de contrôle est limitée à une souris et un clavier. J'envisageais à l'origine l'utilisation d'un clavier MIDI, mais après quelques tests, il semblerait que limiter les capacités harmoniques au clavier d'ordinateur permette aux participant·e·s de mieux se concentrer sur les évolutions du timbre.

Pour pallier aux limites de la souris, qui ne peut moduler qu'un seul paramètre à la fois, j'utilise un pad XY dont un axe contrôle la valeur de la macro 1 et l'autre axe la macro 2. On obtient un élément d'interface qui peut contrôler les deux paramètres simultanément, et qui a l'avantage de donner à voir l'ensemble des combinaisons possibles des deux valeurs.

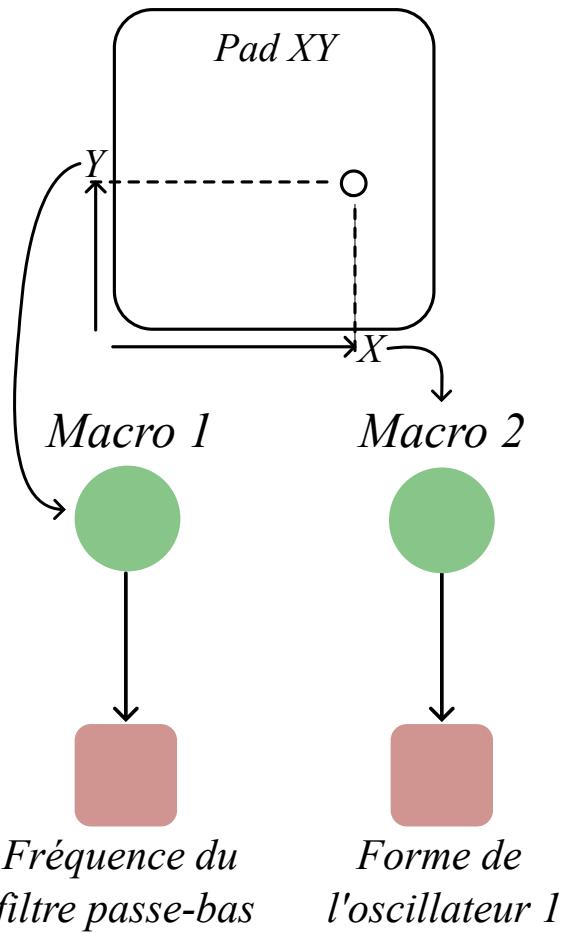


Figure 23. Un des mappings one-to-one

## 4.2 Conception des *presets*

Les 4 sessions utilisent des synthétiseurs natifs de *Ableton Live* :

- *Operator*, par modulation de fréquence
- *Wavetable*, par table d'ondes
- *Drift*, par soustraction
- *Meld*, plus moderne, dispose de plusieurs algorithmes pour les oscillateurs et pour les filtres. En l'occurrence, j'ai utilisé un double oscillateur à table d'ondes passant dans un passe-bas ajoutant du repliement spectral, sommé avec une banque de sinusoïdes passant par un *phaser*.

Le *mapping one-to-one* de *Meld* dispose d'une certaine complexité car les paramètres de ses oscillateurs sont déjà des formes de macros-paramètres.



Figure 24. Les oscillateurs de *Meld*<sup>59</sup>

Dans le cas du *one-to-one* j'ai modulé le paramètre de *spacing* qui modifie le ratio entre chaque oscillateur de la banque, les fréquences de ces oscillateurs étant verrouillées sur une gamme prédéfinie.

De même pour la session modulation de fréquence, la modification d'un seul des paramètres du synthétiseur peut suffire à engendrer des changements drastiques du timbre.

## 4.3 Prototype de *mapping*

Les *mappings* prototypés devront pouvoir être intégrés dans des logiciels existants sans invalider les systèmes déjà en place et sans nécessiter une modification de l'interface graphique ou du système de traitement. Ils doivent pouvoir être l'objet d'une mise à jour mineure, ceci pour les raisons évoquées plus haut : proposer un système abordable à la fois par les utilisateur·rice·s et par les développeur·se·s.

### 4.3.1 Mapping one-to-one et one-to-many

Pour les *mappings one-to-one* et *one-to-many*, j'utilise un *macro-knob M4L* disponible par défaut sur *Ableton* qui permet de lier un potentiomètre à huit paramètres.

---

<sup>59</sup> Capture d'écran de Ableton Live 12



Figure 25. Les deux macro-paramètres du mapping one-to-many, session wavetable<sup>60</sup>

### 4.3.2 Interpolation

Le *mapping* par interpolation est effectué par un *device Max For Live* nommé *PresetMorph* de Frabrizio Poce<sup>61</sup> qui permet de faire de l'interpolation entre 4 *presets*.

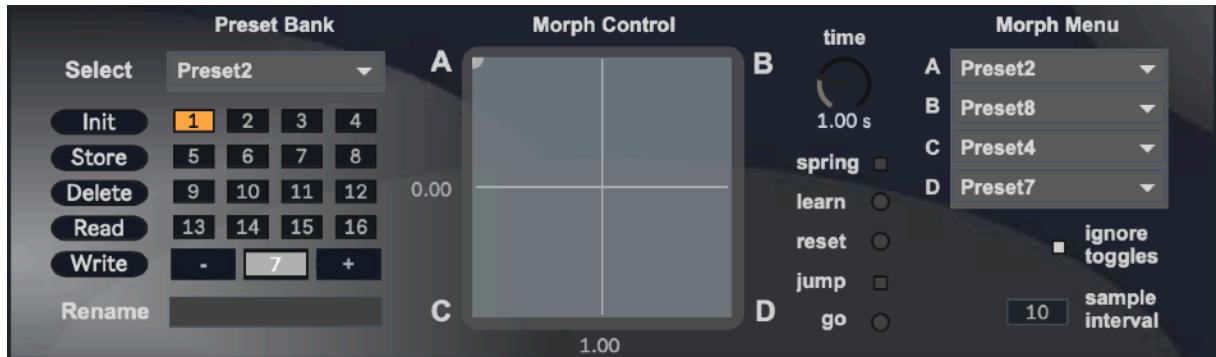


Figure 26. PresetMorph

### 4.3.3 Mapping non-linéaire

Les propriétés de la non-linéarité que je cherche à émuler sont les suivantes :

- lorsque les utilisateurs se concentrent sur une zone du pad, les réglages doivent être fins et permettre un ajustement précis du timbre.
- lorsque l'utilisateur se déplace plus largement sur le pad, l'espace doit évoluer, de façon à obtenir un son nouveau en revenant sur une zone déjà explorée.

<sup>60</sup> La non-linéarité est désactivé, voir paragraphe suivant sur le mapping non-linéaire

<sup>61</sup> <http://www.fabriziopoce.com/morph.html> [consulté le 24.05.2024]

Il existe de nombreuses manières d'avoir un espace qui évolue avec les manipulations. J'ai commencé par expérimenter avec la fonction emblématique de la non-linéarité : l'hystérésis, dont la courbe n'est pas la même à la monté et à la descente.

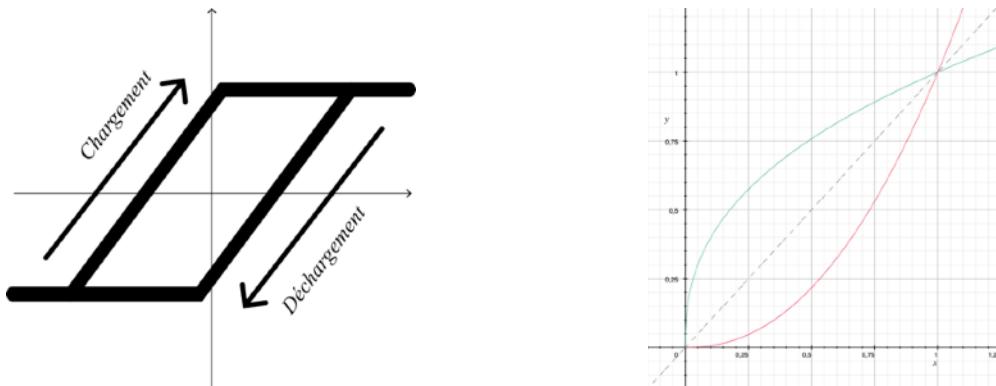


Figure 27. Hystérésis type et implémentation par l'utilisation d'un exposant variable

Cette utilisation est néanmoins compliquée à régler : pour percevoir une différence, il est nécessaire d'accentuer l'effet de déformation, mais ce faisant, on obtient une courbe qui compresse une grande partie des valeurs dans une zone minime de l'espace. J'ai donc essayé d'autres solutions permettant de conserver des plages de valeurs similaires tout en faisant évoluer l'espace. Une autre expérience peu concluante consistait par exemple à générer des vecteurs de nombres aléatoires avec lesquels était interpolée la valeur du paramètre.

Pour les tests, j'ai finalement choisi de travailler sur l'inversion des plages de valeur. Un filtre allant de 20 Hz à 20 kHz, une fois inversé, ira de 20 kHz à 20 Hz.

L'effet sur une seule variable est négligeable, mais lorsque plusieurs variables sont liées au paramètre et que chaque variable a une chance de s'inverser indépendamment, l'espace évolue plus subtilement. Par exemple, l'inversion d'un filtre peut faire apparaître une modulation dans les aigus auparavant masquée dans cette partie de l'espace.

Il existe plusieurs conditions pour qu'une plage de valeur s'inverse :

- le paramètre est à sa valeur médiane, ceci afin d'éviter des ruptures dans le timbre (on retombe ainsi sur la même valeur lors de l'inversion)

- une part d'aléatoire, pour que les variables se “désynchronisent”. Si toutes les valeurs s'inversent simultanément, on perd en subtilité dans l'évolution de l'espace
- une nouvelle tentative d'inversion ne peut intervenir qu'au bout d'un certain temps, ceci afin que l'utilisateur ait le temps d'explorer le nouvel espace

J'ai ajouté ce système au *macro-knob* par défaut vu plus haut (figure 24), afin de pouvoir l'allumer ou l'éteindre, les *mappings* non-linéaires sont donc des versions dupliquées des *mappings one-to-many*, avec le traitement non-linéaire activé.

## 4.4 Protocole de test

### 4.4.1 Questionnaire

Le questionnaire est composé de 2 sections :

- une première section dans laquelle les participant·e·s sont interrogé·e·s sur leurs habitudes de travail
- une deuxième section où il est interrogé sur son expérience lors de l'utilisation des 4 instruments proposés

Dans cette deuxième section, les participant·e·s indiquent leurs réponses en sélectionnant les instruments correspondant le mieux au critère ciblé. Exemple :

*Auriez-vous aimé avoir plus de temps avec certains instruments ? (possibilité de cocher plusieurs cases)*

- A (Rouge)
- B (Vert)
- C (Jaune)
- D (Bleu)

Une autre possibilité aurait été de proposer pour chaque question 4 échelles de Likert (une pour chaque instrument), ce qui aurait permis de récolter des données

plus précises, mais aurait rendu le questionnaire plus long et fastidieux pour les participant·e·s à cause de la redondance des questions.

#### 4.4.2 Déroulé de l'expérience

La consigne est “d'essayer chaque instrument pendant deux minutes”. Les participant·e·s n'ont pas d'objectif particulier, le but étant de reproduire l'expérience de la phase d'exploration musicale et de la recherche de matière sonore. La consigne de “tester pendant deux minutes” ne place pas les artistes dans leur processus de composition habituel : bien que certain·e·s artistes se posent la contrainte d'utiliser des sons aléatoires, ou dont elles.ils n'ont pas l'habitude, la majorité d'entre eux change de *preset* très rapidement si le précédent ne convient pas. On est ici plus proche du contexte de la découverte d'un instrument physique dans une boutique de synthétiseurs : on s'assoit devant un instrument avec lequel on prend le temps d'interagir avant de passer au suivant.

Les tests se sont déroulés dans des lieux variés, parfois au casque, parfois avec des enceintes. Ils ont tous été effectués sur le même ordinateur portable.

Les participant·e·s prennent d'abord connaissance du questionnaire et répondent à la première section sur leurs habitudes de travail.

La consigne leur est ensuite présentée (ainsi qu'une rapide explication de la méthode pour changer d'instrument pour les personnes n'étant pas familières avec le logiciel).

Les participant·e·s testent une des sessions, attribuée aléatoirement.

Une fois le test terminé, les participant·e·s remplissent la deuxième section du questionnaire. Elles.ils peuvent retourner rapidement sur la session pour mieux comparer les *mappings* en cas de doute.

## 4.5 Analyse des résultats

En tout, 26 personnes ont participé au test. Elles ont été équitablement réparties entre les 4 sessions : 6 sur la modulation de fréquence, 6 sur la table d'onde, 7 sur *Meld* et 7 sur la soustractive.

Les participant·e·s pouvant sélectionner plusieurs réponses, la somme des sélections dépasse parfois le nombre de participant·e·s.

### 4.5.1 Profil des participant·e·s

Les participant·e·s ont des centres d'intérêt variés dans le domaine de la composition, mais ont en grande majorité le DAW comme outil de travail principal.

Le synthétiseur est présent dans les compositions de la plupart des participant·e·s.

Tous n'utilisent pas l'automation dans la phase d'exploration, mais l'utilisation d'automation dans la phase d'arrangement est unanime.

Une majorité utilise régulièrement des *presets* dans leur composition, et presque tous·tes les modifient après les avoir sélectionnés.

Elles·ils ont largement apprécié les sons proposés.

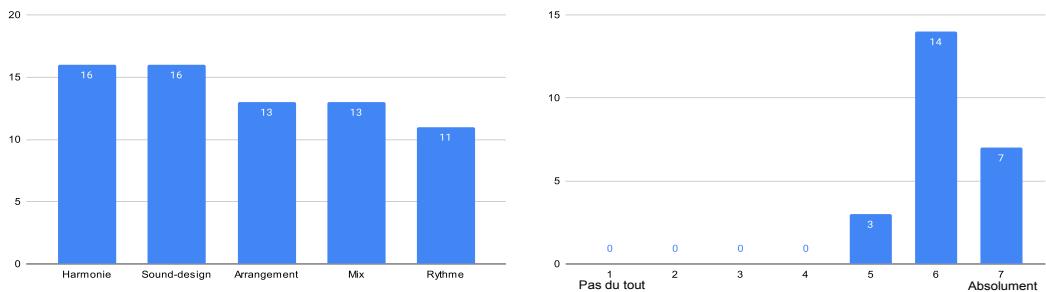


Figure 28. Domaine d'expérimentation (gauche), Appréciation des sons proposés (droite)

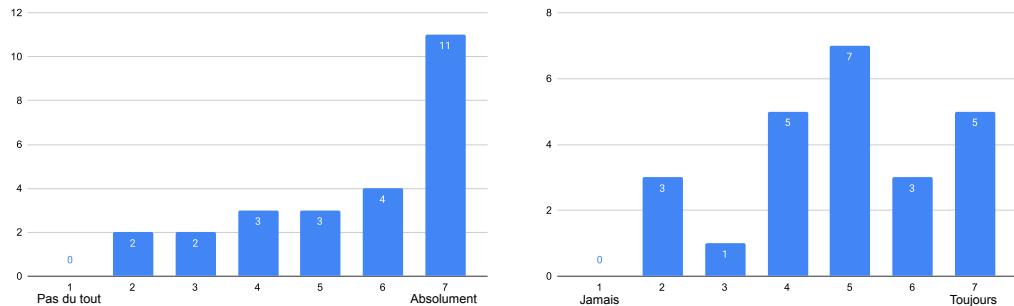


Figure 29. DAW comme outils principal (gauche), Fréquence d'utilisation des synthétiseurs (droite)

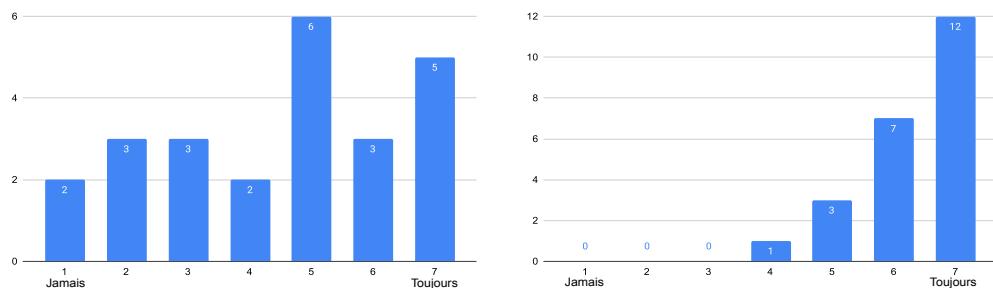


Figure 30. Fréquence d'utilisation de l'automation en phase d'exploration (gauche) et d'arrangement (droite)

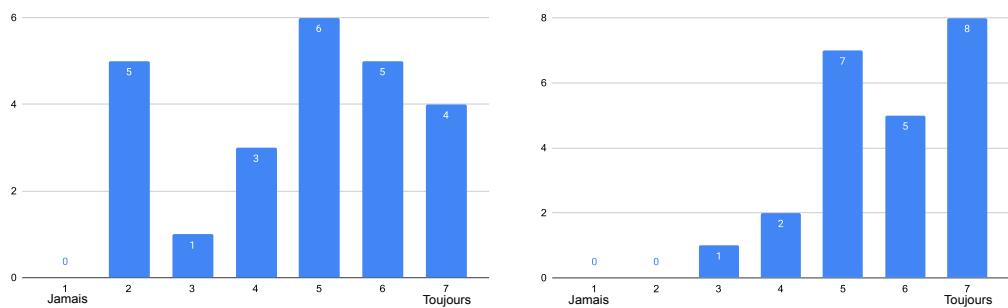


Figure 31. Fréquence d'utilisation de preset (gauche) et fréquence de leur modification (droite)

#### 4.5.2 Calcul de l'écart type entre les sessions

Afin de vérifier que les résultats ne sont pas entièrement déterminés par les sessions, j'ai comparé les résultats obtenus : pour chaque question j'ai calculé le pourcentage d'apparition de chaque *mapping*, puis l'écart type entre les pourcentages des quatre sessions, et enfin la moyenne de tous les écarts types.

Exemple :

À la question “*Auriez-vous aimé avoir plus de temps avec certains instruments ?*”

Le *mapping one-to-one* représente :

- 16,67% des sélections (soit 2 sur 11) sur la session FM,
- 6,67% (1 sur 14) sur la section soustractive
- 13,33% (2 sur 14) sur la session Meld
- 0,00% (0 sur 9) sur la session Wavetable

Avec ces 4 pourcentages, on obtient un écart type de 7,39%.

La moyenne de tous les écarts types est de 8,05%. Étant donné le nombre de résultats par sessions (en général une dizaine) 8,05% correspond à une différence d'*une* sélection entre chaque session. Cela semble assez bas pour que les résultats soient exploitables.

### 4.5.3 Usage personnel

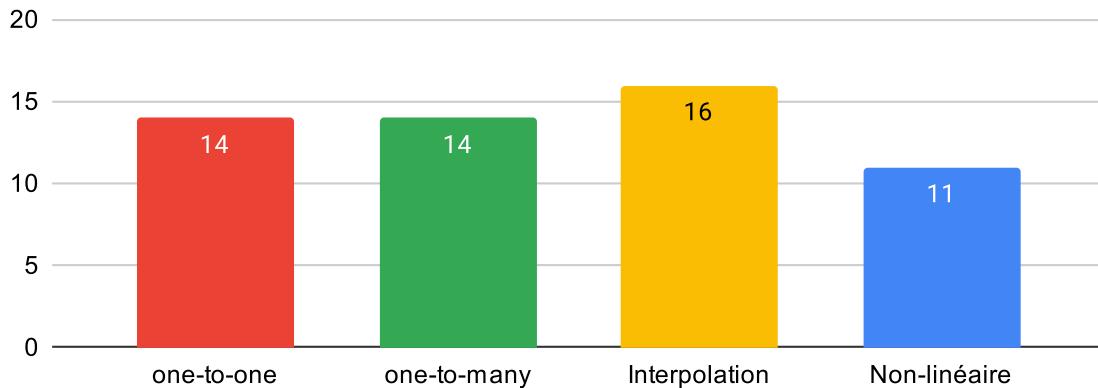


Figure 32. Quels instruments seriez-vous susceptible d'utiliser dans vos compositions musicales personnelles ?

Les participant·e·s semblent ne pas avoir de forte préférence quant à l'utilisation des instruments dans leurs compositions personnelles. Le non-linéaire est légèrement en dessous, un mouvement non-reproductible pouvant être désarçonnant dans le contexte du DAW où l'habitude est plutôt au contrôle et à la reproduction. Ceci nous apprend néanmoins qu'aucun *mapping* n'est particulièrement rebutant : ils ont tous leur place dans la panoplie d'outils des compositeur·rice·s.

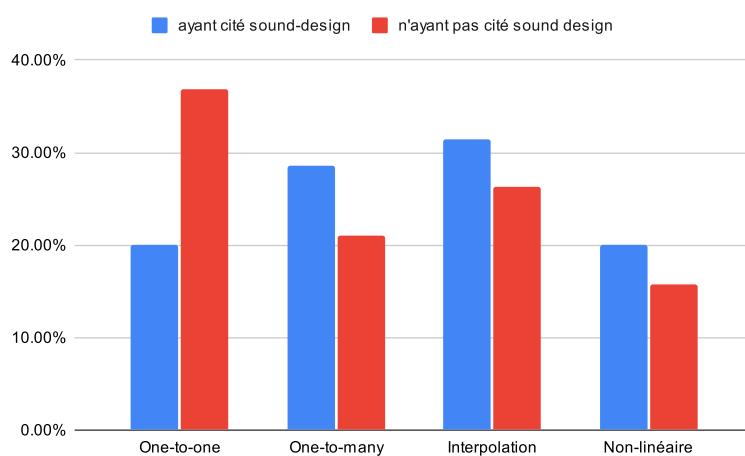


Figure 33. Utilisation dans les compositions musicales personnelles croisé avec le domaine d'expérimentation

Les personnes n'ayant pas cité le sound-design dans leurs domaines d'expérimentation ont plus tendance à citer le *mapping one-to-one*.

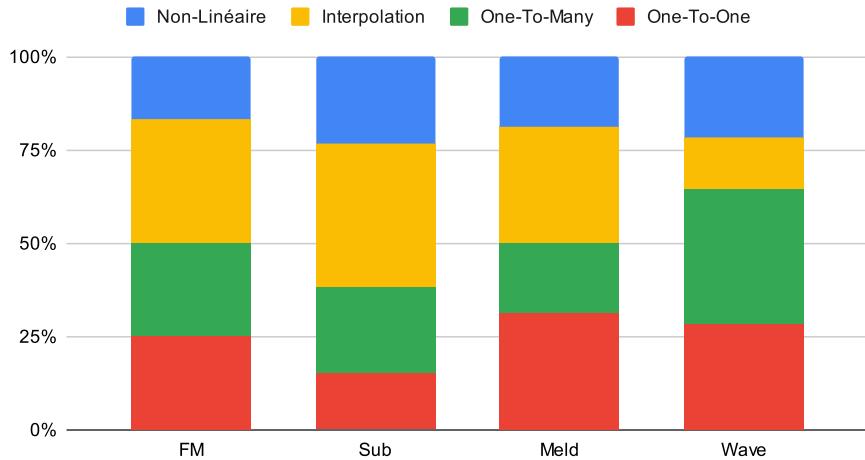


Figure 34. Utilisation dans les compositions musicales personnelles croisé avec le type de session

Deux sessions sortent du lot :

- la sessions soustractive, pour laquelle le *mapping one-to-one* est moins présent. Je suppose que cela est dû au type de synthèse, les paramètres ayant des effets assez simples sur le timbre. Le *mapping one-to-one* utilisant la fréquence de coupure du filtre et la forme de l'oscillateur, les modulations sont assez pauvres et moins intéressantes que pour les autres type de synthèse.
- la session wavetable, pour laquelle le *mapping* par interpolation est en retrait. À titre personnel, je pense que le *mapping* par interpolation est simplement moins réussi. Il propose moins de diversité et des choix assez particuliers sur les LFOs, là où le *mapping one-to-many* propose un espace timbral plus travaillé.

Plusieurs personnes ont été gênées par les *presets* de Meld. L'harmonie étant un de leurs domaines d'expérimentation, elles peuvent être gênées par un son qui contient déjà un accord et constraint donc leur réflexion harmonique.

Certain·e·s ont tout de même été très décontenancés par le fait de ne pas avoir le contrôle sur la non-linéarité. Une fois le fonctionnement expliqué après la fin de l'expérience, un.e des utilisateur·rice·s a répondu “C'est horrible !”. D'autres ont aussi soulevé des questionnements techniques : est-ce que ce type de *mapping* conservera correctement les automations et les états des paramètres, même après avoir fermé et réouvert la session. Pour pouvoir supporter ce type d'utilisation, l'implémentation de la non-linéarité devient rapidement plus complexe.

#### 4.5.4 *Mappings* inhabituels

Afin de vérifier que les participant·e·s remarquent bien que l'espace timbral change lors de l'utilisation dans le *mapping* non-linéaire, il a été demandé si certains instruments semblaient inhabituels.

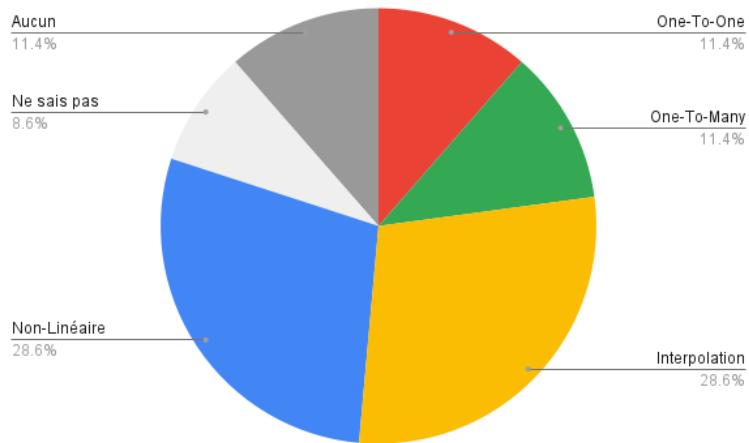


Figure 35. Certains paramètres macro vous ont-ils semblés inhabituels ?<sup>62</sup>

La non-linéarité et l'interpolation ressortent largement. L'interpolation revient souvent car elle produit des changements de timbre drastiques entre les *presets* et ajoute parfois des ruptures lors de la modification de paramètres discrets ; on a donc un plus fort sentiment d'hétérogénéité là où le *one-to-many* est en général plus fluide et homogène.

En croisant ces résultats avec la fréquence d'utilisation de synthétiseurs ou de *presets*, on s'aperçoit que la plupart des occurrences de l'interpolation viennent de participant·e·s utilisant moins de synthétiseurs et de *presets*, ceci est peut-être dû à un biais de session, la moitié de ces réponses provenant de la session soustractive. Je n'ai pas suffisamment de participant·e·s pour déterminer la significativité du biais.

---

<sup>62</sup> Pourcentage d'apparition sur le nombre totale de sélections (possibilité pour les participant de cocher plusieurs cases)

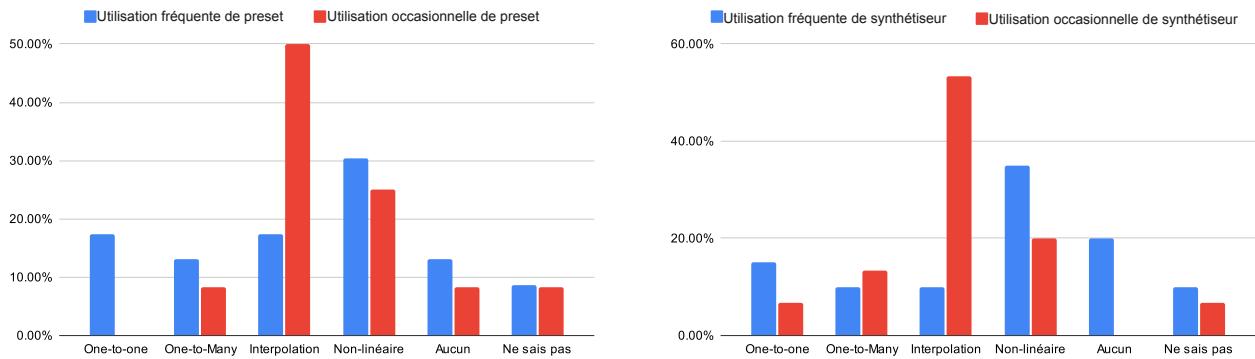


Figure 36. Caractère inhabituel des paramètres croisé avec l'utilisation de presets et de synthétiseurs<sup>63</sup>

#### 4.5.5 Espace timbral complexe

Les *mappings* complexes offrent des espaces de timbre plus complexes. En effet, les participant·e·s se lassent en général assez vite du *one-to-one* (parfois avant même les 2 minutes de jeu imposées), les deux *mappings* les plus complexes sont cités un peu plus souvent que le *one-to-many*.

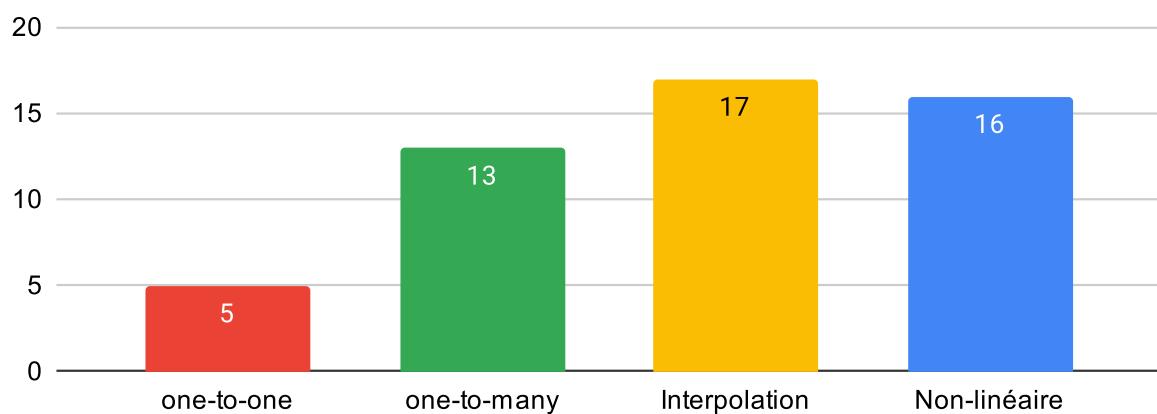


Figure 37. Auriez-vous aimé avoir plus de temps avec certains instruments ?

<sup>63</sup> L'utilisation est considéré comme fréquente si la·le participant·e à répondu 5 ou plus (sur 7) à la question correspondante

#### 4.5.6 Automation

Afin d'obtenir des informations sur l'utilisation de la modulation, j'ai préféré questionner sur le concept d'automation, qui est plus familier aux utilisateurs de DAW.

Lorsque l'on demande quel instrument les participant·e·s préféreraient automatiser, les réponses varient fortement selon les sessions. Sur la session *Meld*, par exemple, on constate qu'il y a une préférence appuyée pour le non-linéaire par rapport au *one-to-many* (5 occurrences contre 2), ce qui est étonnant compte tenu du fait que les deux ont un espace de timbre assez similaire (les paramètres choisis et leurs amplitudes de variation sont les mêmes), l'imprévisibilité pousse peut-être à vouloir plus interagir avec le paramètre lors de l'automation. Sur la session *wavetable*, le choix des *mappings one-to-one* et interpolation pourrait être expliqué par la plus grande saillance de la modulation dans ces derniers, qui est moins mise en avant dans le mapping *one-to-many* (et donc le *mapping* non-linéaire). Tout ceci tend à démontrer que dans le contexte de la modulation, le type de *mapping* compte moins que le son obtenu.

Croiser ces réponses avec celle des questions “*Utilisez régulièrement des automations pendant la phase d'exploration de la composition ?*” et “*Utilisez régulièrement des automations pendant la phase d'arrangement de la composition ?*” n'apporte pas de résultat significatif.

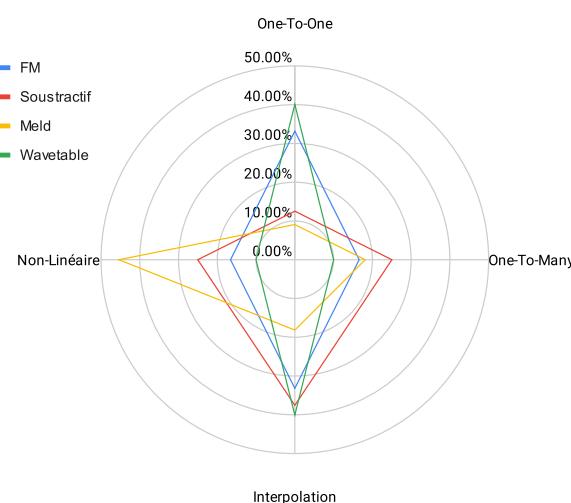


Figure 38. Quel instrument aimeriez-vous le plus automatiser ?<sup>64</sup>

<sup>64</sup> Pourcentage d'apparition sur le nombre totale de sélections pour la session (possibilité pour les participant de cocher plusieurs cases)

#### 4.5.7 Contexte *live* ou studio

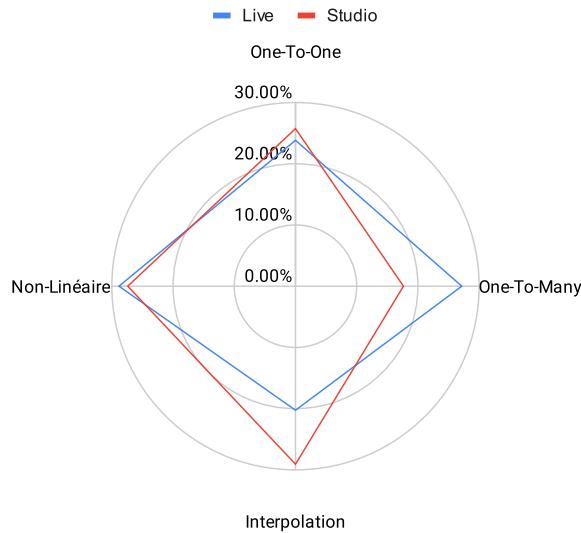


Figure 39. Choix du mapping en fonction du contexte

Les objectifs de chacun·e lors de la composition de musique en *live* ou en studio sont très variables : certain·e·s travaillent en studio comme d'autres travaillent en *live* et inversement. On constate néanmoins un plus grand nombre d'occurrences du *one-to-many* en *live* par rapport au studio, et un plus grand nombre d'occurrences de l'interpolation en studio par rapport au *live*. Les *mappings one-to-one* et non-linéaire ont un nombre d'occurrences semblables.

#### 4.5.8 Amusement

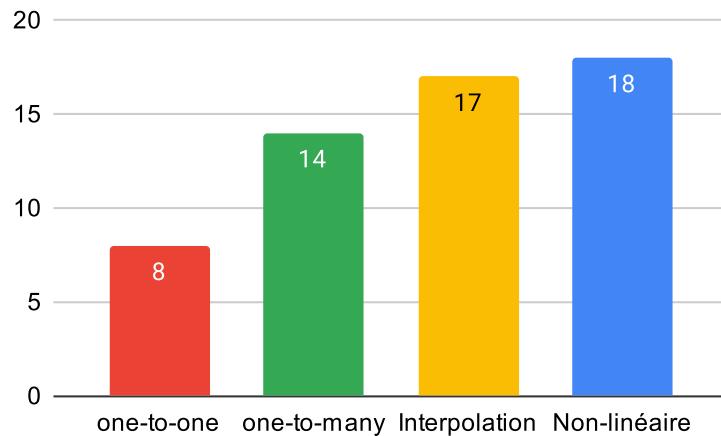


Figure 40. Quels sont les instruments qui vous ont semblé les plus amusants à utiliser ?

Une question était posée sur l'instrument que les participant·e·s trouvaient le plus amusant. Le nombre d'occurrences augmente avec la complexité, le *one-to-one* se situant largement en dessous. En croisant ce résultat avec l'utilisation de synthétiseur, on remarque que les personnes plus habituées à la synthèse prennent plus de plaisir avec un espace timbral plus complexe (l'interpolation). On retrouve le même penchant chez les personnes ayant cité le sound-design dans leurs domaines d'expérimentation.

#### 4.5.9 Questions ouvertes

À la fin du questionnaire, les participant·e·s avaient la possibilité d'ajouter une réflexion personnelle. Deux thèmes en ressortent<sup>65</sup> :

On retrouve de nombreuses fois “l'envie de comprendre” :

- “J'aurais aimé avoir plus de temps pour explorer les sons et comprendre les paramètres”
- “On a envie d'aller plus loin et de comprendre ce qu'on fait, pour pouvoir paramétrier plus facilement les automations par exemple”
- “Je me suis beaucoup préoccupé à essayer de comprendre ce que faisait chaque bouton”
- “On essaye de deviner à quoi ils correspondent en les modifiant et en écoutant”
- “J'étais à la recherche de la logique derrière : quels paramètres sont modifiés pour quelles variations du son, afin de pouvoir plus facilement les anticiper”
- “La réflexion que j'ai à la suite du questionnaire c'est qu'il y a 2 manières d'aborder l'instrument, essayer de le comprendre, ou jouer plus intuitivement selon le son produit. La deuxième option étant plus appropriée pour ce test j'ai l'impression mais je me suis trop accroché à la première pour complètement pouvoir explorer les possibilités sonores”

Certain·e·s participant·e·s acceptent néanmoins de ne pas avoir un contrôle total :

---

<sup>65</sup> Certaines citations ont été légèrement altérées pour corriger des fautes de grammaire et d'orthographe

- “*Certaines macros sont claires dans leurs fonctionnement (activation LFO, contrôle filtre ou position dans la table d'ondes) mais d'autres sont plus mystérieuses, ce qui n'est pas plus mal et est assez amusant en soi*”
- “*J'ai essayé de comprendre sur quels paramètres ça jouait et ce n'était pas forcément évident, ça donne des sonorités intrigantes et je trouve ça super intéressant de ne pas avoir forcément la main sur tout, surtout pour du live*”

#### 4.5.10 Note sur l'utilisation du pad

Le pad XY a généré certains comportements qui ne seraient probablement pas apparus avec les seuls potentiomètres.

Le pad peut facilement être utilisé pour simuler des modulations : un.e des participant·e·s simulait par exemple l'effet d'une enveloppe de modulation en effectuant un mouvement synchronisé avec son jeu. D'autres utilisateur·rice·s faisaient des mouvements répétitifs dans l'espace pour reproduire l'effet d'un LFO. Certain·e·s bougeaient erratiquement pour générer des sons rugueux et fortement modulés.

Face à un carré, les participant·e·s ont tendance à rester sur les bords et les coins (un.e des participant·e·s les a décrits comme “rassurants”). Conséquemment, certaines personnes explorent assez peu le centre du pad.

Les bords ont l'avantage de ne modifier qu'un seul des deux macro-paramètres, ce qui permet d'essayer de les comprendre individuellement avant de les manipuler simultanément.

Les *mappings* manipulaient les paramètres d'enveloppe, mais les participant·e·s ont souvent tendance à maintenir une note appuyée pour explorer les timbres. Sans redéclencher la note, les utilisateur·rice·s passent à côté de nombreuses subtilités, en particulier pour les enveloppes de modulation. La modulation de fréquence est un bon exemple de ce type de variation, des changements minimes sur le temps d'attaque de l'enveloppe d'un oscillateur peuvent avoir un impact drastique sur l'évolution du timbre au début de la note.

Le pad permet deux approches principales :

- essayer rapidement plusieurs états stables
- essayer rapidement plusieurs types de modulation

Même pour des *presets* qui ne supportent pas bien la modulation, comme la modulation de fréquence qui devient vite atonale, un pad conserve un intérêt pour itérer dans un espace de possibilités.

Le pad permet une interaction très intuitive avec les instruments.

Un.e des participant·e s'est éloigné.e peu à peu des consignes de départ, d'abord en utilisant le *piano roll*<sup>66</sup>, puis en commençant à jouer avec chaque instrument comme une piste supplémentaire au lieu de les tester successivement. Dans les faits, son utilisation se rapprochait beaucoup de l'utilisation habituelle qu'il avait de son DAW. Ses réponses ont été mises de côté, puisque l'expérience n'avait rien à voir avec les consignes d'origine, néanmoins il était intéressant de voir les instruments dans leur utilisation normale. La·le utilisateur·rice a rapidement pu ébaucher une musique, en ajoutant facilement des automations à l'aide du pad. Ce type de mapping et d'interface a créé très peu de friction pour cet.te utilisateur·rice habitué.e au DAW.

Par ailleurs, le pad XY s'est révélé très pratique dans la conception des *presets*. L'espace à deux dimensions contenant l'ensemble des interactions possibles entre les deux macros-paramètres, on peut mieux envisager l'ensemble des possibilités et construire une proposition plus complète, intéressante en tout point de cet espace.

#### 4.5.11 Influence des *mappings* sur la conception des *presets*

Les *mappings*, outre les différences d'interaction pour les utilisateur·rice·s, modifient aussi les processus de travail des sound-designeur·e·s. Une généralisation nécessiterait une seconde expérience qui se concentrerait sur ce sujet. Néanmoins je tiens à partager quelques remarques sur l'influence que ces *mappings* ont eu sur mon travail de *sound design*, en particulier avec un nombre limité de potentiomètres et l'utilisation d'un pad XY.

---

<sup>66</sup> Outil permettant de disposer des évènements MIDI dans le temps.

Un *mapping* complexe permet d'avoir une meilleure amplitude dans la complexité des mouvements du timbre.

Un *mapping one-to-one* restera souvent très simple ou très complexe puisqu'on ne peut choisir que 2 paramètres à modifier. On préférera en général des paramètres ayant un impact fort sur le timbre, comme la fréquence de coupure du filtre ou la forme d'onde de l'oscillateur. Un *mapping one-to-many* permettra d'aller plus en détail et de contrôler notamment les sources de modulations. On peut donc aller d'un son avec peu de mouvement à des modulations très complexes, somme de plusieurs enveloppes et LFO, tout en modifiant les paramètres du timbre. Le *mapping one-to-many* m'a poussé à plus développer l'évolution de ces modulations.

Le *mapping one-to-many* semble permettre une meilleure "intentionnalité" par rapport à l'interpolation qui se limite parfois à concevoir les extremums et à tester ensuite le résultat obtenu dans l'entre-deux. Par la manipulation des paramètres un par un, le *one-to-many* permet de choisir précisément l'évolution du son, et ce, tout particulièrement en utilisant des *mappings* absous au lieu de relatifs comme je l'ai fait.

L'interpolation, à l'inverse, permet de cibler très précisément les résultats obtenus dans les coins, ce qui est un peu moins vrai avec le *one-to-many*.

À partir d'un certain nombre de paramètres, le *one-to-many* devient lui aussi plus empirique, il est plus ardu de prévoir l'interaction du nouveau paramètre avec ceux déjà en place.

## 4.6 D'autres espaces à explorer

Suite à cette expérience, une autre propriété du *no-input mixing* attirent mon attention : lors de l'élaboration des *presets*, certains comportements inattendus entraînaient de fortes ruptures lors du déplacement sur le pad. Ces zones particulières permettent des "edge-like interactions" (Mudd, T et al. 2019). Lors de la modulation d'un paramètre, on recherche en général un changement fluide du timbre. La rupture - qui peut par exemple être amenée par la modulation de paramètres discrets - offre de nouvelles possibilités d'interaction. Elle permet notamment d'utiliser la modulation pour marquer un rythme, elle n'est alors plus perçue comme l'évolution d'un évènement sonore déjà présent, mais comme un évènement sonore à part entière.

Ce type d'espace complexe, contenant des lieux de rupture et de lieux de continuité, pourrait s'obtenir assez aisément en manipulant l'abscisse et l'ordonnée du pad. Par exemple, en utilisant une fonction modulo comme  $f(x) = (x+0.5) \% 1$ , on obtiendrait un paramètre continu de 0.5 à 1 puis de 0 à 0.5, avec une forte rupture en son centre, l'ensemble de l'espace serait toujours accessible, seulement réarrangé.

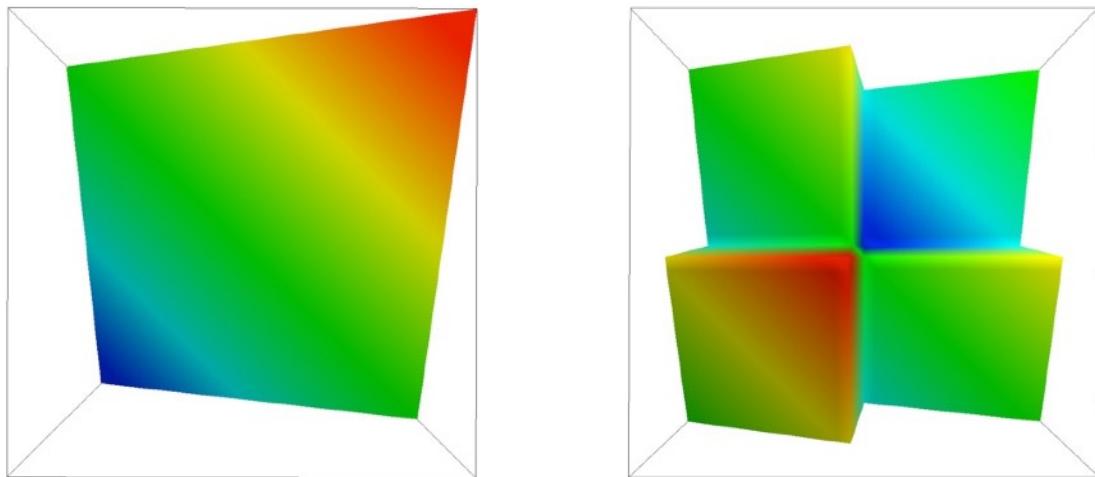


Figure 41. Pad normal (gauche), pad contenant des ruptures (droite)

La troisième dimension (accentuée par la couleur) représente la somme de  $x$  et  $y$ , l'image de gauche nous montre un pad normal, l'espace de droite un pad dont l'abscisse et l'ordonnée sont passé à travers la fonction modulo évoqué plus haut. On voit apparaître des zones fluides entrecoupées de ruptures. On reste néanmoins sur des zones aux sonorités assez similaires.

Pour aller plus loin, on pourrait contrôler des paramètres discrets de façon à produire des zones drastiquement différentes des deux côtés de la rupture. On peut par exemple activer une modulation seulement dans certaines parties de l'espace.

De nombreux tests seraient à faire sur les réactions des utilisateurs face à ce type de comportement : y'a-t-il un niveau de complexité idéal ? Peut-on se réapproprier facilement l'espace conçu par un autre ? Devrions-nous toujours laisser un chemin permettant d'accéder sans rupture à un point ?

Ce type d'interaction serait fortement facilité par l'ajout d'une interface adaptée, symbolisant la topologie de l'espace comme dans les graphiques ci-dessus. Ainsi,

l'utilisateur serait moins surpris par les accros, naviguerait plus facilement et pourrait jouer avec les ruptures en sachant où elles se trouvent.

Bien que le *mapping* puisse assurer un premier pas dans la complexification de l'interaction avec les *presets*, il me semble compliqué d'aller plus loin sans intervenir sur l'interface dont le design pourrait offrir une complexité plus maîtrisable. Les participants sont rebutés non pas par l'aspect complexe de la non-linéarité testée, mais par le fait qu'elle est purement aléatoire : on ne peut pas apprendre à perfectionner son interaction avec elle. Les utilisateur·rice·s peuvent être stimulé·e·s par une part d'incompréhension, du moment que le système dispose d'une logique interne et de comportements stables. Bien qu'intriguant pour certain·e·s, un instrument qui agit de son propre chef est plus difficile à accepter.

À l'origine, ce mémoire devait être une tentative de construire des espaces symbolisant l'ensemble des possibilités sonores d'un instrument. Je m'aperçois ici qu'il est plus simple d'inverser le raisonnement et que ce soit la construction de l'espace qui détermine les possibles. Bien que plus limité, cet espace sera plus simple à naviguer et permettra de mieux explorer les interactions alternatives que la recherche d'universalité tend à faire disparaître.

# Conclusion

Une complexification du *mapping* tel que nous venons de le voir permettrait d'ouvrir les possibilités d'interaction avec les *presets*.

Les retours des participant·e·s sur l'expérience de comparaison des *mappings* nous apprennent que des macro-paramètres plus complexes peuvent venir compléter ceux déjà présents. Chacun·e pourrait faire choisir ce qu'il préfère en termes de contrôle et de précision. Ceux désirant se faire surprendre ou cherchant une proposition de modulation peuvent interagir avec les macro-paramètres complexes des sound-designer·e·s. Ceux préférant un accès direct aux paramètres fondamentaux conservent leurs accès. Idéalement, les utilisateur·rice·s pourraient mélanger les deux types d'interactions en fonction de leurs l'objectifs.

Chaque type de *mapping* apporte un espace différent et une méthode de travail alternative. Au même titre que les sound-designer·e·s choisissent un algorithme d'oscillateur plutôt qu'un autre, elles·ils pourraient choisir un type de *mapping* plus ou moins exploratoire, adapté au type d'interaction recherché.

L'étage de *mapping* est un outil permettant de construire des micro-instruments. Il peut constituer un premier pas vers la construction d'instruments personnels. L'aspect amusant et ludique de ces *mappings* complexes peut être une bonne manière de faciliter l'approche de ces outils par les néophytes. Il facilite une première découverte du sound-design, éveillant la curiosité et le désir de compréhension presque toujours présent chez les compositeur·rice·s ayant participé à l'étude. Parmi eux, combien de futurs artiste-luthier·e·s potentiel·le·s ? Parfois, l'abstraction des paramètres n'est pas nécessaire car l'utilisateur a une forte expertise. Dans ce cas, l'utilisation de *mappings* complexes peut être une étape de préparation du jeu, faisant émerger des voies de création par de nouvelles formes d'interactions. Il est courant pour les artistes de créer leurs propres sons pendant une phase de sound-design. Lors de ce processus, insérer une réflexion sur l'interaction permettrait de construire des instruments plus personnels encore.

Le prisme du *mapping* offre aussi un regard différent sur l'évolution du timbre : les sons et les modulations obtenues sont complexes, mais l'espace timbral réduit pousse à jouer des contraintes qu'il impose.

Afin de nous aider à mieux penser le développement des futurs logiciels de musique, nous pourrions envisager d'approfondir les méthodes de construction des instruments et préciser nos objectifs lors de la conception. La multitude de problématiques entrelacées dans la création musicale nécessiterait une forte interdisciplinarité et le croisement des expertises. Dans ce cadre, une réflexion sur la place de l'ordinateur et des *GUIs* permettrait de mieux cerner les possibilités de cette interface si particulière, et de faire évoluer les logiciels vers des usages plus ambigüs, à la frontière entre l'outil et l'instrument.

# Bibliographie

- Analog Lab V User Manual (2021) Arturia [https://dl.arturia.net/products/analoglab-v/manual/AnalogLab-V\\_Manual\\_5\\_3\\_EN.pdf](https://dl.arturia.net/products/analoglab-v/manual/AnalogLab-V_Manual_5_3_EN.pdf)
- Arp 2600 patch book <https://www.korg.com/us/support/download/manual/0/842/4471/>
- Arp instrument Inc. (1976). ARP Odyssey Owner's Manual [http://arpodyssey.com/Arp\\_Odyssey\\_Manual.pdf](http://arpodyssey.com/Arp_Odyssey_Manual.pdf)
- Bowler, I., Purvis, A., Manning, P.D., & Bailey, N.J. (1990). On Mapping N Articulation Onto M Synthesiser-control Parameters. International Conference on Mathematics and Computing. [https://www.researchgate.net/publication/243775690\\_On\\_Mapping\\_N\\_Articulation\\_onto\\_M\\_Synthesiser-Control\\_Parameters](https://www.researchgate.net/publication/243775690_On_Mapping_N_Articulation_onto_M_Synthesiser-Control_Parameters)
- Buxton, W. (1986) There's More to Interaction than Meets the Eye: Some Issues in Manual Input. In Norman, D. A. and Draper, S. W. (Eds.), (1986), User Centered System Design: New Perspectives on Human-Computer Interaction. Lawrence Erlbaum Associates, Hillsdale, New Jersey, pp. 319-337.
- Carlos M., Stewart T. (1985). Fairlight CMI Operation manual <https://archive.org/details/JL10273>
- Chowning J. M. (1977) The Synthesis of Complex Audio Spectra by Means of Frequency Modulation <https://www.jstor.org/stable/23320142>
- Frauenberger, C. (2019). Entanglement HCI The Next Wave? ACM Transactions on Computer-Human Interaction (TOCHI), 27. <https://frauenberger.name/research/publications/EntanglementHCI.pdf>
- Garnet, G. E., & Goudeseune, C. (1999) Performance Factors in Control of High-Dimensional Spaces, ICM Proceedings <https://quod.lib.umich.edu/i/icmc/bbp2372.1999.393/--performance-factors-in-control-of-high-dimensional-space>

- Gaver, W. (1991). Technology Affordances. Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. [https://www.researchgate.net/publication/221518931\\_Technology\\_Affordances](https://www.researchgate.net/publication/221518931_Technology_Affordances)
- Gibson, D., & Polfreman, R. (2019). A framework for the development and evaluation of graphical interpolation for synthesizer parameter mappings. <http://eprints.bournemouth.ac.uk/32726/1/A%20Framework%20for%20the%20Development%20and%20Evaluation%20of%20Graphical%20Interpolation%20for%20Synthesizer%20Parameter%20Mappings%20FINAL%20ver5%20%281%29.pdf>
- Gibson, D., & Polfreman, R. (2020). Analyzing journeys in sound: usability of graphical interpolators for sound design. Personal and Ubiquitous Computing, 1-14. <https://link.springer.com/content/pdf/10.1007/s00779-020-01398-z.pdf>
- Gibson, D., & Polfreman, R. (2020). Star Interpolator - A Novel Visualization Paradigm for Graphical Interpolators. New Interfaces for Musical Expression. [https://zenodo.org/records/4813168/files/nime2020\\_paper10.pdf](https://zenodo.org/records/4813168/files/nime2020_paper10.pdf)
- Goldman, R. F. (1961). REVIEWS OF RECORDS. The Musical Quarterly, XLVII(1), 133-134. <https://doi.org/10.1093/mq/xlvii>.
- Harman, G. (2015). Object-Oriented Ontology. In: Hauskeller, M., Philbeck, T.D., Carbonell, C.D. (eds) The Palgrave Handbook of Posthumanism in Film and Television. Palgrave Macmillan, London. [https://doi.org/10.1057/9781137430328\\_40](https://doi.org/10.1057/9781137430328_40)
- Hunt, A., & Kirk, R. (2000). Mapping Strategies for Musical Performance. Trends in Gestural Control of Music. [https://www.researchgate.net/publication/243774325\\_Mapping\\_Strategies\\_for\\_Musical\\_Performance](https://www.researchgate.net/publication/243774325_Mapping_Strategies_for_Musical_Performance)
- Hunt, A., & Wanderley, M. M. (2002). Mapping performer parameters to synthesis engines. Organised Sound, 7(2), 97–108. doi:10.1017/S1355771802002030
- Interaction Design Foundation - IxDF. (2016, June 4). What is Skeuomorphism?. <https://www.interaction-design.org/literature/topics/skeuomorphism>

- Jézégou, A. (2022). Agentivité. Dans : Anne Jorro éd., *Dictionnaire des concepts de la professionnalisation* (pp. 41-44). Louvain-la-Neuve: De Boeck Supérieur. <https://doi.org/10.3917/dbu.jorro.2022.01.0041>
- Kirkegaard, M., Bredholt, M.,& Wanderley, M.(2020). An Intermediate Mapping Layer for Interactive Sequencing. doi : 10.1007/978-3-030-50017-7\_33
- Komplete Kontrol User Manual (2023) Native Instrument [https://www.native-instruments.com/fileadmin/ni\\_media/downloads/manuals/komplete\\_kontrol/Komplete\\_Kontrol\\_MK3\\_Manual\\_English\\_17102023.pdf](https://www.native-instruments.com/fileadmin/ni_media/downloads/manuals/komplete_kontrol/Komplete_Kontrol_MK3_Manual_English_17102023.pdf)
- Larkin, O. (2007). Int.Lib - a Graphical Preset interpolator for Max MSP. International Conference on Mathematics and Computing. <https://quod.lib.umich.edu/i/icmc/bbp2372.2007.057?rgn=main;view=fulltext>
- Lee, M.A., & Wessel, D. (1992). Connectionist Models for Real-Time Control of Synthesis and Compositional Algorithms. International Conference on Mathematics and Computing. [https://cnmat.berkeley.edu/sites/default/files/attachments/1992\\_Connectionist-Models-for-Real-Time-Control-of-Synthesis.pdf](https://cnmat.berkeley.edu/sites/default/files/attachments/1992_Connectionist-Models-for-Real-Time-Control-of-Synthesis.pdf)
- Levitin D. J., McAdams S., Adams R. L. (2002). Control parameters for musical instruments: a foundation for new mappings of gesture to sound. Organised Sound, 7, pp. 171-189 doi:10.1017/S135577180200208X
- McGregor, J. (2019) Knobs and Nodes: A Study of UI Design in Audio Plugins [Mémoire de Master, Massey University, Wellington, New Zealand]. <https://mro.massey.ac.nz/items/f630372b-5871-4864-b4ab-aefc49e4e1fc>
- Momeni, A., & Henry, C. (2006). Dynamic Independent Mapping Layers for Concurrent Control of Audio and Video Synthesis. *Computer Music Journal*, 30(1), 49-66. <https://doi.org/10.1162/014892606776021272>
- Montgermont, N. (2005). Modèles physiques particulaires en environnement temps-réel : application au contrôle des paramètres de synthèse. <http://www.atiam.ircam.fr/Archives/Stages0405/Montgermont.pdf>

- Mudd, T. (2023.) Playing with feedback: Unpredictability, immediacy, and entangled agency in the no-input mixing desk. CHI '23: Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems., 243, ACM, pp. 1-11. <https://doi.org/10.1145/3544548.3580662>
- Mudd, T., Holland, S., & Mulholland, P. (2019). The Role of Nonlinear Dynamics in Musicians' Interactions with Digital and Acoustic Musical Instruments. Computer Music Journal, 43, pp. 25-40. <https://www.semanticscholar.org/paper/The-Role-of-Nonlinear-Dynamics-in-Musicians'-with-Mudd-Holland/922fdab66eefaf1fdecbebe665c42a1dbd3795a4>
- Nirchio, L. (2018) Les aléas du design ou l'aléatoire dans le processus de création [Mémoire de Master, Université Paris 1]. <https://dumas.ccsd.cnrs.fr/dumas-01877194/document>
- Norman, D. A. (1988). The psychology of everyday things. Basic Books, New York
- Roberts, C., & Wakefield, G. (2016). Live Coding the Digital Audio Workstation. [https://www.nime.org/proceedings/2015/nime2015\\_310.pdf](https://www.nime.org/proceedings/2015/nime2015_310.pdf)
- Rodger, M., Stapleton, P., Van Walstijn, M., Ortiz, M., & Pardue, L. (2020). What Makes a Good Musical Instrument ? A Matter of Processes, Ecologies and Specificities. NIME, pp. 484-490. [https://pureadmin.qub.ac.uk/ws/files/215811524/nime2020\\_paper79.pdf](https://pureadmin.qub.ac.uk/ws/files/215811524/nime2020_paper79.pdf)
- Rovan, J., Wanderley, M., & Dubnov, S. (1997). Instrumental Gestural Mapping Strategies as Expressivity Determinants in Computer Music Performance. [https://www.researchgate.net/publication/2765549\\_Instrumental\\_Gestural\\_Mapping\\_Strategies\\_as\\_Expressivity\\_Determinants\\_in\\_Computer\\_Music\\_Performance](https://www.researchgate.net/publication/2765549_Instrumental_Gestural_Mapping_Strategies_as_Expressivity_Determinants_in_Computer_Music_Performance)
- Ryan, J. (1991). Some remarks on musical instrument design at STEIM. Contemporary Music Review, 6, pp. 3-17. [https://www.researchgate.net/publication/243784414\\_Some\\_remarks\\_on\\_musical\\_instrument\\_design\\_at\\_STEIM](https://www.researchgate.net/publication/243784414_Some_remarks_on_musical_instrument_design_at_STEIM)
- Serge – Modulisme. (s. d.). Modular Station. Consulté le 19 Mars 2023 sur <https://modular-station.com/modulisme/itatiom/serge/>

- Smith, J. (2021). The Functions of Continuous Processes in Contemporary Electronic Dance Music. [https://www.researchgate.net/publication/352569487\\_The\\_Functions\\_of\\_Continuous\\_Processes\\_in\\_Contemporary\\_Electronic\\_Dance\\_Music/citation/download](https://www.researchgate.net/publication/352569487_The_Functions_of_Continuous_Processes_in_Contemporary_Electronic_Dance_Music/citation/download)
- Spain, M., & Polfreman, R. (2001). Interpolator: a two-dimensional graphical interpolation system for the simultaneous control of digital signal processing parameters. *Organised Sound*, 6, pp. 147 - 151. <https://www.dmu.ac.uk/documents/technology-documents/research/mtirc/nowalls/mww-spain.pdf>
- Sramek, Z., Sato, A., Zhou, Z., Hosio, S., & Yatani, K. (2023). SoundTraveller: Exploring Abstraction and Entanglement in Timbre Creation Interfaces for Synthesizers. pp. 95-114. 10.1145/3563657.3596089. <https://iis-lab.org/wp-content/uploads/2023/05/DIS2023.pdf>
- Tagi, E. (2023, 11 juillet). An Interview with Peter Blasser. Perfect Circuit. consulté le <https://www.perfectcircuit.com/signal/peter-blasser-interview>
- Vinet, H.,& Delalande, F. (1999) Interface homme-machine et création musicale. Hermes Science Publications.
- Wanderley, M. M., Schnell, N., and Rovan, J. B. (1998). Escher – modeling and performing composed instruments in real-time. Proc. of the 1998 IEEE Int. Conf. on Systems, Man and Cybernetics (SMC'98), pp. 1,080–4.
- Wessel, D. L. (1979). Timbre Space as a Musical Control Structure. *Computer Music Journal*, 3(2), pp. 45–52. <https://doi.org/10.2307/3680283>
- Williams, A. (2015) Technostalgia and the cry of the lonely recordist, Journal on the art of record production, no. 9. <https://www.arpjournal.com/asarpwp/technostalgia-and-the-cry-of-the-lonely-recordist/>

## **Recommendation de lecture**

Quelque articles “introduction” qui m’ont aidé :

- *Mapping performer parameters to synthesis engines* (Hunt, & Wanderley, 2002) qui donne un bon tour d'horizon de l'historique du mapping, mais qui commence à dater un peu.
- *SoundTraveller: Exploring Abstraction and Entanglement in Timbre Creation Interfaces for Synthesizers*. (Sramek et al. 2023). Pour une rapide introduction sur l'agentivité dans le contexte musical et la place des presets.
- *Entanglement HCI The Next Wave?* (Frauenberger, C. 2019). Pour aller plus loin sur la notion agentivité, propose un tour historique de l'ensemble des théories.
- *The Role of Nonlinear Dynamics in Musicians' Interactions with Digital and Acoustic Musical Instruments* (Mudd, T., et al. 2019). Pour la réflexion sur la non-linéarité.
- A framework for the development and evaluation of graphical interpolation for synthesizer parameter mappings (*Gibson, D., & Polfreman, R. 2019*). Pour l'historique des interpolateurs graphique.

# Table des Figures

Figure 1. Plug-ins de marques variées	14
Figure 2. Architecture d'un plug-in	17
Figure 3. Roland JUPITER-8 (gauche), widgets par défaut de JUCE (droite)	20
Figure 4. De gauche à droite :	22
Neon - Steinberg (1er instrument vst) 1999 - Skeuorphism	22
VintageVerb - Valhalla DSP - 2012 - Flat Design	22
Transit - Baby Audio - 2023 - neumorphism	22
Figure 5. Un "Paperface" restauré	24
Figure 6. Un des Filtres de Argeiphontes Lyre (capture d'écran du logiciel)	25
Figure 7. Schéma de filtre biquadratique, les z sont des retards, les a et b des coefficients	26
Figure 8. Le "patch browser" de Omnisphere, synthétiseur réputé pour ses banques de preset	28
Figure 9. Registres de l'orgue Yves Fossaert	29
Figure 10. Détail du ARP pro Soloist, un des premiers synthétiseurs à preset, sorti en 1972	30
Figure 11. Cartouches de presets pour le DX7	31
Figure 12. Macro contrôles d'un preset de Analog Lab (capture d'écran)	33
Figure 13. Un des exemples du thread	43
Figure 14. Évolution de la plage de valeur	47
Figure 15. Topologies de mapping fondamentales	49
Figure 16. Serge Modular DUSG (gauche) - Mutable Instrument Tides (droite)	50
Figure 17. La même enveloppe sur le DUSG (gauche) et Tides (droite)	51
Figure 18. Utilisation de paramètres intermédiaires	52
Figure 19. L'interpolation agit sur de multiples variables simultanément	53
Figure 20. Fonctionnement du Polybrute de Arturia	54
Figure 21. Modèle masse-ressort comme couche intermédiaire	55
Figure 22. Une des sessions Ableton	60
Figure 23. Un des mappings one-to-one	61
Figure 24. Les oscillateurs de Meld	62
Figure 25. Les deux macro-paramètres du mapping one-to-many, session wavetable	63
Figure 26. PresetMorph	63
Figure 27. Hystérésis type et implémentation par l'utilisation d'un exposant variable	64
Figure 28. Domaine d'expérimentation (gauche), Appréciation des sons proposés (droite)	68
Figure 29. DAW comme outils principal (gauche), Fréquence d'utilisation des synthétiseurs (droite)	68
Figure 30. Fréquence d'utilisation de l'automation en phase d'exploration (gauche) et d'arrangement (droite)	68
Figure 31. Fréquence d'utilisation de preset (gauche) et fréquence de leur modification (droite)	68
Figure 32. Quels instruments seriez-vous susceptible d'utiliser dans vos compositions musicales personnelles ?	
70	

Figure 33. Utilisation dans les compositions musicales personnelles croisé avec le domaine d'expérimentation	70
Figure 34. Utilisation dans les compositions musicales personnelles croisé avec le type de session	71
Figure 35. Certains paramètres macro vous ont-ils semblés inhabituels ?	72
Figure 36. Caractère inhabituel des paramètres croisé avec l'utilisation de presets et de synthétiseurs	73
Figure 37. Auriez-vous aimé avoir plus de temps avec certains instruments ?	73
Figure 38. Quel instrument aimeriez-vous le plus automatiser ?	74
Figure 39. Choix du mapping en fonction du contexte	75
Figure 40. Quels sont les instruments qui vous ont semblé les plus amusants à utiliser ?	75
Figure 41. Pad normal (gauche), pad contenant des ruptures (droite)	80

# Annexes

## Synthétiseurs des années 90



*DX7 (1983)*

<https://review.wolfarchitects.design/wp-content/uploads/2019/01/DX7-1-1.jpg>

[consulté le 14.05.2024]

*Korg M1 (1988)*

<https://cdn.mos.cms.futurecdn.net/49d89c7f50b06195b4179e36356da3d5.jpg> [consulté le 14.05.2024]

*Roland D-50 (1987)*



<https://www.keyboards.de/app/uploads/2016/07/3-1024x685.jpg>

[consulté le 14.05.2024]

*Kawai K1 (1989)*

<https://files.soniccdn.com/images/products/640/398/kawai-k1-73398.jpg>

[consulté le 14.05.2024]

## Questionnaire

\* Indicates required question

1. Composez vous de la musique ? \*

OUI

NON

*Si la réponse est non, passer directement à la section 2*

2. Le DAW (plug-in compris) est votre outil de travail principal

1    2    3    4    5    6    7

Pas du tout

Absolument

3. Vous utilisez des synthétiseurs

1    2    3    4    5    6    7

Jamais

Toujours



8. Quels sont les domaines dans lesquels vous aimez expérimenter ? (possibilité de cocher plusieurs cases)

- Rythme
- Harmonie
- Sound-design
- Arrangement
- Mix
- Other:

9. Précisions sur les questions précédentes ou tout ajout vous semblant pertinent.

10. Type de session

- Sub
- FM
- Meld
- wave

11. Les sons proposés vous ont plu



12. Vous avez utilisé

- Le pad avec la souris
- Les potentiomètres avec la souris
- Les potentiomètres du clavier MIDI
- Le clavier du clavier midi
- Le clavier de l'ordinateur

*Les instruments ne vous ont pas forcément été présentés dans l'ordre, vérifiez bien la lettre associée à chaque instrument*

13. Quels sont les instruments qui vous ont semblé les plus amusants à utiliser ? (possibilité de cocher plusieurs cases) \*

- A (Rouge)
- B (Vert)
- C (Jaune)
- D (Bleu)
- Aucun
- Ne sais pas

14. Auriez-vous aimé avoir plus de temps avec certains instruments ? (possibilité de cocher plusieurs cases)

\*

- A (Rouge)
- B (Vert)
- C (Jaune)
- D (Bleu)
- Aucun
- Ne sais pas

15. Quel instrument aimeriez vous le plus automatiser ?  
(possibilité de cocher plusieurs cases)

\*

- A (Rouge)
- B (Vert)
- C (Jaune)
- D (Bleu)
- Aucun
- Ne sais pas

16. De quel produit sonore êtes vous le plus satisfait ?  
(possibilité de cocher plusieurs cases)

\*

- A (Rouge)
- B (Vert)
- C (Jaune)
- D (Bleu)
- Aucun
- Ne sais pas

17. Quels instruments seriez-vous susceptible d'utiliser dans vos \* compositions musicales personnelles ? (possibilité de cocher plusieurs cases)

- A (Rouge)
- B (Vert)
- C (Jaune)
- D (Bleu)
- Aucun
- Ne sais pas

18. Certains paramètres macro vous ont-ils semblé inhabituels ? \* (possibilité de cocher plusieurs cases)

- A (Rouge)
- B (Vert)
- C (Jaune)
- D (Bleu)
- Aucun
- Ne sais pas

19. Quels instruments vous ont semblé les plus appropriés pour \* une utilisation en live ? (possibilité de cocher plusieurs cases)

- A (Rouge)
- B (Vert)
- C (Jaune)
- D (Bleu)
- Aucun
- Ne sais pas

20. Quels instruments vous ont semblé le plus appropriés pour une utilisation en studio ? (possibilité de cocher plusieurs cases) \*

- A (Rouge)
- B (Vert)
- C (Jaune)
- D (Bleu)
- Aucun
- Ne sais pas

21. Autre commentaire sur les instruments ou sur le questionnaire

22. L'utilisation de ces instruments a-t-elle éveillé des réflexions pendant cette expérience ?

Merci pour votre participation !