



Mémoire de fin d'études

*Quels nouveaux paradigmes pour
l'écriture de l'espace sonore ?*

Manuel Deneu

promotion 2013, section Son

Sous la direction de

Anne Sédes et Thierry Coduys

Mémoire présenté publiquement le 17 juin 2013 devant un jury composé de

Gérard Pelé	Rapporteur	Ecole Nationale Supérieure Louis Lumière
Thierry Coduys	Directeur	Ecole Nationale Supérieure Louis Lumière
Anne Sédes	Directrice	Centre de recherche en Informatique et Création Musicale, Paris VIII

Résumé

Ce mémoire s'attache à réfléchir à des moyens alternatifs pour la spatialisation de sources sonores, non par l'emploi de traitements du signal, mais plutôt par l'introduction de nouvelles modalités de construction d'une scène sonore. Notre réflexion s'inscrit donc dans la recherche de nouveaux paradigmes de pilotage d'entités, dont le nombre est sans cesse grandissant, mais aussi de paradigmes d'interfaces pour une construction temporelle et spatiale facilitée, naturelle et musicale.

Mots clés

spatialisation – interface – ergonomie – geste musical – création sonore – composition

Abstract

In this dissertation, we propose a discussion about alternative approaches in sound spatialization by introducing new ways of building an audio scene. This work involves the definition of new paradigms to animate entities, but also to design improved interfaces in order to ease the temporal and spatial construction of a scene.

Keywords

spatialization – interface – ergonomics – musical gesture – sound design
– composition

Remerciements

Je tiens à remercier Anne Sedes de m'avoir accueilli au sein de l'équipe de développement de la librairie HOA ainsi que d'avoir suivi ce mémoire tout au long de son élaboration. De la même manière, je remercie Thierry Coduys pour ses conseils, son suivi régulier de mon projet, et surtout pour m'avoir permis de participer à l'expérience *Iannix*.

Je remercie également Guillaume Jacquemin, pour ses conseils, son attention et la licence de développeur Apple de *Buzzing Light*, que lui et Guillaume Marais m'ont gracieusement mis à disposition.

Un grand merci à Véronique Beaufiles et Marie Bisson, documentaristes au Conservatoire de Caen pour leur précieuse aide dans mes recherches.

Je tiens à remercier mes parents, Patricia Fize et Philippe Deneu, qui comme toujours s'intéressent à mes projets et s'investissent dans leur réalisation. Merci à eux, ainsi qu'à mon frère Adrien pour leurs conseils et leurs patientes relectures.

Enfin, merci à Anna.

Table des matières

Avant propos	vii
Introduction	1
I Principes de mise en espace du son	14
1 L'entité de base : la source sonore	24
2 La notion de groupe	27
3 Paradigmes	34
3.1 Le suivi d'une courbe	37
3.2 La cinétique des gaz	40
3.3 Le modèle des <i>boids</i> - ou "essaim particulaire"	43
3.4 Vers la synthèse d'un modèle	46
4 La question de la gestion du temps	52
II L'interface proposée au créateur	59
5 La partition	64
6 Une interface tangible de dessin	72

7	Une interface temporelle	82
8	Synthèse du système	88
III	Développement logiciel d'une série de prototypes	91
9	Une télécommande pour <i>IanniX</i>	97
10	Une interface de dessin de courbes pour tablette tactile	100
11	Une interface de contrôle de la trajectoire de sources	103
12	Une interface de séquenceur temporel pour ordinateur	116
13	L'implémentation de notre système	122
	Conclusion	128
	Annexes	133
A	Le fichier de sauvegarde d'un projet	135
B	La paramétrisation des courbes	139
C	Le calcul de la position d'un point sur une courbe de Bézier	143
D	La liaison OSC	145
E	Le protocole de communication séquenceur-tablette	149
F	Les codes sources	152
G	Spatialisation : Etat de l'art	154

Bibliographie	159
Table des figures	164

Avant propos

Le projet de ce mémoire repose sur une question qui s'est avérée récurrente tout au long de ma formation musicale et professionnelle et au fil de projets menés depuis longtemps : la question des conditions d'une écriture musicale permettant de noter tout type de musique, prenant en charge des modalités de production et de diffusion.

Ce mémoire se fonde donc sur une synthèse de plusieurs projets de développement sur lesquels je travaille actuellement, et qui portent sur la création d'interfaces pour des outils liés à la spatialisation. On peut citer le logiciel IanniX¹, un séquenceur graphique open-source inspiré des travaux de Iannis Xenakis et destiné à la création numérique ou encore la librairie d'objets *Max* HOA² dédiée au système ambisonique, créée par le Centre de recherche Informatique et Création Musicale (CICM). Ces différents projets, ainsi que ma pratique musicale personnelle, m'ont non seulement permis de prendre conscience de la diversité des outils de création sonore, mais aussi de la complexité de leur utilisation au sein d'un environnement de travail.

L'approche employée ici relève d'une *analyse systémique*, d'abord parce que tout en restant ancrée dans une recherche "fondamentale", celle-ci permet

1. <http://www.iannix.org/>

2. <http://www.mshparisnord.fr/hoalibrary/>

d'élargir les champs de l'investigation, de favoriser des relations inter-disciplinaires et inter-catégorielles mais aussi, et c'est ce que nous allons constater tout au long de ce mémoire, parce que cette approche et l'objet de notre étude partagent des concepts tels que le structuralisme, la gestion de l'information ou encore l'élaboration d'échelles communes à différents niveaux d'organisation.

Ainsi, ce travail de recherche se veut-il résolument pragmatique. Pragmatique dans le sens où les idées exposées ici seront testées en parallèle, par leur développement logiciel effectif, mais aussi parce qu'il s'agit de proposer un outil de création qui soit résolument *user-friendly*, en offrant aux utilisateurs des interfaces innovantes, non pas fondées sur des concepts exclusivement techniques relevant de l'ingénierie sonore, mais davantage fondés sur une démarche d'écriture de la musique dont les caractéristiques et les enjeux sont essentiellement ceux de la création artistique.

Introduction

«Il est impératif de contenir la complexité et la complication des solutions autant que faire se peut, sans quoi l'homme perd la maîtrise d'un outil dont il ne domine plus le concept et que toute tentative de modification conduit à la dégradation gracieuse ou le plus souvent, disgracieuse.»

– Paul Jorion, in *Principes des systèmes intelligents*³.

La plupart des outils de spatialisation disponibles aujourd'hui, bien que très évolués en termes de capacité de rendu, sont très souvent limités, voire inhibants pour ce qui concerne la prise en main et la dimension peu intuitive de l'interface. Cette dernière est bien souvent d'approche très technique, trop compartimentée ; mais sa principale lacune consiste sans doute en l'absence de dispositifs facilitant l'interaction entre les sources elles-mêmes. De plus, la puissance de calcul des systèmes sonores est devenue telle qu'il est assez courant, dans les projets actuels, d'avoir une centaine d'objets sonores à spatialiser. Or ces logiciels ne facilitent pas la réalisation, l'écriture devenant rapidement fastidieuse car trop technique, mais également peu fructueuse sur le plan de la créativité. Dans la création sonore, et tout particulièrement dans le domaine de la musique électro-acoustique, il n'est pas toujours nécessaire d'avoir un contrôle total sur tous les paramètres utilisés ; nous pouvons même rechercher un effet global dans lequel les fluctuations seraient

³. *Principes des systèmes intelligents*, Paul Jorion, Editions Du Croquant 2012, page 13.

aléatoires. C'est particulièrement vrai lorsqu'un grand nombre de sources est mis en jeu, et qu'une hiérarchie d'importance a tendance à émerger au sein des entités sonores : certaines d'entre-elles relèvent alors de ce que l'on pourrait appeler de manière assez péjorative le « fond sonore » ou encore « l'habillage » et peuvent tout à fait être contrôlées de manière assez globale, au contraire des sources « solistes » qui nécessitent une intention, et une attention toute particulière.

L'enjeu de ce travail est donc de proposer un système d'écriture de la mise en espace du son, qui sera fondé sur une approche musicale, et qui incorporera une dimension performative forte. Cependant, on fera le choix de se dégager volontairement de la question du moteur de spatialisation employé, et plus généralement de l'environnement sonore, en supposant que le système produit pourra s'insérer à n'importe quel endroit de la chaîne de production sonore et qu'il utilisera les standards de communication universels systématiquement employés. Il ne s'agit en aucun cas d'éluder la question fondamentale de l'algorithme de spatialisation effectivement employé, mais plutôt de faire le pari qu'un "simple" système adaptatif placé entre notre système d'écriture et le moteur audio pourra faire la transcription des données.

Une hypothèse de travail

«Already upon hearing two synchronized layers, [...] I became increasingly pale and helpless : I had imagined something completely different ! »

– Stockhausen, à propos de *Etude* (1952).

La question de la différenciation actuelle des données manipulées d'une part par la pensée lors de l'idéation d'une œuvre, et d'autre part par la

réalisation, lorsqu'il s'agit de la « coucher » sur un support, semble centrale dès lors que l'on cherche à définir un outil de composition. Du côté de la linguistique, on manipule un seul et même langage, dont l'unité de base, ou atome, est le *sème*⁴; et même si les supports de communication sont multiples pour une même langue, il nous semble que le passage de l'écrit au parlé, de la lecture à l'écoute se fasse de manière naturelle, transversale. En musique la question du passage de l'oral à l'écrit a toujours été essentielle. C'est la base de l'invention du système *neumatique*, dont l'évolution conduira au système musical occidental. Cependant, l'avènement d'une nouvelle musique, dont l'ensemble instrumental n'est plus codifié ni canonique, dont les modes de jeu ont évolué, et surtout qui fait appel à la notion de dispositif, de mise en espace demande une certaine adaptation tout au long de la genèse d'une œuvre.

Parce que le monde sonore n'obéit pas forcément à une structure syntaxique, il semble très difficile pour le compositeur de pouvoir matérialiser sur une feuille de papier ses idées musicales. De fait, c'est de l'immensité du domaine du « bruit » et de l'impossibilité de se le représenter que vient le problème. Comment organiser, hiérarchiser un ensemble d'événements sonores autrement que par tâtonnements, collages et expérimentations diverses ? C'est ce à quoi répond Stockhausen *supra* à propos de sa pièce de musique concrète *Etude*, qui lui a été la cause d'un désespoir⁵ certain. Et quand bien même le compositeur arrive au terme d'un projet, c'est bien souvent sa réalisation qui se trouve hasardeuse voire brutalement avortée. John Cage raconte : « *J'ai demandé [au directeur des effets sonores de CBS] quels sons je pouvais utiliser et il a dit qu'il n'y avait aucune limite à ce que je pourrais faire. Alors j'ai commencé [...]. J'écrivais verbalement et musicalement*

4. Terme introduit par le linguiste Éric Buyssens

5. De manière officieuse, la pièce se nomme d'ailleurs *Etude aux mille collants*.

*et quand je les ai apportés à l'homme des effets sonores, il a osé me dire que ce que j'avais écrit était impossible. »*⁶ On peut facilement imaginer combien un support autre qu'une écriture verbale et musicale, pour reprendre les termes de John Cage, aurait permis une communication plus aisée avec le technicien de studio.

Afin de concevoir une éventuelle solution à ce problème d'intermédiaire, de traducteur entre la pensée d'un compositeur et sa réalisation effective, il peut être intéressant de prendre le cheminement classique de création à rebours, c'est-à-dire à partir de l'expérience perceptive de l'auditeur. Dans son article *La dynamique pré-réfléchie de l'expérience vécue*⁷, Claire Petitmengin cherche à démontrer l'existence de mécanismes, conscients ou non, qui dans notre perception jouent un rôle important à la compréhension, à la construction d'un sens. Parmi ces mécanismes, elle s'attache à un état intermédiaire de la perception :

*« Par expérience pré-réfléchie, j'entends la part de notre expérience qui est vécue sans être reconnue, sans être immédiatement accessible à la conscience et à la description verbale. C'est la part de notre expérience qui est ordinairement oblitérée par l'absorption de notre attention dans le contenu, l'objet de notre activité. Par exemple, lorsque je fais un mouvement, mon intérêt pour l'objet vers lequel le mouvement est dirigé — le ballon, la pomme — occulte le mouvement de mon bras, qui lui-même occulte les sensations internes que j'éprouve dans les muscles et les articulations qui initient en fait le mouvement de mon bras. [...] l'absorption dans l'objet ou l'objectif, le "quoi" du processus, occulte le "comment", qui reste ordinairement non remarqué, non reconnu. »*⁸

6. *Conversations avec John Cage*, Richard Kostelanetz, Editions des Syrtes 2000.

7. *La dynamique pré-réfléchie de l'expérience vécue*, Claire Petitmengin, Alter : revue de phénoménologie 2012

8. *op. cit.*

La question est donc de savoir si l'aspect pré-réfléchi de notre perception ne pourrait pas avoir une correspondance dans le domaine de la création, ou autrement dit s'il ne serait pas possible qu'un artiste ait des "réflexes de création" qu'il appliquerait, de manière consciente ou non. Il n'est cependant pas question d'aborder tous les procédés qui relèvent du "style" d'un auteur, ces points communs qu'il est possible d'extraire d'un ensemble d'œuvres, mais plutôt d'aller dans le sens de ce que Kant appelle "*productions du génie*"⁹, et qu'il réfère à "*ce pour quoi on ne saurait donner de règle déterminée*"¹⁰, c'est-à-dire justement à des mécanismes conscients, ou non.

L'existence de mécanismes transitoires à la perception est donc au cœur même de notre réflexion, car elle fournit un élément de réponse à la question de la pertinence d'un ensemble de systèmes intermédiaires entre l'idée exprimée et sa réalisation effective, établissant un lien sémantique entre le langage artistique de la pensée, et celui — technique — de la réalisation sonore.

Hypothèse heuristique : considérons qu'il doit être possible de modéliser un comportement *pré-réfléchi* dans un système d'écriture, dont le principal but serait de rendre opaque à l'utilisateur le fonctionnement sous-jacent d'un processus, afin de n'en garder que ce qui relève de la perception.

C'est cette hypothèse qui justifie l'introduction de paradigmes, dont nous allons maintenant tenter de définir le cadre. Notre hypothèse sera validée dès lors que nous serons parvenu à proposer un système d'écriture dont l'utilisation sera intuitive, spontanée, naturelle grâce à une dimension motrice préexistante favorisant la création.

9. Emmanuel Kant, *Critique de la faculté de juger*.

10. *op. cit.*

De nouveaux paradigmes pour l'écriture de l'espace sonore.

Il nous faut donc maintenant tenter de dégager plusieurs principes¹¹ auxquels le système d'écriture devra répondre. L'*extension* de notre titre doit alors dire les termes les conditions de notre projet.

Avec l'adjectif « nouveau » nous insistons sur notre souhait d'avoir une approche alternative de ce que peut être un tel système, sans pour autant différer radicalement de ce qui existe déjà. Peut être faut-il plutôt interpréter cette innovation comme « ce qui succède à une chose du même ordre », ou plus simplement encore ce qui la renouvelle.

Le terme de paradigme définit, dans le domaine scientifique, l'approche avec laquelle un problème peut être étudié, et qui va de l'observation d'un phénomène à son interprétation, en passant par la structuration des données et des inconnues. Un paradigme n'est jamais unique puisqu'il dépend d'un point de vue, mais surtout il peut être réfuté. Les sciences sociales associent le paradigme à une vision du monde donnée, à l'ensemble des perceptions, expériences ou encore valeurs ayant une incidence sur la façon dont on perçoit la réalité et interagit avec elle. L'utilisation de ce terme veut refléter le changement de point de vue nécessaire à l'appropriation de la spatialisation sonore au domaine de la musique. Il ne nous faut donc plus raisonner sur les entités géométriques que sont les sources ponctuelles, mais sur des systèmes de mise en espace dont la sémantique doit être en adéquation avec l'effet sonore et/ou musical désiré. C'est cette exigence que pose le premier critère.

11. On prends ici le terme de *principe* au sens de ce qui est posé vrai et nécessaire, à la manière d'un axiome sur lequel une autre connaissance peut être construite.

Principe 1. *La mise en espace de sources se fait via une interface, appelée Paradigme, dont la fonction est de faire le lien entre l'intention artistique de l'utilisateur et la représentation interne des données.*

L'emploi du terme "d'écriture" renvoie à un « système de signes à caractère duratif ayant un support visuel et spatial »¹². Le « signe » linguistique se définit selon Ferdinand de Saussure¹³ comme le lien indissociable, mais arbitraire, entre le signifiant, image graphique et sonore, le signifié, concept désigné par le signifiant et le référé, l'objet précis désigné par le locuteur. A la suite de Saussure, André Martinet s'attache également à décrire l'agencement des signes selon une double articulation entre des unités significatives, unités de sens ou « morphèmes » et des unités distinctives, unités de son, « les phonèmes ». C'est la combinaison, l'articulation de ces deux axes, qui explique les potentialités infinies d'expression d'une langue humaine. Pour posséder une richesse artistique qui ne le cantonne pas dans l'éphémère, le singulier, un système doit proposer une combinatoire stable de ses éléments, une possibilité d'agencement et de création qui lui permette de se renouveler sans cesse. Ainsi nous faut-il définir une sémantique, dont la finalité serait de donner sens à la distribution du son dans un espace. Son fondement reposerait sur la définition et les relations entre des syntagmes, des briques, constitutifs de notre système. Grâce à la double articulation d'un lexique, composé de paradigmes, et d'une syntaxe, il devient possible d'inventer un langage dont le sens est *univoque*, et surtout qui constitue un « acte de langage »¹⁴ mis en œuvre d'un moyen par l'intermédiaire d'un langage *prescriptif*. Anne Veitl, dans son ouvrage consacré à la notation dans

12. Ducrot Oswald et Schaeffer Jean-Marie, *Nouveau dictionnaire encyclopédique des sciences du langage*, Seuil, 1995.

13. Ferdinand De Saussure, *Cours de linguistique générale*, Payot 1975.

14. Théorie de la communication élaborée dans les années soixante par J.L. Austin, et qui propose que certains énoncés du langage ne servent ni à transmettre un message, ni à décrire le monde, mais à faire quelque chose.

la musique contemporaine¹⁵ énonce les critères suivants pour qu'un système symbolique constitue un mode de notation écrite :

- Visibilité : décryptage des informations immédiates, « l'ensemble ou une partie des informations mémorisées [...] doit pouvoir être visible, à l'oeil nu, ou, à tout le moins, facilement visualisable. »
- Lisibilité : codes graphiques en adéquation avec ce à quoi ils se réfèrent. Caractère univoque de chaque notation, « non-ambiguïté sémantique » selon les termes du philosophe Nelson Goodman dans sa *Théorie de la notation*.
- Performatif : « potentiel d'action, d'expression et de création d'un système d'écriture ».

Principe 2. *La sémantique employée dans l'écriture de l'espace sonore doit être en lien avec le contenu artistique et avoir un aspect performatif fort. Sa matérialisation sera prescriptive, mais aussi descriptive pour permettre une création aisée, intuitive et souple.*

Enfin, la notion "d'espace sonore" désigne une étendue indéfinie ou limitée dans laquelle évolue un champ sonore. Le terme est assez flou, principalement parce que ce à quoi il réfère n'est pas assez bien défini. Ainsi peut-on parler du champ sonore créé par une paire d'enceintes stéréo ou encore de l'espace sonore d'un lieu, même si dans le premier cas, l'espace sonore créé par les transducteurs est couplé à celui du lieu d'écoute. L'absence d'espace sonore absolu n'est cependant pas nécessairement une barrière insurmontable et comme écrit précédemment, on peut gager qu'il est possible d'établir un système de conversion et de calibrage des données géométriques

15. *Notation écrite et musique contemporaine : quelles grandes caractéristiques des technologies numériques d'écriture musicale ?*, Anne Veitl, Observatoire Tscimuse-Grenoble, Lyon 2007.

entre l'espace d'écriture et celui du moteur de spatialisation. Le second aspect de l'espace sonore est celui de la temporalité. Dans la mesure où les sources sonores ainsi que les paradigmes employés ne sont pas statiques, il est nécessaire d'intégrer la notion de déroulement temporel.

Principe 3. *Le système d'écriture doit intégrer une gestion du temps qui soit à la fois absolu, dans le cas d'une écriture de partition, ou variable, dans le cas d'une utilisation temps-réel.*

Des échelles différentes

Afin de poursuivre la définition du paradigme comme la vision multiple d'une même chose, il peut être intéressant d'explorer un pan de la musique contemporaine qu'est la musique stochastique. A partir de 1954, le compositeur grec Iannis Xenakis va mettre au point un nouveau concept de composition musicale et sonore dont la base est la définition globale « d'états sonores successifs dont est faite la musique »¹⁶, le détail de ces différents états sonores étant généré, entre autres, par le calcul des probabilités. Constatant l'essoufflement du sérialisme intégral, et plus généralement de tous les courants musicaux issus de la volonté de sortir du système tonal, Iannis Xenakis justifie, dans son ouvrage *Musiques Formelles*¹⁷, l'introduction des mathématiques en musique :

« La polyphonie linéaire se détruit d'elle-même par sa complexité actuelle. Ce qu'on entend n'est en réalité qu'amas de notes à des registres variés. La complexité énorme empêche à l'audition de suivre l'enchevêtrement des lignes et a comme effet macroscopique une dispersion irraisonnée et fortuite des sons sur toute l'étendue du spectre sonore. Il y a par conséquent contradiction entre le système polyphonique linéaire et le résultat entendu qui est surface, masse. Cette contraction inhérente à la polyphonie disparaîtra

16. Cité de l'article « Musique Stochastique » de l'Encyclopédie Larousse ».

17. *Musiques Formelles*, Iannis Xenakis, La revue musicale, 1963.

lorsque l'indépendance des sons sera totale. En effet, les combinaisons linéaires et leurs superpositions polyphoniques n'étant plus opérantes, ce qui comptera sera la moyenne statistique des états isolés et des transformations des composantes à un instant donné. L'effet macroscopique pourra donc être contrôlé par la moyenne des mouvements des objets choisis par nous. Il en résulte l'introduction de la notion de probabilité, qui implique d'ailleurs, dans ce cas précis, le calcul combinatoire. Voilà, en peu de mots, le dépassement possible de la " catégorie linéaire " de la pensée musicale.
»¹⁸

Vers la même époque, en 1963, Pierre Boulez dans son ouvrage *Penser la musique aujourd'hui*¹⁹ constate à propos de la musique sérielle généralisée la même impasse face à un système trop contraignant :

« Le compositeur fuyait sa responsabilité dans le choix, la détermination, pour la faire assumer par une organisation numérique. »

La solution préconisée par le compositeur pour dépasser ce problème est soit de « faire craquer le système de l'intérieur en ne demandant aux nombres que ce qu'ils peuvent nous donner — c'est-à-dire fort peu », soit d'« esquiver les difficultés par le libertinage, en se justifiant par des considérations psychologiques et parapsychologiques somme toute assez banales » .

Ce qu'il y a de commun entre Xenakis et Boulez, c'est la volonté de créer des méthodes de travail sur le matériau sonore qui soient, d'une part efficaces, mais surtout appropriées par le compositeur comme éléments structurants de la forme. Bien qu'opposés sur de nombreux points, tous deux semblent s'accorder sur l'idée d'*objet complexe* fait "de milliers de sons isolés dont la

18. Cet extrait est en fait une citation de l'auteur lui-même, du premier opus des *Gravesaner Blätter*, série de publications éditée par Hermann Scherchen en 1954. L'ouvrage *Musiques Formelles* en est une compilation revue de cette série d'articles.

19. *Penser la musique aujourd'hui*, Pierre Boulez, Editions Gonthier, Gallimard 1963.

multitude crée un événement sonore nouveau sur un plan d'ensemble"²⁰ et *"ayant ses exigences propres réclamant des lois d'organisation dérivées, bien sûr, d'une loi première mais organiquement spécifique"*²¹. Dans le cas de Boulez cette loi première est le *sérialisme intégral* et dans le cas de Xenakis, ce sont les *lois stochastiques*.

Le principe suivant repose sur une définition de Pierre Boulez dans *Penser la musique aujourd'hui* et pose le concept de différences d'échelles :

*« les relations verticales [les relations définies entre plusieurs entités ponctuelles] se conçoivent comme matériel direct de travail, comme intermédiaire dans l'élaboration d'objets complexes, soit encore comme supervision du travail sur les objets complexes [...] »*²²

Principe 4. *Le système d'écriture doit proposer plusieurs niveaux de manipulation des données musicales, quel que soit le domaine du paramètre à éditer, qu'il soit temporel, spatial ou autre. Ce principe propose donc les échelles microscopiques et macroscopiques comme étant des représentations canoniques.*

Vers un structuralisme des données

«Le contenu tire sa réalité de la structure, et ce qu'on appelle forme est la mise en structure des structures locales.»

– Claude Lévi-Strauss, *in Anthropologie structurale II*, Pocket 1997.

L'approche systémique, qui nous semble stimulante, propose une analyse de la structure fondée sur l'ensemble des relations qu'elle impose aux entités qui la constituent, et non sur la description groupée de ces mêmes entités. Or tous les principes convergent vers l'idée de manipulation de structures

20. Iannis Xenakis, *op. cit.*

21. Pierre Boulez, *op. cit.*

22. Pierre Boulez, *op. cit.*

au travers d'interfaces, dont les niveaux de représentations sont eux-mêmes structurés. Il convient donc de définir comment l'organisation sémantique va s'effectuer au sein d'un système.

D'abord énoncé par Ferdinand De Saussure dans son *Cours de linguistique générale*²³, le concept sera plus tard appliqué à l'anthropologie par Claude Lévi-Strauss qui en dégagera l'idée de *structure élémentaire de parenté*²⁴, le sous-ensemble élémentaire à son étude des sociétés.

Cette opposition de la structure à l' "atome" est le propre de la musique stochastique, et plus généralement de l'étude de tout système complexe. C'est aussi l'idée de base de la mécanique quantique que de travailler sur une *probabilité de présence*, d'un atome plutôt que sur sa position effective, qui ne peut de toute manière être connue avec certitude. La proposition de paradigmes pour l'écriture de l'espace sonore semble aller dans ce sens, celui de vouloir privilégier des "entités de dépendances internes"²⁵, plutôt que des groupements d'atomes.

Principe 5. *Afin de compléter le concept de paradigme appliqué à un ensemble, il est nécessaire d'introduire une organisation structurelle des sources, dont l'attribution à tel ou tel groupe pourra être faite de manière dynamique, voire informelle et dont la définition en tant que structure sera établie par les relations induites par ses constituants.*

On peut noter que ce principe ouvre la possibilité d'attribuer les sources à des structures de façon non rigide, et pourquoi pas de conduire à la notion nouvelle de *probabilité d'appartenance d'une source* à un groupe, ou de manière encore plus générale d'imaginer une structure de paradigme pour

23. Ferdinand De Saussure, *Cours de linguistique générale*, Payot 1975.

24. concept défini dans l'ouvrage *Les structures élémentaires de la parenté*, Claude Lévi-Strauss, Mouton 1967.

25. Concept introduit par Louis Hjelmslev dans son livre, *Prolegomènes à une théorie du langage*, Les Editions de minuit 1968

l'attribution des sources à leurs structures.

Dans les chapitres suivants, nous allons construire pas à pas les éléments de notre système d'écriture de l'espace sonore. Les critères proposés *supra* serviront de base à l'élaboration des différents principes mis en place. La première partie porte sur la définition des différents éléments fondamentaux, la seconde est consacrée à l'interface proposée à l'utilisateur, tandis que la troisième partie se veut la synthèse générale du système, sous la forme d'un développement d'une série de prototypes logiciels.

Première partie

Principes de mise en espace
du son

La musique dans sa relation à l'espace

Traditionnellement, la configuration du concert de musique sépare les musiciens du public. Chacun étant à sa place, la scène est un espace délimité, sacralisé et la disposition face à face rend la dimension spatiale du son réduite à une simple relation à la distance, dépendant également des caractéristiques acoustiques de la salle. Cependant, la question de l'espace sonore apparaît très tôt dans la musique occidentale, tout en restant relativement anecdotique dans son utilisation. On peut d'abord citer l'emploi d'un double chœur, disposé dans deux tribunes opposées dans les œuvres *SacræSymphoæ* de Giovanni Gabriely²⁶ en 1597 et dans la *Passion selon Saint Mathieu* de Bach en 1727. En 1830, Hector Berlioz sera le premier à employer un instrument de musique en le cachant du public dans sa *Symphonie fantastique*. Dans le troisième mouvement, la *Scène aux champs*, le cor anglais et un hautbois — en coulisse — conversent autour d'un même thème créant un effet d'orchestration d'autant plus saisissant qu'une des sources n'est pas révélée au public.

Mieux, dans le Requiem de sa *Grande messe des morts Op.5*, Berlioz propose quatre petits orchestres placés aux points cardinaux autour de l'orchestre et du chœur :

« *Mes exécutants étaient divisés en plusieurs groupes assez distants les uns des autres et il faut qu'il en soit ainsi pour les quatre orchestres d'instruments de cuivre que j'ai employés dans le « Tuba murum » et qui doivent occuper chacun un angle de la masse vocale et orchestrale.* »²⁷

L'effet produit est étonnant : la contribution de chaque petit orchestre

26. On peut d'ailleurs noter que le compositeur Giovanni Gabriely est considéré comme le premier à avoir précisément indiqué sur ses partitions à quel instrument était réservée une partie musicale, la pratique ayant été jusque-là de jouer les oeuvres avec les instruments à disposition.

27. Hector Berlioz, *Mémoires*, C. Lévy 1878.

III.
Auf dem Lande.
Scène aux champs. In the country.

Adagio. (♩ = 54.)

2 Flauti.
Oboe.
Corno inglese.
(= Oboe II.)
2 Clarinetti in B (Si \flat).
Corni I. II in F (Fa).
Corno III in Es (Mi \flat).
Corno IV in C (Ut).
4 Fagotti.

FIGURE 1 – Extrait du conducteur de la *Symphonie Fantastique*

dans des entrées successives, est nettement localisée dans l'espace jusqu'à complètement fusionner sur un accord saisissant, qui semble alors provenir du centre de l'orchestre.

L'effet employé par Berlioz peut préfigurer un courant musical qui va se développer au XX^e siècle, et qui est qualifié de *Musique Acousmatique*. Issu des recherches débutées dans les années 1940 par Pierre Schaeffer au "Club d'Essai de la Radio", ce courant musical d'abord qualifié de *concret* joue sur la possibilité — nouvelle à l'époque — de manipuler des sons *via* leurs supports d'enregistrement, d'abord le vinyle puis rapidement la bande magnétique. L'écoute des pièces musicales ainsi créées s'effectue alors par le biais de haut-parleurs, tandis que le support est figé sur une bande de magnétophone. A ce propos, Michel Chion parle d'*art des sons fixés*²⁸. Le terme de musique *acousmatique* sera proposé en 1955 par le poète Jérôme Peignot pour désigner «*cette distance qui sépare les sons de leur origine*». La

28. Michel Chion, *La Musique concrète, art des sons fixés*, Entre-deux/Môméludies 2010.

référence au terme grec *Akousma* n'est d'ailleurs pas anodine, si l'on fait le rapprochement entre la manière dont Pythagore dispensait ses cours, caché derrière un rideau, et le fait que la musique concrète a recours à la diffusion de ses pièces par haut-parleurs, cachant ainsi la source originelle des sons.

Parallèlement au développement de la musique concrète/acousmatique, d'autres expérimentations sont effectuées dès 1951 au studio de musique électronique de la WDR (Westdeutscher Rundfunk) à Cologne par Herbert Eimert. C'est dans cet environnement de création que sortira en 1956 l'œuvre de Stockhausen, *Gesang der Jünglinge im Feuerofen (Dritte Elektronische Studie)*.

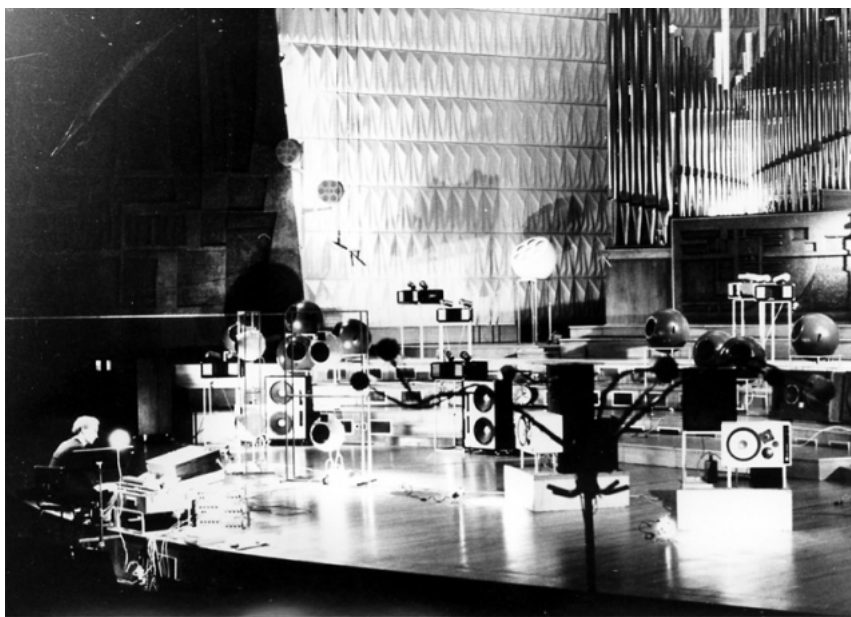


FIGURE 2 – L'*Acousmonium* du GRM salle Olivier Messiaen. *Photo : François Bayle*

Ces deux courants de la musique électro-acoustique useront abondamment des systèmes de multi-diffusion dont l'expérimentation sera conduite dans les années soixante-dix et quatre-vingt. On peut notamment citer le

Gmebaphone, un instrument de diffusion sonore conçu au Groupe de musique expérimentale de Bourges en 1973, dont le principe préfigure quelque peu celui de l'*acousmonium* mais qui s'en oppose par le fait qu'il cherche à recréer un espace plus "global". L'*acousmonium*, quand à lui, voit le jour en 1974 au Groupe de Recherche Musicale (GRM) et François Bayle en est un idéateur :

*«Disposés en séries d'"écrans sonores" multiphoniques, variés en calibres, distances, directions, [les enceintes acoustiques] aident à organiser l'espace acoustique selon les données de la salle, et l'espace psychologique selon les données de l'œuvre. Aménageant tutti et soli, nuances et contrastes, reliefs et mouvements, le musicien au pupitre devient le concepteur d'une orchestration et d'une interprétation vivante. »*²⁹

L'Acousmonium n'est donc pas un dispositif de spatialisation : c'est avant tout un système de jeu avec les différents timbres et couleurs de chaque système de diffusion. La "fidélité" de restitution sonore ne compte pas, c'est justement la "touche" sonore introduite par le transducteur, sa position dans la salle et son orientation que l'on souhaite exploiter. L'effet de spatialisation en est la conséquence, de par la répartition souvent éloignée de chaque enceinte.

La mise en espace

Ce qui revient systématiquement dans tous les exemples cités *supra*, c'est l'idée de dépassement d'un simple positionnement des sources sonores dans l'espace : il y a dans ces œuvres une intention de "connoter" une source sonore, que ce soit en la cachant pour créer un effet inattendu dans l'orchestration ou encore en la projetant sur un système de haut-parleurs dont la coloration et le placement viennent s'ajouter aux qualités sonores de la source.

29. *Musique acousmatique : propositions... positions*, Buchet/Chastel Paris, 1993

On peut aussi évoquer des effets qui n'impliquent pas l'idée de spatialisation comme point de départ, mais comme conséquence. Citons la *Klangfarbenmelodie*³⁰, chère à la Seconde Ecole de Vienne, qui consiste à distribuer note par note un thème à différents groupes d'instruments. Il en résulte alors une sensation de déplacement, de "pointillisme" du thème au sein de l'orchestre. Pour caractériser une esthétique musicale, le concept de « spatialisation » sonore semble ainsi peu opportun. En effet, celui-ci connote un aspect bien trop technique pour être envisagé dans la phase d'idéation et d'écriture d'une oeuvre sonore. De la même manière que l'on oppose l'orchestration à l'instrumentation, on peut alors proposer le concept de *mise en espace* pour décrire la démarche artistique, non seulement de placement des sources sonores dans l'espace mais aussi de création de textures, de masses, de timbres répartis dans l'espace. La spatialisation relève alors du dispositif technologique sous-jacent à cette mise en espace.

Aujourd'hui, les pratiques de la musique électro-acoustique et de la musique dite mixte vont vers une uniformisation des outils de travail, notamment en ce qui concerne la mise en espace d'une oeuvre sonore. On constate que les héritiers de la musique concrète ont intégré l'idée de spatialisation d'une source par l'utilisation presque normée d'une, voire de plusieurs couronnes de haut-parleurs en complément de l'*acousmonium* classique. La spatialisation n'est pas une fin en soi, elle doit servir le propos musical, venir le renforcer et son intentionnalité est très forte. Il semble alors nécessaire d'en revoir l'approche, en proposant pour le compositeur une philosophie de gestion de l'espace fondée sur une *intention* musicale, et non sur une approche *technique*.

30. Terme allemand. Littéralement, jeu de mélodie et de timbre.

Les éléments de travail de la mise en espace

Nous allons maintenant nous intéresser aux différentes briques constitutives d'un environnement de travail du son. Les formats de diffusion sonore pour la musique sont multiples (stéréophonie, quadriphonie, 5.1, octophonie, etc.) et dépendent du moteur de spatialisation employé. En revanche, ces mêmes formats sont également manipulés en tant que *sources sonores* au sein d'une œuvre et constituent donc le point d'entrée de la spatialisation. Ce simple constat pose déjà deux problématiques :

- Celle des dimensions et de l'étendue de l'espace de restitution du champ sonore.
- Celle d'un système de description de la position des sources dans l'espace.

A propos des dimensions de l'espace, et face au constat qu'une majorité des systèmes de diffusion actuels se font selon un plan idéalement parallèle au sol et à hauteur de tête de l'auditeur, on est tenté de répondre qu'une simple description de l'espace à deux dimensions est suffisante à une restitution standard d'une scène sonore. Cependant, l'idée de prendre en compte l'élévation sonore à la restitution n'est pas nouvelle ; elle a déjà été largement testée³¹ et semble bien sur la voie d'une normalisation avec des formats tels que le Dolby *Atmos*, la sphère ambisonique (la *AlloSphere* de l'université de Santa Barbara, Californie) ou encore la W.F.S³², notamment à l'Ircam et plus récemment dans la salle de cinéma *Grauman's Chinese Theatre* à Los Angeles. D'une manière générale, on peut privilégier le choix d'un espace

31. *Multicanal au cinéma : un canal zénithal ?* Mémoire de fin d'étude de l'ENS Louis Lumière rédigé par Aymeric Dupas, 2007.

32. Le système W.F.S n'est pas à proprement parler un système 3D mais on le trouve souvent complété par un système de restitution azimutal, système ambisonique dans le cas de l'Espace de projection de l'Ircam.

à trois dimensions spatiales comme étant le moins limitatif et donc le plus ouvert.

Quant à la problématique d'un système de description de la position spatiale d'une source, la question en est plus délicate. En effet, il existe une multitude de systèmes de coordonnées pour la description de l'espace, les plus répandus étant les systèmes cartésiens, sphériques ou encore cylindriques. Le fait que beaucoup de spatialisateurs proposent à l'utilisateur une description polaire ou cartésienne d'une même scène est purement pratique. Si l'on souhaite faire tourner une source à rayon constant autour d'un point, le système polaire est plus judicieux puisqu'il ne fait varier qu'un seul paramètre (l'angle θ) là où le système cartésien doit faire varier les deux paramètres x et y .

Cependant, une description qui relève de la géométrie analytique contribue à notre sens à l'opacité et au manque de lisibilité des outils de création musicale proposés jusqu'ici. En cohérence avec notre projet initial qui veut que l'on ne fasse pas manipuler les sources directement par l'utilisateur, on fait alors le choix — arbitraire — du système de coordonnées cartésien comme description interne de l'ensemble des constituants de l'espace sonore, dont le point d'origine se situe au centre de la scène.

$$P \equiv \begin{cases} x \in \mathfrak{R} \\ y \in \mathfrak{R} \\ z \in \mathfrak{R} \end{cases}$$

FIGURE 3 – Description du point P dans l'espace de spatialisation

Notre espace, que l'on notera \mathbf{E} est défini comme *vectoriel*, c'est-à-dire qu'il possède une structure permettant d'effectuer une série de combinaisons linéaires de ses éléments. Dans notre cas, cette structure repose sur la théorie des ensembles classiques, à laquelle on adjoindra toutefois la série de principes

définie en introduction.

Le corps \mathbf{K} de base de notre espace est donc l'ensemble des points de l'espace tels que définis dans la *Figure 2*, et c'est donc un corps commutatif assimilé à \mathfrak{R}^3 sur lequel il est possible d'effectuer des additions, soustractions, multiplications et divisions, de manière commutative.

Ces précisions nous seront nécessaires dans les chapitres suivants, lorsqu'il s'agira de définir les opérations sur l'espace sonore, disponibles à l'utilisateur.

Un *pattern* de réflexion

Tout au long des chapitres qui suivent, nous allons construire selon les principes énoncés dans l'introduction une série d'entités fondamentales à notre système d'écriture. Il sera donc nécessaire de définir leur morphologie par une analyse *orientée objet*, c'est à dire par une réflexion sur les données nécessaires au fonctionnement de chaque objet ainsi que des relations à établir entre eux. Le langage de modélisation employé ici sera l'UML ³³.

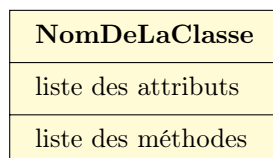


FIGURE 4 – Exemple de modélisation d'une classe en UML

33. Unified Modeling Language, ou langage de modélisation unifié.

Notre espace sonore étant défini, on peut alors en décrire l'élément constituant la base de la représentation géométrique, le point :

Point
<u>- x : float</u>
<u>- y : float</u>
<u>- z : float</u>
+ setX(x : float) : void
+ setY(y : float) : void
+ setZ(x : float) : void
getX() : x
getY() : y
getZ() : z

FIGURE 5 – Modélisation de la classe Point

Dans ce modèle sont définies trois *variables internes*, x y et z qui sont de types entiers flottants, et dont l'accessibilité à l'utilisateur de la classe n'est pas directe. La lecture (respectivement la modification) de ces valeurs se fait via les *méthodes publiques* getX, getY et getZ (respectivement setX, setY et setZ) appelées accesseurs (respectivement mutateurs).

Le principe de "cacher" les données internes d'un objet peut paraître à première vue inutile et fastidieux, mais c'est en fait un très bon moyen pour limiter ou contrôler la modification de certains paramètres afin, par exemple, de ne pas placer un point dont les coordonnées le situeraient en dehors de l'espace utile. Ce principe se nomme l'*encapsulation des données*.

L'espace dans lequel s'inscrit une scène sonore étant défini, nous allons pouvoir construire l'élaboration de notre système. Les premières définitions étant abstraites, Il s'agit maintenant d'y adjoindre les principes de *sémantique* et de *structure* comme définis dans l'introduction à cette réflexion. Les quatre chapitres suivants vont énoncer les fondements de notre système sur lesquels tout le concept de l'écriture de l'espace sonore va reposer.

Chapitre 1

L'entité de base : la source sonore

Afin de créer une description la plus générale possible d'un ensemble d'événements sonores articulés dans le temps et l'espace, il nous faut d'abord chercher sa plus petite entité, son unité minimale de signification. La technique audio prend comme point de départ le concept de source *monophonique*, un simple flux sur un seul canal, en général enregistré par un unique microphone. A partir de là se construisent les formats *stéréophoniques*, ainsi que tous les formats multi-canaux tels que la quadriphonie, le système 5.1, l'octophonie, etc. Dans notre cas, l'entité de base sera donc définie comme source sonore simple, issue d'un flux monophonique. Nous verrons au chapitre suivant qu'il est possible de reconstruire très aisément chaque format multi-canal à partir de ce système de base.

Cette entité est donc la représentation de la position d'un objet sonore dans l'espace. Elle possède ses propres coordonnées spatiales dans un référentiel défini. On fait alors appel au modèle du *Point* défini dans l'introduction pour la description des coordonnées.

Opérations géométriques

Afin de progresser vers la notion de mouvement, sans laquelle notre principe de mise en espace serait réduit à un statisme fort peu imaginaire, il nous faut définir une série d'opérations spatiales applicables à une source sonore. Les transformations nécessaires à la création d'un dynamisme spatial sont la translation, la rotation autour d'un axe et l'homothétie.

Définition 1. Une translation de vecteur \vec{u} est une transformation qui, à tout point $M \in E$, associe le point $M' \in E$ tel que :

$$\overrightarrow{MM'} = \vec{u}$$

Définition 2. Une rotation de centre $C \in E$ et d'angle θ est une transformation qui laisse le point C invariant et qui transforme tout point $M \in E$ autre que C en un point $M' \in E$ tel que :

$$\begin{aligned} |\overrightarrow{CM}| &= |\overrightarrow{CM'}| \\ (\overrightarrow{CM}; \overrightarrow{CM'}) &= \theta \end{aligned}$$

Définition 3. Une homothétie de centre $C \in E$ et de rapport $k \in \mathfrak{R}$ est une transformation f qui laisse le point C invariant et qui transforme tout point $M \in E$ autre que C en un point $M' \in E$ tel que :

$$M' = f(M)$$

Les points C , M et M' sont alignés

$$|\overrightarrow{CM'}| = k \cdot |\overrightarrow{CM}|$$

Modélisation

On peut maintenant modéliser la classe *Atom*. On ajoutera un numéro d'identifiant de la source, un nombre entier fixé et assigné à chaque entité créée, issue de cette classe.

Atom
- <u>id</u> : int - <u>point</u> : Point
+ setPoint(point : Point) : void # <i>getPoint()</i> : point

FIGURE 1.1 – Modélisation de la classe *Atom*

Voilà l'unité minimale de notre principe définie. Le chapitre suivant sera consacré à l'étude des groupement de sources, ou atomes, premier pas vers une mise en interaction des constituants de la scène sonore.

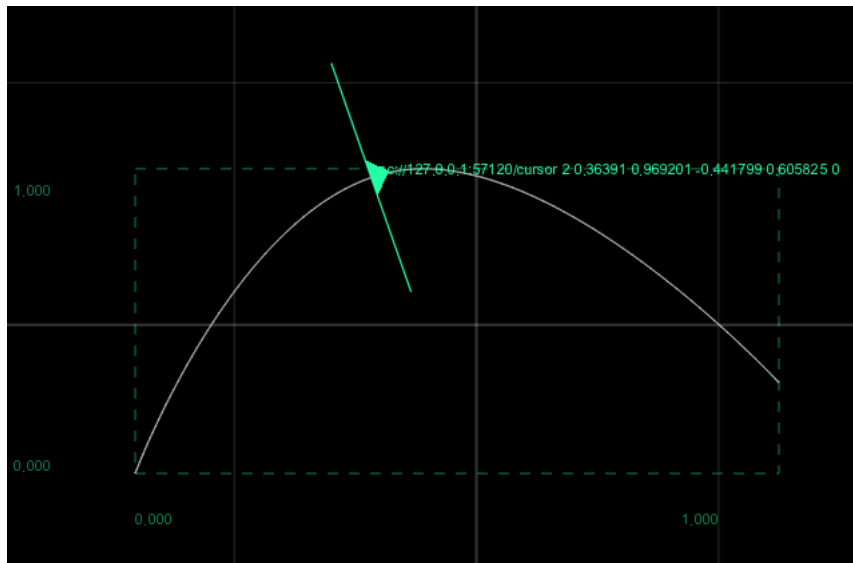


FIGURE 1.2 – Une description de l'espace dans le logiciel *IanniX*

Chapitre 2

La notion de groupe

« Une forme est autre chose ou quelque chose de plus que la somme de ses parties. »¹

Paul Guillaume

Dans l'introduction à ce travail, nous avons énoncé le besoin de faire des opérations sur un groupement de sources pour faciliter la gestion d'un nombre d'entités sonores toujours croissant. Nous avons d'ailleurs établi le **principe 5** qui définit une relation d'ensemble envisageable aux différents niveaux de notre système. Outre l'idée de simplification d'un pilotage, il est d'autres raisons pour lesquelles la notion de groupe est absolument nécessaire et même fondatrice à notre principe de mise en espace.

– Le principe d'*atome* a été posé comme étant l'entité de base de la description d'une scène, et aussi comme point d'entrée d'un flux monophonique à notre système. La notion de groupe devra donc permettre la description d'une source polyphonique, en tenant compte de la position et de l'espacement de ses canaux dans l'espace.

1. Citation de Paul Guillaume in *La Psychologie de la forme*, Flammarion 1937.

- L'idée même d'une mise en espace *stochastique* des objets sonores, via une série de paradigmes, repose sur la création de grands d'ensembles d'atomes sonores définis.

Une description mathématique

Un ensemble est une série d'éléments de même nature, ordonnée ou non. Dans notre cas, l'élément de base est l'atome, défini sur le corps \mathbf{K} , et la définition de chaque ensemble — fini — repose sur la liste de sources que l'utilisateur souhaite grouper.

Soient i atomes $a_i \in K$. On peut définir l'ensemble E ainsi :

$$E = \{a_0, a_{20}, a_3, a_{10}, a_7\}$$

Bien que pouvant apparaître comme tautologique, cette description *extensive* possède néanmoins quelques propriétés intéressantes :

- Il n'y a pas de notion d'ordre dans notre ensemble, la position de l'atome dans la série n'a pas d'importance.
- Rien n'interdit à un même atome d'appartenir à deux groupes. Bien que cette conséquence puisse sembler de prime abord préjudiciable à la compréhension ainsi qu'à une bonne utilisation de notre système, l'appartenance simultanée à différents groupes peut favoriser la création d'effets intéressants en termes de mise en espace, comme nous le verrons dans les prochains chapitres.
- En revanche, la répétition d'un même élément au sein d'un groupe ne modifie pas l'ensemble. Il n'y a ici aucune nécessité à la mise en place de *multi-ensembles*. On en déduit alors la définition suivante :

Soient deux ensembles E et E' définis ainsi :

$$E = \{a_0, a_{20}, a_3\}$$

$$E' = \{a_0, a_3, a_{20}, a_3\}$$

Alors

$$E = E'$$

L'idée d'un mouvement coordonné

Maintenant que la liste des entités sonores appartenant à un groupe est définie, il nous faut proposer un premier moyen de pilotage coordonné des sources afin de pouvoir déplacer facilement l'ensemble des atomes, mais aussi d'effectuer tout aussi aisément n'importe quelle transformation géométrique.

Dans le domaine de la conception assistée par ordinateur (C.A.O) existe le concept de *point d'ancrage*, point qui sert de référent à tout type de transformation d'objet, notamment pour les rotations, où il sert de point pivot.

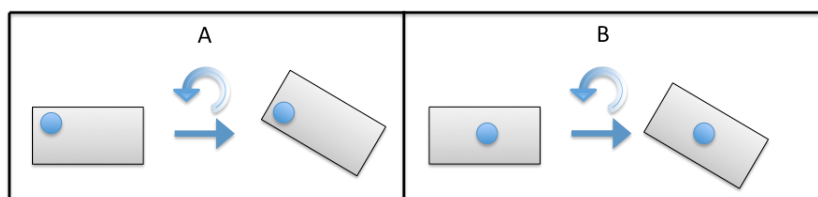


FIGURE 2.1 – Principe du point d'ancrage en C.A.O

Dans la figure 2.1, on peut voir comment le positionnement du point d'ancrage affecte le comportement de la rotation du rectangle. Dans le cas d'une rotation ou d'une homothétie, le point d'ancrage du groupe fait office d'*invariant pour la transformation*.

On est donc amené à ajouter au concept de groupe un point d'ancrage, dont le comportement est celui d'un point classique de l'espace E tel que défini page 21 dans la figure 3. De plus, les définitions 1, 2 et 3, page 25 vont servir de base géométrique à la gestion des atomes d'un groupe.

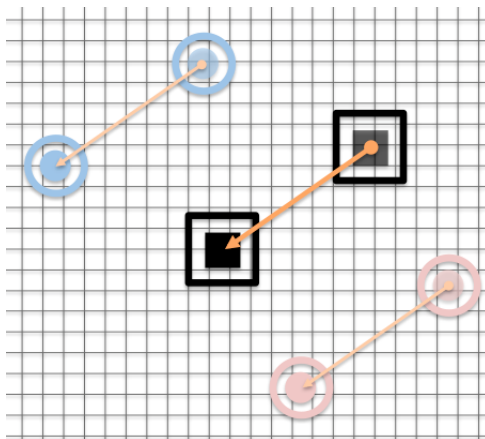


FIGURE 2.2 – Simple translation des atomes d'un groupe

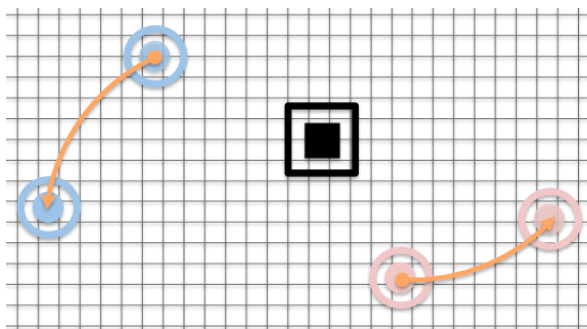


FIGURE 2.3 – Rotation et homothétie combinées des atomes d'un groupe

Modélisation

On peut maintenant modéliser la classe *Group*. De la même manière que pour la classe *Atom* définie dans le chapitre précédent, on associe à chaque entité issue de la classe *Group* un identifiant unique.

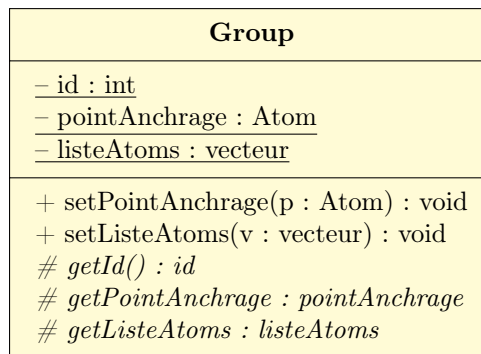


FIGURE 2.4 – Modélisation de la classe *Group*

La relation de dépendance des classes *Atom* et *Group* est importante. En effet, un groupe ne possède pas les sources, au sens où elles ne sont pas créées par lui. Il ne s'agit pas d'une relation d'appartenance, mais bien d'une association : l'attribut *listeAtoms* est une simple liste de références, chacune pointant vers la position de la source associée.

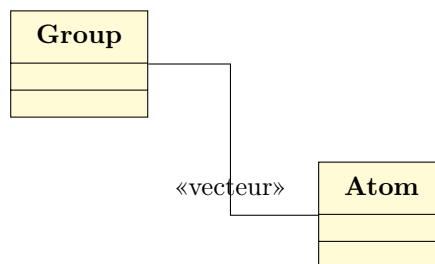


FIGURE 2.5 – L'interaction des classes *Atom* et *Group*

La notion de groupe nous permet une gestion simplifiée d'un grand

nombre de sources. Elle ouvre en outre la possibilité d'une orchestration des différentes parties. Ainsi, de la même manière que le système d'écriture traditionnel de la musique a contribué à la définition de l'orchestration telle qu'elle existe depuis le XIX^e siècle grâce à une notation verticale forte, on peut très bien imaginer d'organiser nos atomes et groupes selon une disposition proche de celle de la notation musicale traditionnelle.

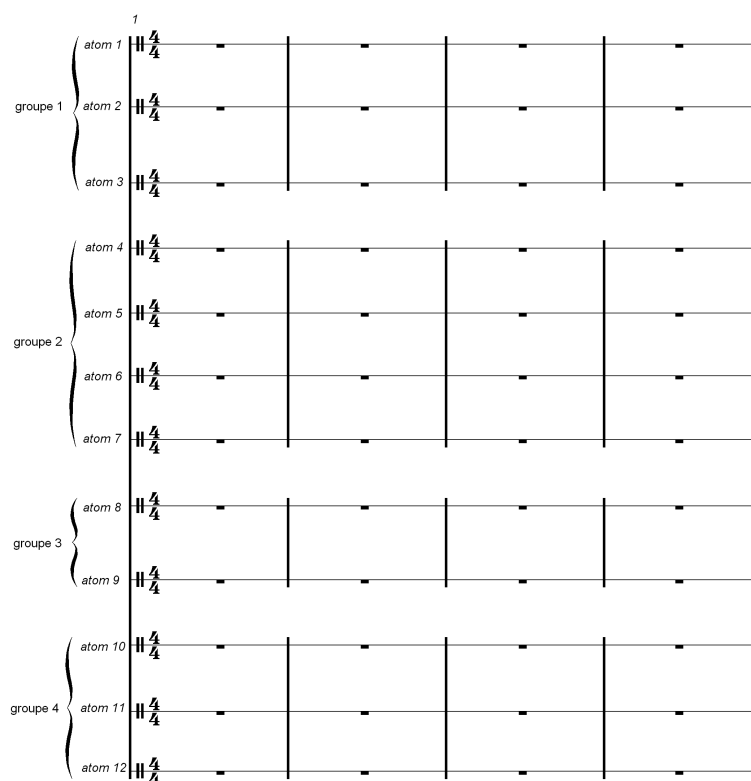


FIGURE 2.6 – Exemple de structuration des sources selon une écriture musicale standard

L'orchestration étant, entre autres, une combinaison d'effets et de rapports de poids entre différentes sources musicales dans le but de créer une série globale — macroscopique — d'effets, c'est ici que le principe de **Paradigme** défini en introduction prend tout son sens : c'est lui qui

va nous permettre d'avoir une vision et une prise en main plus globale, plus efficace, pour la création d'effets combinés de mise en espace de différentes sources. C'est l'objet du chapitre suivant.

Chapitre 3

Paradigmes

Les modèles comportementaux

Dans les domaines de l'animation graphique, il existe une branche appelée « Behavioral Animation » qui pourrait être traduite par « animation comportementale ». Il s'agit en fait de laisser une entité autonome décider de ses propres actions en fonction d'un contexte donné. Bien sûr, cette forme de « libre arbitre » est très limitée mais elle permet d'apporter à l'animation une touche d'aléatoire, d'improvisé, tout en libérant l'animateur de la contrainte d'avoir à mettre toutes les actions en oeuvre. Le principe récurrent de tous les modèles comportementaux est de mettre à disposition d'une entité l'accès aux paramètres globaux d'une scène, mais uniquement dans son voisinage direct. Le modèle physique sera chargé de calculer la position image par image de chaque élément. La plus-value en termes de réalisme est significative : les mouvements paraissent plus naturels et une part de pseudo-aléatoire permet une fluctuation autour d'une valeur initiale définie.

Ce principe d'animation comportementale n'est rien d'autre en fait que la modélisation d'un ou plusieurs phénomènes physiques telle qu'elle

est pratiquée depuis très longtemps dans les domaines de la recherche en physique appliquée ou encore dans l'industrie automobile. La seule différence est le champ d'application qui, dans le cas de l'image de synthèse, fait de l'animateur le seul juge et référent du résultat. Il ne s'agit en aucune façon de proposer un modèle sur lequel l'utilisateur ne pourrait avoir la main *in fine*, ni un modèle qui le forcerait à adapter son contenu à l'algorithme. Il est donc nécessaire de construire des modèles simples en termes d'intégration à un milieu existant, et qui ne forcent pas le créateur à adapter sa démarche artistique à une contrainte technique.

Microscopie & Macroscopie

En parcourant la majorité des modèles comportementaux déjà créés pour l'industrie de l'image de synthèse, on peut très vite constater leur fort degré de ressemblance en ce que tous proposent de simuler le cycle de vie de petites entités très nombreuses au sein d'un milieu dont l'ordre de grandeur est nettement supérieur. Tous présentent deux niveaux d'échelle, ou tout du moins définissent un cadre d'évolution des éléments qui est toujours fini et de taille nettement supérieure à celle des entités. Sont donc associées deux dimensions, la première plutôt microscopique, celle des éléments sur lesquels on souhaite appliquer un modèle de comportement, et la seconde à l'échelle macroscopique renvoyant au milieu dans lequel ils évoluent. A priori, les paramètres microscopiques sont ceux que l'on souhaite automatiser (dans le sens où l'on ne souhaite pas avoir à les contrôler et qu'ils relèvent donc du modèle physique) tandis que les grandeurs macroscopiques sont laissées au choix de l'utilisateur.

L'enjeu principal d'un modèle comportemental est donc de permettre de définir un niveau d'évolution microscopique grâce à une série de gran-

deurs macroscopiques. Et c'est en cela qu'il semble intéressant dans notre cas. En effet si on se réfère au **Principe 4** défini en introduction et relatif aux échelles de manipulation des données, un tel modèle comportemental devient un formidable intermédiaire de travail, une puissante interface entre création artistique et réalisation.

Nous allons maintenant présenter trois modèles, très répandus, que sont le *suivi d'une courbe*, la *cinétique de gaz* et les *boids*.

3.1 Le suivi d'une courbe

Ce premier modèle, le plus simple, propose de déplacer un groupe de sources de manière coordonnée, en agissant sur le point d'ancrage de l'ensemble. Bien qu'élémentaire, ce système reste néanmoins la base de la mise en espace — qu'il soit appliqué à une seule source ou à plusieurs — et va servir aux autres paradigmes, comme nous allons le montrer dans les parties suivantes.

L'introduction d'une courbe nécessite d'en expliciter quelque peu la description, bien qu'au stade actuel de notre travail, il ne soit pas nécessaire d'en approfondir les détails. Mathématiquement, une courbe désigne un sous-ensemble de l'espace, E dans notre cas, et cette dernière est décrite par un système d'équation qui lie chaque composante spatiale à un paramètre unique.

Définition 4. Soient trois fonctions f , g et h au moins définies sur C^0 .

C est le lieu des points $M(x,y,z) \in E$ tel que :

$$C \equiv \begin{cases} x = f \\ y = g \\ z = h \end{cases}$$

De plus, chaque point $M(t)$ est défini comme :

$$M(t) = \begin{cases} x = f(t) \\ y = g(t) \\ z = h(t) \end{cases}, t \in \mathfrak{R}$$

Ainsi, avec une définition aussi générale, il est possible d'envisager que le support de la trajectoire soit un ensemble de lignes brisées, une courbe de Bézier, un cercle etc.

Modélisation

La classe *Courbe* est une classe mère abstraite, c'est à dire que son implémentation n'est pas complète et n'est donc pas utilisable en l'état. Il est alors nécessaire de réaliser une série de classes filles, dans le but de définir si la courbe est une ligne, une courbe de Bézier, etc.

Afin de pouvoir effectuer les opérations géométriques de base sur les courbes, on ajoute le principe du point d'ancrage à la classe *Courbe*, comme on l'a fait pour la classe *Group*. Enfin, cette classe doit permettre de renvoyer à l'utilisateur la position d'un point $M(x,y,z)$, ainsi que la tangente à la courbe, en fonction d'un paramètre t donné.

Courbe
<u>- pointAnchrage : Atom</u> <u>- listePoints : vecteur</u>
+ setPointAnchrage(p : Atom) : void + setListePoints(v : vecteur) : void # <i>getPointAnchrage</i> : <i>pointAnchrage</i> # <i>getListePoints</i> : <i>listePoints</i> # <i>getPositionForT</i> (t : float) : <i>Point</i> # <i>getTangenteForT</i> (t : float) : <i>Float</i>

FIGURE 3.1 – Modélisation minimale de la classe *Courbe*

Application à notre système

Dans notre cas, l'utilisateur va créer un chemin sous la forme d'une courbe quelconque, puis il lui assignera un groupe dont les atomes effectueront la trajectoire en fonction de leur position relative au point d'ancrage.

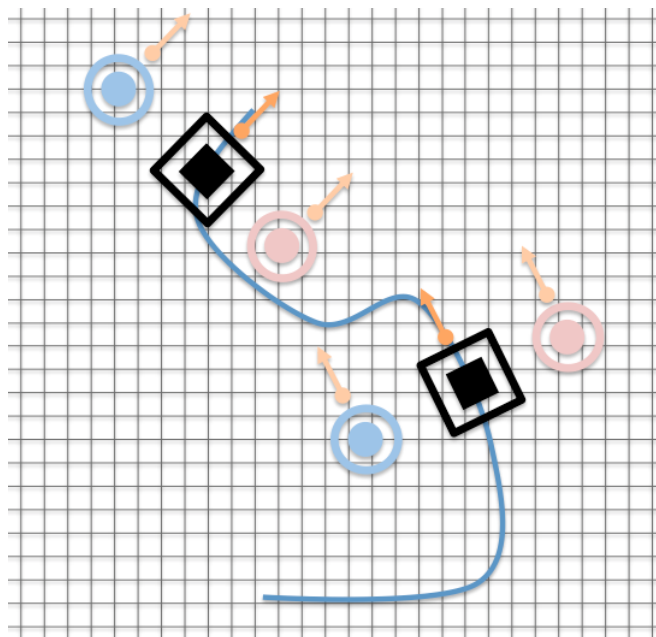


FIGURE 3.2 – Représentation du suivi d'une courbe par un groupe

En revanche, s'il souhaite ne piloter qu'une seule source, l'utilisateur a la possibilité de n'assigner qu'un seul atome au groupe et de le placer sur le point d'ancrage. Cette méthode peut paraître pour l'instant un peu lourde ; nous verrons dans la partie relative aux interfaces comment en faciliter la gestion.

3.2 La cinétique des gaz ¹

Développée depuis le XVIII^e siècle sur les travaux de Bernoulli, Clausius, Waterston, Krönig, Maxwell et Boltzmann, cette théorie permet de rendre compte du comportement macroscopique d'un gaz à partir des caractéristiques des corpuscules qu'il contient. Un gaz est constitué de molécules mono ou poly-atomiques qui subissent des interactions de natures magnétiques, mécaniques mais aussi quantiques dans des cas de températures et de pressions particulières. Le modèle des gaz parfaits permet de simplifier quelque peu les choses en proposant les hypothèses suivantes :

- Le gaz est à faible pression, dans un volume V fini. Il y a donc peu de molécules, on peut donc négliger l'énergie d'interaction :

$$E_{tot} = \sum_{i=1}^N E_i$$

- La densité moléculaire est uniforme :

$$n = \frac{\Delta N}{\Delta V} = \text{constante}$$

- La distribution des vitesses est homogène et isotrope et elle suit le modèle de mouvement brownien.

1. Cette partie s'inspire du cours *Phénomènes de transport et Mécanique des fluides* dispensé par Marc Levallois en licence de physique à l'université de Caen-Basse Normandie, 2008-2009.

- Les molécules sont assimilées à des sphères dures dont le diamètre est négligeable devant la distance moyenne qui les sépare.

On démontre facilement, par l'étude des chocs contre les parois de l'enceinte, que la température cinétique macroscopique dans le volume est alors :

$$T_c = \frac{mv^2}{3k_B}$$

où m désigne la masse d'une molécule, v la vitesse moyenne et k_B la constante de Boltzmann.

En d'autres termes, l'effet macroscopique de température dépend uniquement de la nature des chocs des molécules contre la paroi du conteneur, la quantité de matière n'étant pas prise en compte. Seules la vitesse moyenne, l'inertie des molécules ainsi que la géométrie de l'enceinte prévalent dans ce modèle simple.

Il est maintenant possible de modéliser le comportement individuel de chaque molécule en entrant les données de masse et de vitesse moyenne, c'est-à-dire des données inhérentes à la nature de la molécule mise en jeu, ainsi qu'aux données macroscopiques que sont la taille et la géométrie du volume conteneur.

Application à notre système

L'emploi d'un tel paradigme nécessite la description d'un contour géométrique qui délimite une portion close de l'espace sonore. Dans l'exemple précédent, le suivi d'un chemin, nous avons justement défini l'objet *courbe*. Celui-ci va donc nous servir de frontière pour l'enceinte dans laquelle évoluent les particules. Cependant, il est nécessaire d'ajouter

une contrainte de "fermeture" de la courbe : dans le cas où le point de départ de la courbe n'est pas confondu avec le point d'arrivée, on relie automatiquement ces deux points afin d'en clore le lieu.

Les paramètres de masse et de vitesse moyenne sont laissés à l'utilisateur sous une seule variable que l'on pourrait assimiler à l'inertie : en effet, la notion de poids d'un atome n'a ici que peu de sens, on préférera donc en fixer une valeur unitaire et ne faire jouer qu'un facteur de vitesse.

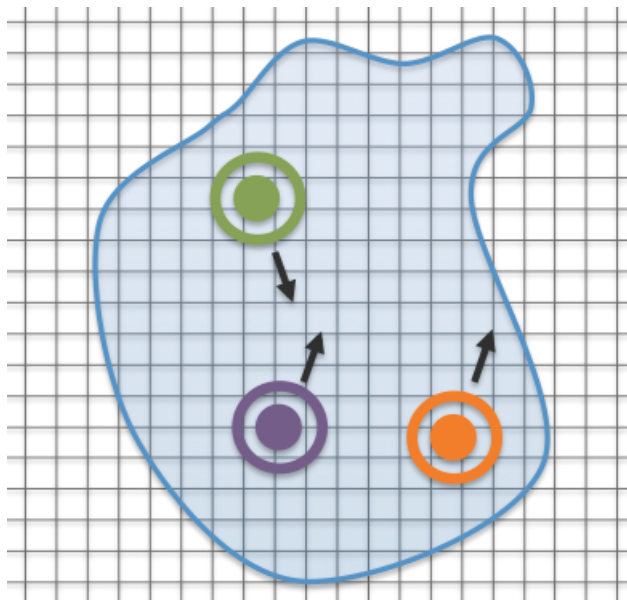


FIGURE 3.3 – Représentation d'un mouvement aléatoire au sein d'une enceinte close

3.3 Le modèle des *boids* - ou "essaim particulaire"

En 1986, Craig W. Reynolds crée un programme de simulation du comportement d'une nuée d'oiseaux en vol. L'idée de base de cet algorithme est de pouvoir passer d'une représentation de groupe — le mouvement fluide et compact d'une nuée d'oiseaux — à la description détaillée des mouvements de chaque individu.



FIGURE 3.4 – Une implémentation des *boids* dans le logiciel *PureData*

Le principe de base de ce programme est de donner à chaque entité une sorte d'intelligence propre qui régit un comportement individuel en fonction du reste de la scène. Ainsi, dans son exemple, Reynolds explique que le comportement de base de chaque oiseau dans une nuée est "à la fois une volonté de rester proche de la nuée tout en évitant les collisions avec ses congénères"². En résulte un algorithme dans lequel est simulée la manière dont est perçue par une entité la scène globale, tout en tenant compte des paramètres de distance, d'angle de vue, mais aussi du milieu où se situe la scène :

2. *Flocks, Herds, and Schools : A Distributed Behavioral Model*, article de Craig W. Reynolds in *Computer Graphics* n°21, 1987. Traduit par nos soins.

« Dans un banc de poissons, la perception visuelle du voisinage immédiat d'un poisson est réduite par l'absorption et la diffusion de la lumière causée par l'eau parfois boueuse. » ³

Il est maintenant nécessaire de définir un *but* précis à la nuée d'oiseaux — ou de poissons —, ce que Reynolds nomme "désir de migrer" : ce modèle comportemental n'est utile que si l'animateur peut en contrôler la position. En résulte un paramètre dynamique, une localisation spatiale vers laquelle se dirige le groupe, mais qui n'est accessible à chaque entité que de proche en proche, créant un effet de "suivi" dans lequel chaque entité suit son voisin le plus proche.

Application à notre système

Si l'on reprend la notion de groupe telle que nous l'avons définie dans le chapitre précédent, le point d'ancrage permet d'effectuer une série coordonnée de transformations sur chaque atome constituant le groupe. Si l'on considère que le point de convergence de chaque entité est confondu avec le point d'ancrage d'un groupe, l'algorithme des *boids* va permettre d'apporter un certain dynamisme au mouvement, le tout de manière très simple. Ainsi, si l'utilisateur déplace le point d'ancrage du groupe, les atomes suivront ce point comme le "désir de migration" défini par Reynolds, induisant de ce fait un mouvement conjoint, dont la description et la mise en place seront grandement facilitées.

De la même manière que pour la cinétique des gaz, les paramètres définissant le comportement global des entités (angle de vue, vision de la scène en fonction de la distance, etc.) sont dévolus à l'utilisateur, et dans le même souci de concision, sont regroupés autour d'une variable

3. *Op. cit.*

globale qui mesure la réactivité et la capacité à suivre le point d'ancrage :
l'inertie en quelque sorte.

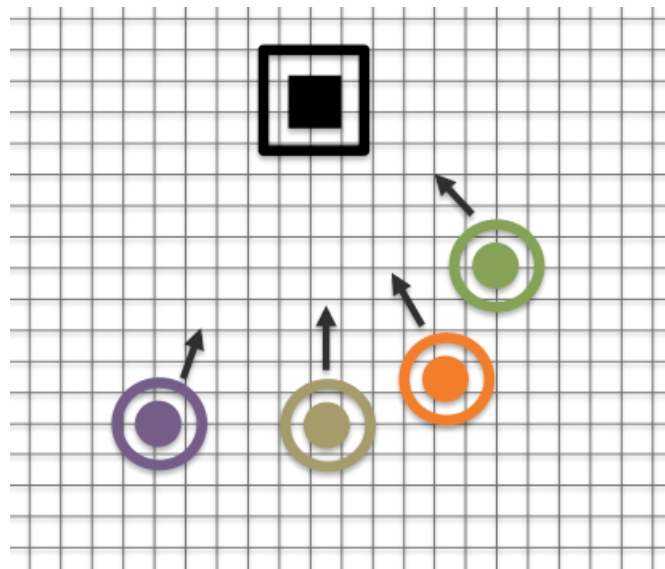


FIGURE 3.5 – Représentation du suivi d'un point par un essaim d'atomes

Enfin, il est possible d'adjoindre au point d'Ancrage une trajectoire,
de la même manière que dans le paradigme du suivi d'une courbe.

3.4 Vers la synthèse d'un modèle

Si l'on souhaite dégager une série de règles pour la généralisation du concept de paradigme de pilotage des atomes dans l'espace sonore, on peut tenter une synthèse des trois exemples présentés précédemment. S'en suit une liste de caractéristiques globales déduites des modèles cités précédemment que l'on peut interroger :

- Point d'ancrage : le paradigme emploie-t-il le point d'ancrage pour déplacer les atomes ?
- Courbe : Le paradigme emploie-t-il une courbe ? Si oui, quelle est sa fonction ?
- Espace fini : Le paradigme emploie-t-il une portion finie de l'espace ?
- Vitesse d'avancement : Existe-t-il une vitesse d'avancement qui soit proportionnelle à la durée du paradigme ?
- Paramètre secondaire : Existe-t-il un paramètre qui n'est pas être assimilé à une caractéristique des notions d'atomes ou de groupes ?

	Cinétique des gaz	Boids	Suivi de trajectoire
Point d'ancrage		×	×
Courbe	Contour	Trajectoire	Trajectoire
Espace fini	×	×	×
Vitesse d'avancement		×	×
Paramètre secondaire	×	×	

FIGURE 3.6 – Tableau récapitulatif des caractéristiques propres à chaque paradigme

On peut d'abord remarquer la similitude entre le paradigme de suivi d'une trajectoire et celui des boids : la seule différence est l'existence du paramètre secondaire relatif à la capacité de chaque atome à suivre le point d'ancrage. Finalement, c'est la manière dont le lien entre chaque

atome et son point d'ancrage est défini qui change : dans le cas d'un suivi de trajectoire, le lien est fixe, solide, tandis que dans le cas des boids, le lien est dynamique et défini selon un algorithme. On introduit alors le concept de *lien* entre chaque atome et le point d'ancrage comme un ensemble d'équations liant la position relative de chaque entité au sein du groupe à la position absolue du point d'ancrage, à un éventuel paramètre secondaire ainsi qu'à une variable d'avancement global.

Définition 5. *Soit un groupe G_1 contenant un ensemble E d'atomes noté $E = \{A_i\}, i \in \mathbb{N}$. Le point d'ancrage du groupe G_1 est désigné par P_1 . Pour un paradigme quelconque, on définit le lien entre P_1 et chaque atome A_1 comme une fonction vectorielle \vec{f}_i qui, pour un triplet de valeurs donné de P_1 , d'un paramètre secondaire ν et d'une variable d'avancement globale t renvoie la position que doit occuper chaque atome par rapport au point d'ancrage :*

$$\overrightarrow{A_i P_1} = \vec{f}_i(P_1, \nu, t)$$

Cas particulier : si $\vec{f}_i = \overrightarrow{\text{constanté}}$, on retrouve la description statique de la position des atomes par rapport au point d'ancrage telle que définie dans le chapitre 2 sur la notion de groupe.

Vers la combinaison de différents paradigmes

De la même manière que les transformations de l'espace E peuvent être combinées dans le but de créer des mouvements complexes, il est nécessaire de définir un moyen de construire un pilotage plus complet des atomes. Pour ce faire, il nous faut revenir quelque peu sur la définition d'un groupe. Dans le chapitre précédent, nous avons défini le point d'ancrage comme étant un atome. Il est donc possible de considérer ce point

comme étant aussi susceptible d'être piloté par un paradigme. Prenons un exemple : soit un paradigme P_1 pilotant les sources d'un groupe G_1 de manière aléatoire au sein d'un contour que l'on souhaite déplacer dans l'espace. On crée alors un second paradigme P_2 de suivi de trajectoire associé à un groupe G_2 contenant le point d'ancrage du groupe G_1 .

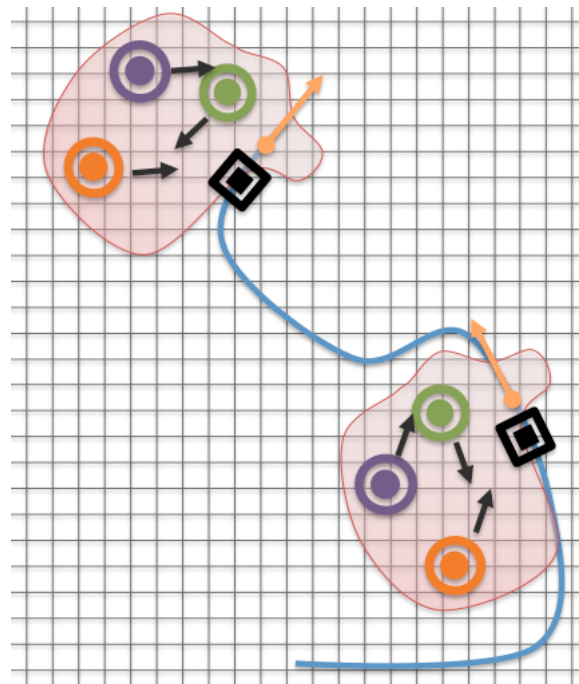


FIGURE 3.7 – Combinaison d'un mouvement aléatoire avec un déplacement global selon une trajectoire

Cette fonctionnalité impose cependant une précaution : un groupe ne peut pas posséder son propre point d'ancrage car il en résulterait, le cas échéant, une récursion fatale à tout système de calcul. Nous verrons donc dans la seconde partie relative à l'interface utilisateur comment empêcher une telle situation de se produire.

Modélisation

Si l'on récapitule ce qui a été énoncé depuis le début de ce chapitre, un *Paradigme* est composé :

- D'un groupe G composé d'un ensemble E d'atomes et d'un point d'ancrage P , sur lequel il applique une transformation complexe.
- D'un objet *Courbe*, qui peut indifféremment servir de chemin ou de contour.
- D'un ensemble de fonctions \vec{f}_i définissant la manière dont les atomes sont pilotés au sein du groupe. Pour l'instant, on regroupera cet attribut sous une forme opaque, en définissant un type de donnée *ParadigmType* pouvant prendre les valeurs "trajectoire", "aléatoire" ou "bois".
- D'un paramètre secondaire (facultatif).

On peut donc maintenant modéliser la classe *Paradigm*.

Paradigm
<ul style="list-style-type: none">- <u>id</u> : int- <u>param</u> : float- <u>listeAtoms</u> : vecteur- <u>groupe</u> : Group- <u>type</u> : paradigmType
<ul style="list-style-type: none">+ setPointAnchrage(p : Atom) : void+ setListeAtoms(v : vecteur) : void+ setParam(p : float) : void+ setGroup(groupe : Group) : void+ setType(type : paradigmType) : void# getId() : id# getListeAtoms : listeAtoms# getParam : param# getGroup : groupe# getType : type

FIGURE 3.8 – Modélisation de la classe *Paradigm*

Récapitulatif du système

Avant d'entamer le chapitre suivant relatif à la question du temps, on peut faire un bilan d'étape de l'élaboration de notre système. Dans un premier temps, nous avons décrit l'espace sonore E le plus général possible, à trois dimensions, dans lequel le système de coordonnées ainsi que le repère est fixé. Dans cet espace, une série de transformations spatiales de base a été définie : la translation, la rotation ainsi que l'homothétie. Puis la notion d' *Atome* a été décrite comme représentation d'une source sonore ponctuelle et sur laquelle il est possible d'effectuer les transformations de l'espace E . Afin d'adapter le système à des situations plus complexes, nous avons introduit le concept de *Groupe*, un ensemble d'atomes possédant un *point d'ancrage*, permettant la gestion et la transformation simultanée de plusieurs entités sonores.

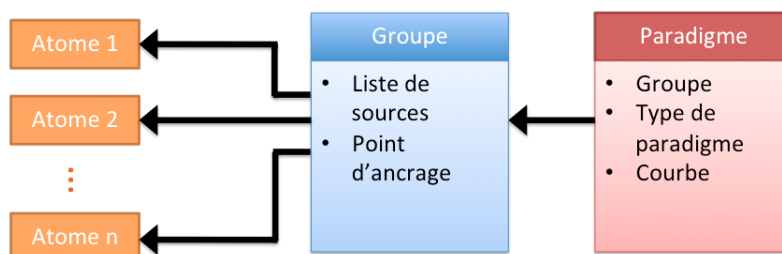


FIGURE 3.9 – Synoptique intermédiaire du pilotage de plusieurs atomes

A ce point de la description du système, nous avons introduit le concept de *Paradigme* comme intermédiaire de création entre une intention de mise en espace d'une source sonore, c'est-à-dire la création d'une dynamique dans la scène *via* des mouvements simples à mettre en œuvre. Les trois paradigmes pris en exemple ont alors montré une convergence vers un modèle unique. Afin de mettre en place ce principe

de pilotage des atomes, il nous a fallu aussi définir les objets mathématiques de *Courbe* et de *Lien* qui ont montré l'émergence d'une nouvelle variable d'avancement, d'évolution : en d'autres termes le *Temps*. Nous verrons dans le chapitre suivant de quelle manière nous pouvons l'intégrer à notre système d'écriture de l'espace sonore.

Chapitre 4

La question de la gestion du temps

La description faite jusqu'alors de notre système est entièrement statique. Hormis l'idée d'évolution temporelle introduite par le concept du *paradigme* dans le chapitre précédent et qui relève d'un temps à l'échelle "microscopique", aucune de nos descriptions ne prend explicitement en compte cette dimension.

Dès lors, ce qui manque encore à notre principe, c'est une articulation temporelle entre différents états. Il nous faut donc définir les concepts de transition, trajectoire, en un mot tout ce qui est nécessaire à faire vivre une scène sonore dans sa temporalité. Ainsi introduit-on le temps comme paramètre absolu de déroulement et de pilotage des différents éléments de notre système. *In fine*, chaque entité du système sera reliée au temps, ce qui apparaîtra de manière explicite ou non à l'utilisateur.

$$t \in \mathbb{R}^+$$

Si l'on prend l'exemple d'une scène sonore dans laquelle évoluent quatorze atomes et pour lesquels sont définis quatre paradigmes, on

peut articuler la scène dans sa dimension temporelle selon l'exemple suivant :

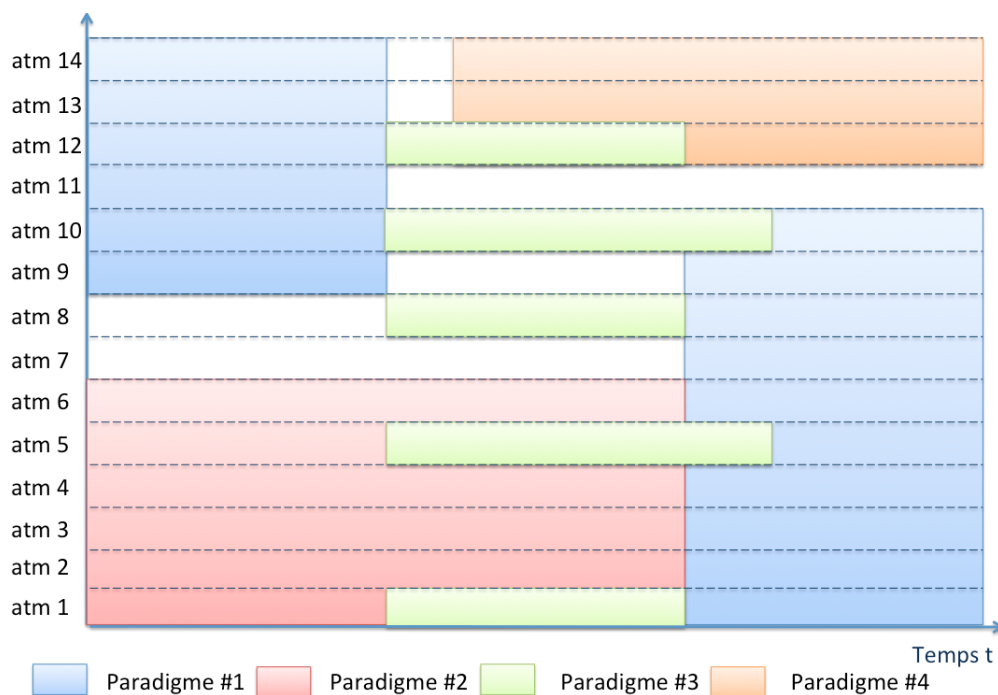


FIGURE 4.1 – Exemple d'articulation de différents paradigmes

Dans cet exemple, on peut tout d'abord constater que l'affectation des atomes aux paradigmes n'est pas statique au cours du temps. Si l'on prend l'exemple de l'atome n°10, il est d'abord contrôlé par le paradigme n°1, puis par le n°3 et enfin de nouveau par le paradigme n°1. De la même manière, l'atome n°14 est initialement piloté par le paradigme n°1 puis, après une période pendant laquelle il n'est affecté à aucun pilotage, on le retrouve associé au paradigme n°4. La question des "temps morts" se pose maintenant : Que faire des atomes lorsqu'ils ne sont affectés à aucun groupe? La réponse est à trouver dans la conception même du système. Comme nous l'avons défini dans la modélisation de la classe Group, figure 2.4 (page 31) les atome existent indépendamment des

concepts de groupe ou de paradigme. Dès lors qu'un atome est assigné à un groupe, ce dernier modifie les coordonnées internes de l'atome (*via* la méthode *setPoint*). Ainsi, dès qu'un atome n'est plus assigné, il reste à sa dernière position enregistrée.

L'exemple de la figure 4.1 rend précisément compte de l'existence limitée des paradigmes au cours du temps : chaque segment de couleur possède une coordonnée temporelle de départ t_i et d'arrivée t_f

Introduction du paramètre t dans le système

Nous allons donc maintenant reprendre brièvement la définition de chaque élément précédemment introduit, pour le paramétrer, si nécessaire, en fonction de t .

Premièrement, un point P de l'espace E ne possède pas de dépendance intrinsèque au temps car il ne s'agit que d'un intermédiaire mathématique à la construction de notre principe. Sa définition proposée en figure 3 page 21 ne change donc pas.

$$P \equiv \begin{cases} x \in \mathfrak{R} \\ y \in \mathfrak{R} \\ z \in \mathfrak{R} \end{cases}$$

FIGURE 4.2 – Description du point P dans l'espace de spatialisation

De la même manière, aucun atome A ne possède pas de "mémoire" propre des trajectoires que l'on souhaite lui appliquer. C'est justement l'intérêt de notre principe que de laisser ce pilotage à une entité intermédiaire, le *Paradigme*. Un *Atome* n'a donc pas davantage de dépendance explicite au temps.

L'objet *Groupe* est un intermédiaire de manipulation des atomes. A aucun moment n'a été introduite dans sa définition de relation de

dépendance formelle au temps. Ainsi son extension ne change-t-elle pas non plus.

C'est donc bien le concept de *Paradigme* qui apporte la dimension temporelle au système. Dans le chapitre précédent, nous avons introduit les notions de *Courbe* et de *lien*, toutes deux incluant un paramètre t visant à faire évoluer au sein d'un groupe, d'une part la position **absolue** du point d'ancrage et d'autre part la position **relative** des atomes d'un groupe par rapport à ce dernier.

La question de la continuité

Les **principe 3** et **4** proposés dans l'introduction impliquent les notions de variabilité entre différentes échelles de représentation, et ce pour n'importe quel paramètre musical. Cela doit donc être appliqué à la gestion du temps dans notre système, tout particulièrement en ce qui concerne sa variabilité, dans le cas d'une utilisation temps-réel. Il est donc nécessaire d'assurer une *interpolation* entre les valeurs successives que pourront prendre les données internes de chaque paradigme. En effet, si l'on prend l'exemple d'un atome suivant un chemin défini par une courbe, sa vitesse de parcours doit pouvoir varier, et cela n'est possible que si le modèle mathématique sous-jacent assure une certaine continuité. C'est pour cela que nous avons imposé une continuité au moins C^0 dans la définition 4 page 37, en introduisant l'objet *Courbe*.

La figure 4.3 montre les différences entre la définition *discrète* d'une courbe et la réalisation de deux interpolations spatiales, la première linéaire, la seconde polynomiale. Le premier cas n'est pas pertinent dans notre système car il ne permet pas un déplacement continu entre deux points, tandis que les deux modes d'interpolation sont tous deux C^0 et

correspondent donc aux critères définis dans la définition 4.

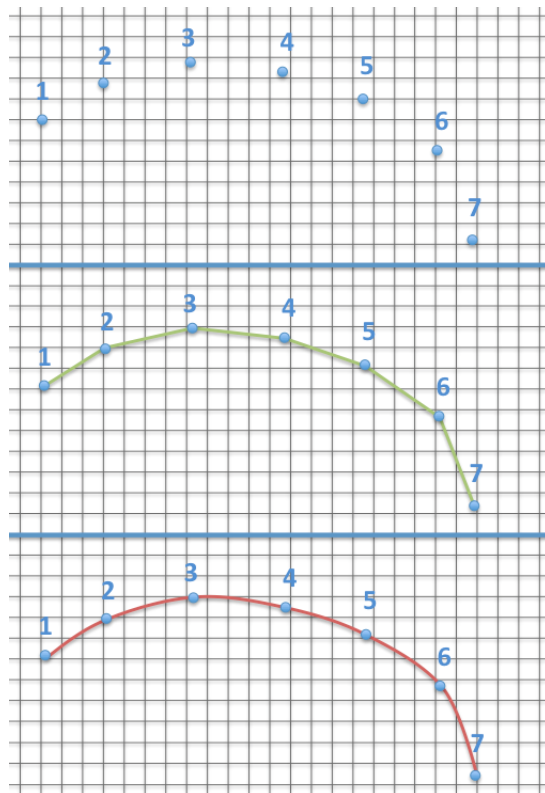


FIGURE 4.3 – Exemples d'interpolation linéaire et polynomiale

note : Le processus d'interpolation des courbes est explicité, annexe B, à la fin de ce mémoire.

Avant de continuer

Avec l'intégration de la variable temps, le système de base de notre principe est maintenant exhaustif. On peut ainsi compléter le synoptique 3.9 page 50 pour y faire apparaître toutes les entités constitutantes, et notamment l'interface proposée à l'utilisateur, objet de la deuxième partie.

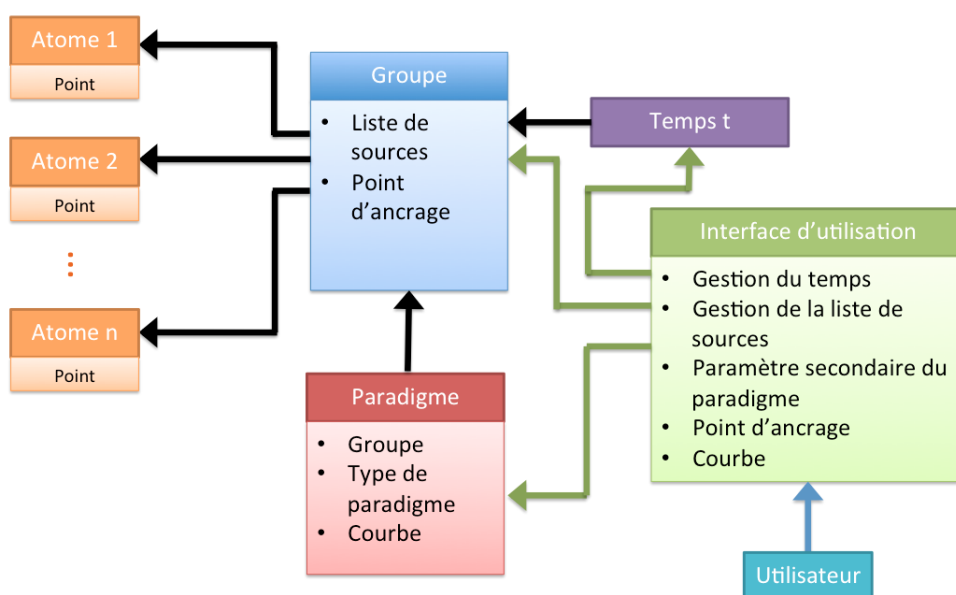


FIGURE 4.4 – Synoptique complet du système

Nous avons évoqué en introduction le besoin d'établir une sémantique nécessaire à la description d'une scène sonore. Les notions d'*atome*, de *groupe* et de *paradigmes* s'apparentent à celle de *syntagme* à *élément unique*. La modélisation de chacune de ces briques va nous permettre de construire la manière dont se construit le réseau sémantique de notre système, à l'image de la figure 4.5, qui esquisse l'encodage de chaque entité, ses propriétés mais surtout les relations d'appartenance et de contrôle qui matérialisent l'aspect *descriptif* de notre langage.

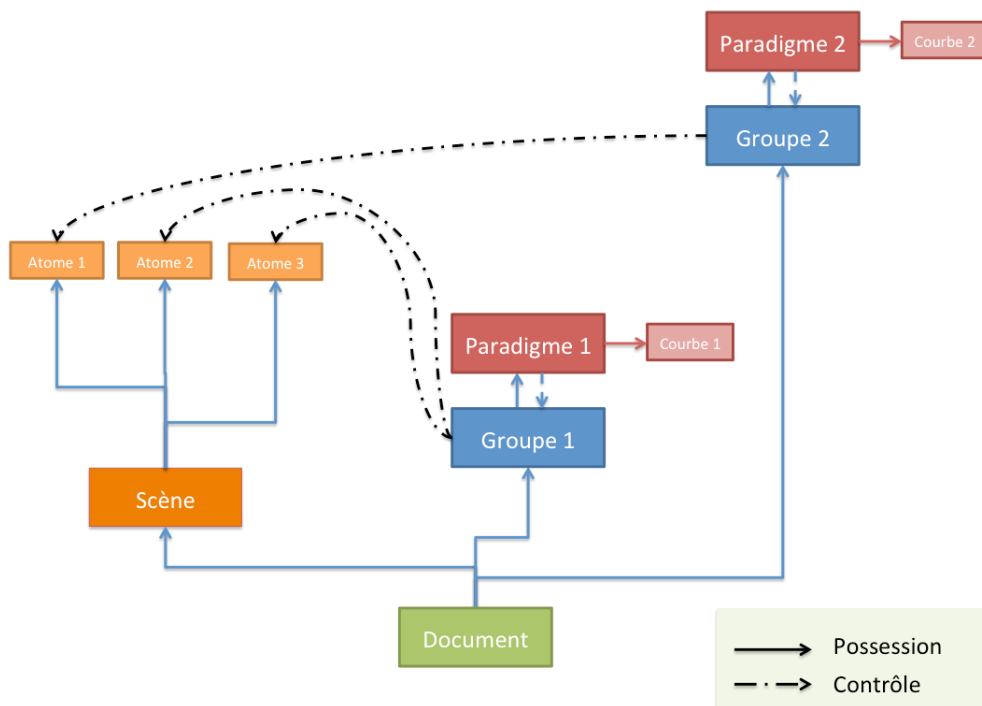


FIGURE 4.5 – Arbre sémantique du système

Ce schéma des différentes instances de notre système fera l'objet d'une description dans le langage XML qui propose une syntaxe "extensible" sous la forme de mot-clefs décrivant l'états de différents objets. On se référera à l'annexe A pour une description plus complète.

Dans les prochains chapitres, nous allons donc établir les différents principes constituant l'interface proposée à l'utilisateur. La démarche de réflexion sera la même que dans cette partie, à savoir une construction progressive de chaque élément, pour aller vers l'élaboration d'un système plus complexe, mais répondant toujours aux prémices énoncées en introduction à ce travail.

Deuxième partie

L'interface proposée au
créateur

Venons-en à la seconde partie notre projet : celui de construire une interface pour l'utilisateur qui réponde aux principes énoncés dans l'introduction, et tout particulièrement au **Principe 2** qui rend nécessaire l'utilisation d'une sémantique forte et pertinente pour élaborer notre système de mise en espace performatif, toujours articulé à la volonté artistique de son utilisateur.

Ainsi est-il important de fédérer et d'harmoniser au mieux la façon dont sont organisés les différents éléments moteurs. Ce chapitre a pour but, partant des éléments définis dans la partie précédente, de dégager les propriétés communes à chaque sous-système afin d'en proposer une utilisation naturelle et efficace.

Voici pour commencer, un récapitulatif des paramètres que l'utilisateur doit renseigner pour mettre en espace une série de sources sonores :

- Une série d'*atomes* présents dans la scène.
- Une série de *groupes*, pour lesquels est précisée la liste d'atomes qu'on souhaite y intégrer.
- Pour chaque groupe défini, l'utilisateur peut choisir d'employer un *paradigme* de pilotage des atomes. Le cas échéant, la position ainsi que le déplacement des entités restent libres.
- Pour chaque *paradigme* défini, l'utilisateur doit saisir une *courbe* ainsi qu'un *paramètre secondaire*.
- Pour chaque groupe, l'utilisateur peut définir si le point d'ancrage est à considérer comme un atome. Dans ce cas, il pourra appartenir à un autre groupe, et donc être lui aussi piloté.

Cette série de paramètres doit être dynamique, pour permettre une évolution aisée de chaque concept au cours du temps.

Notre système possède donc trois dimensions, une première **spatiale** qui correspond à l'espace de la scène sonore, une seconde qui relève de la **structure**, c'est à dire l'organisation des atomes au sein de groupes contrôlés ou non par un paradigme de pilotage, ainsi d'une dimension **temporelle** qui va définir l'évolution de la scène. Chaque paramètre a donc un degré d'appartenance plus ou moins marqué à chacune de ces dimensions. Par exemple, l'appartenance d'un atome à un groupe est un élément de structure ; il peut aussi dépendre du temps mais ne dépend pas de l'espace.

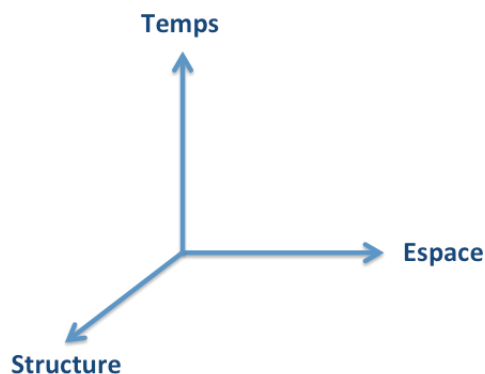


FIGURE 4.6 – Les trois dimensions de la représentation des données.

Une interface claire

Dans les systèmes informatiques, l'interface graphique est le dispositif responsable du dialogue entre l'utilisateur, un humain, et une machine. De l'interface *shell*, en ligne de commande, aux smartphones actuels, peu de temps s'est déroulé, et contrairement à ce que l'on peut penser, ce n'est pas tant le manque de capacité des ordinateurs d'antan à gérer des graphismes complexes qui a été un frein au développement d'interfaces élaborées, que le fait que les ordinateurs aient mis un certain temps à se démocratiser. En effet, l'informatique a longtemps été

un domaine réservé à des catégories sociales et professionnelles réduites qui possédaient une grande maîtrise technique et scientifique. Quand les ordinateurs n'étaient destinés qu'à une application très spécifique, la question de l'interface ne se posait pas vraiment et le recours à la ligne de commande était de mise. C'est au début des années quatre-vingt, avec la commercialisation de l'ordinateur *Xerox star* par la société Xerox que l'idée d'une informatique à disposition du grand public, et surtout d'un public non initié, émerge face au constat que " les usagers de nouveaux ordinateurs [étaient] souvent frustrés et déçus par de lourdes procédures de manipulation, des messages d'erreurs obscurs, des systèmes intolérants et confus au comportement incompréhensible, mystérieux et intimidant " ¹ .

Pour répondre au **Principe 2** proposé dans l'introduction qui pose le besoin d'une sémantique en adéquation avec le langage employé à la mise en espace du son, il est aussi nécessaire de définir une interface naturelle, qui sache se faire "oublier" de l'utilisateur, et dont la manipulation s'abstienne de tout recours à ce qui pourrait relever d'un jargon informatique ou technique inutile.

Notre système va donc proposer une interface utilisateur sous la forme d'un *environnement* constitué d'un ensemble de vues orientées, en fonction d'une série de tâches à effectuer, mais aussi du contexte dans lesquelles elles interviendront. Si l'on se réfère à la figure 4.6 *supra*, on obtient une série de trois vues principales que sont l'interface d'édition graphique, l'interface d'édition temporelle ainsi qu'une interface de gestion des données constituant une scène sonore. Au cours des prochains

1. *An introduction to human-computer interaction*, Paul A. Booth, Psychology Press 1989.

chapters de cette partie, nous allons définir la manière dont l'utilisateur interagit avec notre système.

On peut noter que nous avons jusqu'ici traité les termes de *partition* et d'*interface* comme des synonymes. Pourtant cette équivalence n'est pas évidente en soi. Nous allons voir dans le chapitre suivant comment le rapprochement de ces deux notions peut être légitime et en quoi il peut être fructueux.

Chapitre 5

La partition

La diversité des outils de création numérique d'aujourd'hui offre aux artistes un environnement de travail qui englobe chaque étape de la production d'une œuvre : de l'étape de composition à celle de la réalisation sonore, en passant par celle de l'interprétation, l'omniprésence de l'informatique est représentative de modes de fonctionnements qui ont profondément évolués au cours du XIX^e siècle. Et ce phénomène n'est pas uniquement constaté dans le domaine de la musique électroacoustique. Ainsi beaucoup de compositeurs de musique instrumentale utilisent-ils les outils de la Musique Assistée par Ordinateur dans les phases d'idéation de leurs œuvres. La mise en partition n'est effectuée que plus tard, et souvent dans le seul but de servir de matériel d'orchestre.

De plus, l'explosion des styles d'écriture de la musique, ainsi que leur complexification a conduit certains compositeurs à définir de nouveaux formats de partitions. On peut citer le compositeur américain Colon Nancarrow (1912-1997), dont le travail s'est centré exclusivement autour de l'étude de polyrythmies complexes dont l'exécution n'était à l'époque possible que par le biais d'un piano mécanique. La notation de

ses œuvres ne pouvait donc se faire que sur des rouleaux perforés.

La question d'une notation se pose donc avec l'émergence d'une musique dont le matériau n'est plus instrumental mais fondé sur des objets sonores concrets ou de facture électro-acoustique. Il faut donc développer une écriture adaptée à des besoins qui ne sont plus les mêmes que ceux rencontrés dans la musique instrumentale. Ainsi, nous allons détailler dans ce chapitre la manière dont les données sont gérées par l'utilisateur, et surtout la façon dont notre principe va s'insérer dans un processus de création. Pour commencer, il nous faut définir le cadre général, le document sur lequel vont s'appuyer le compositeur et le système pour créer la scène sonore, c'est à dire la *partition*. Cette dernière devra englober tous les objets que nous avons définis au cours de la partie précédente afin de fournir au compositeur un *canevas* propice et fédérateur.

Difficultés d'un système de notation pour fixer la musique électro-acoustique

Parmi les premiers questionnements à propos des sons « extramusicaux » on peut signaler, en 1913, le Manifeste de l'artiste futuriste italien Luigi Russolo, *L'Art Des Bruits*. Y est établie une classification des bruits en fonction de leur esthétique sonore. L'artiste réalisera des machines sonore, les *Intonarumori*, préfigurant d'une certaine manière ce que sera plus tard la musique concrète que Pierre Schaeffer théorise avec son *Solfège des objets sonores*¹ où il généralise la notion de note en tant qu'objet sonore, entité de base de tout l'univers sonore. Il inclut à ce solfège les sons qui n'ont pas de hauteur définie, dont le profil, les caractères, ne rentrent pas dans les critères habituels d'une musique

1. *Traité des objets musicaux*, Pierre Schaeffer, Paris Seuil 1966.

plus traditionnelle. Ces deux tentatives pour une catégorisation du son peuvent être considérées comme représentatives de la profonde confusion qui règne alors sur l'univers sonore. La musique instrumentale a toujours élaboré son système de notation à la fois sur le geste, sur le « faire jouer », considérant la sonorité globale comme acquise par le lecteur, et sur une quantification de tous les paramètres tels que hauteur, durée, dynamique, etc. En revanche l'univers sonore implique une causalité qui n'est plus d'ordre instrumental — on ne « joue » pas du train musicalement — et devant le caractère continu et infini de la palette sonore, il n'est plus possible ni de se fonder sur une connaissance a priori des sonorités ni d'établir un ensemble d'échelles quantifiantes.

Bien que relativement adaptée à une pratique instrumentale globale, permettant même une certaine modularité, la notation musicale dans sa version du début du XX^e siècle se révèle totalement inadaptée à une musique électroacoustique. Dans ses recherches au studio d'essai de l'ORTF, Pierre Schaeffer court-circuite très rapidement cette problématique : parce que trop structurant, le système de notation traditionnel influe directement sur l'idée musicale, il impose des relations privilégiées entre les objets musicaux. Selon lui, il faut donc s'en défaire, tout au moins lors de la phase de création.

Avec Schaeffer, on peut donc enregistrer l'échec d'une notation unique et globale de la musique électro-acoustique et l'impossibilité de réunir sur un même support toutes ses fonctions : élément de genèse de l'œuvre ou « brouillon » du compositeur, guide destiné à l'exécution, guide pour l'auditeur et trace pour la conservation de l'œuvre.

Le système d'écriture de l'espace que nous proposons ici n'est ni *total* ni *unique*, et ce pour plusieurs raisons. La première vient du fait qu'il ne prend pour l'instant en compte que la mise en espace des sons, laissant ainsi de côté tous les autres paramètres inhérents à la composition sonore. Notre principe s'insère "en parallèle" à un autre système générateur, quant à lui de sons. La seconde raison tient au fait que nous n'avons pour l'instant défini aucun élément de syntaxe proposant une interaction souple avec la partition.

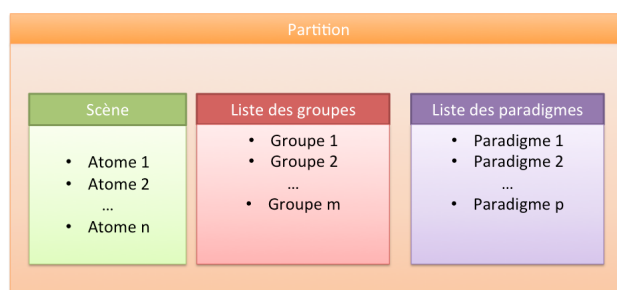


FIGURE 5.1 – La structure d'une partition dans notre système.

Une sémantique forte

L'idée de base de notre système est, rappelons le, de fournir un intermédiaire de travail au compositeur pour la mise en espace d'une scène sonore, par la mise en place de paradigmes de pilotage des sources ou atomes. Il n'est à aucun moment question de raisonner uniquement en termes de coordonnées spatiales, mais bien de se poser la question d'une représentation plus englobante et surtout plus musicale. Les paradigmes montrés en exemples dans le chapitre 3, le suivi d'une courbe, la cinématique des gaz et les *boids* possèdent l'aspect d'un " geste musical " dans le sens où ils établissent une mise en espace fondée sur la représentation d'un mouvement, d'une motricité plutôt que sur une simple description spatiale. On peut d'ailleurs rappeler que l'hypothèse heuristique for-

mulée en introduction à ce travail évoque l'existence d'une expérience *pré-réfléchie* qui occulte d'une certaine façon la manière dont est effectuée une action au profit de l'objet de cette action. Cela s'applique ici par une certaine *automatisation* de la tâche à effectuer — déplacer chaque atome — au profit d'une spontanéité de l'acte.

Ainsi, l'objectif de notre principe est bien de proposer une manipulation qui s'apparente à un *geste* que l'on veut qualifier de *musical*.

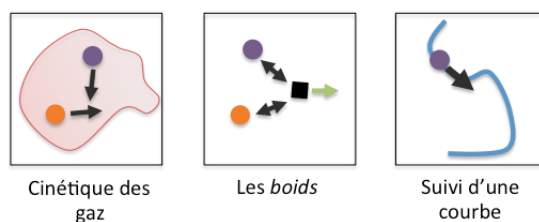


FIGURE 5.2 – Exemples d'icônes pour les paradigmes

La figure 5.2 montre l'exemple d'une série d'icônes pour les trois paradigmes cités dans le chapitre 3. Selon l'adage qui veut qu'un "bon croquis vaut mieux qu'un bon discours", la notation des événements de mise en espace sera fondée sur un jeu d'icônes similaire dans notre système. Ainsi, si l'on reprend l'exemple proposé dans le chapitre 2, figure 2.6, le principe de notre notation devient plus complet (figure 5.3).

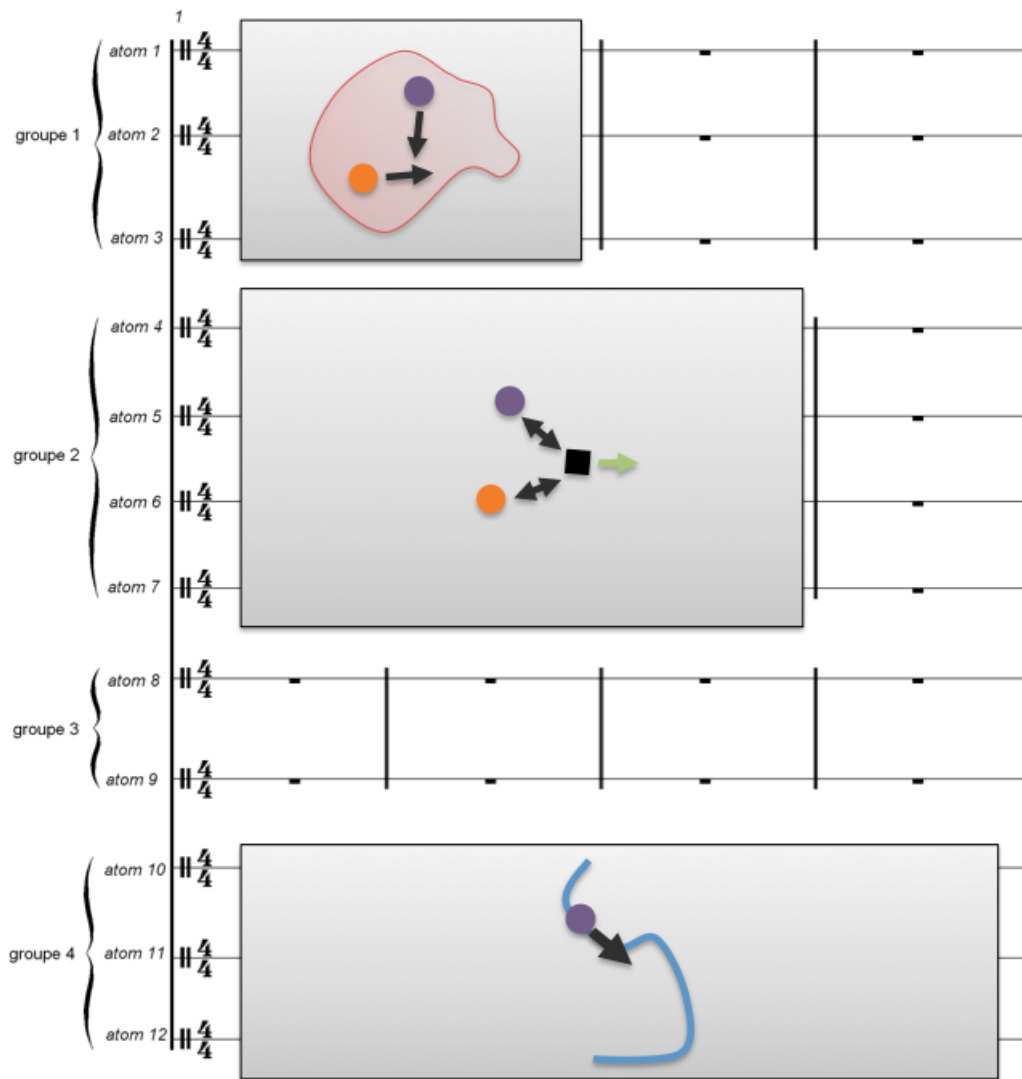


FIGURE 5.3 – Vers une notation plus complète

Cette notation met bien l'accent sur la notion de groupes, montrant de façon claire les interactions entre chaque atome, entre chaque groupe, faisant ainsi émerger une notion d'*orchestration spatiale*, fondement de notre dissociation entre spatialisation et mise en espace que l'on a proposé en introduction à la première partie. Elle reste cependant incomplète voire ambiguë, car il lui manque encore un certain nombre d'informations, notamment une description plus complète du cheminement spatial des sources, de la description complète du chemin à employer, de la vitesse, etc.

Finalement, nous sommes dans un cas de figure où le "conducteur" s'apparente à la notion d' "*aural score*"² proposée par György Ligeti et Rainer Wehinger en 1958 pour la pièce *Artikulation*, composée par Ligeti et dont Wehinger réalisa une interprétation graphique de la bande audio en utilisant une série de codes graphiques en couleurs pour symboliser les différents événements sonores.

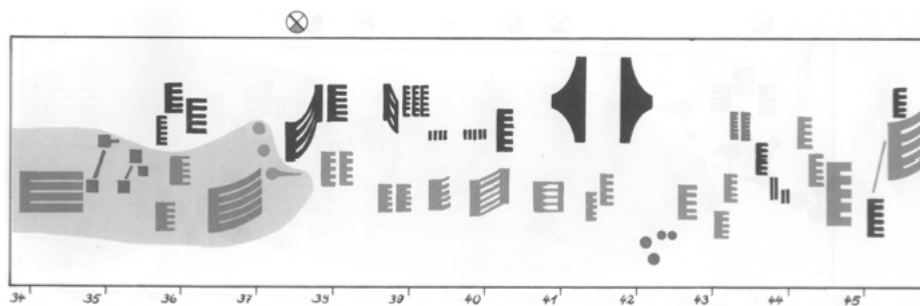


FIGURE 5.4 – *Artikulation*, Ligeti 1958. Partition réalisée par Rainer Wehinger

2. Kurt Stone, dans son ouvrage *Music Notation in the twentieth century* décrit l'aural score comme "*une partition destinée à l'auditeur [dont] la notation peut varier d'un code graphique complètement inventé (qui réinterprète les sons enregistrés avec ou sans l'aide de couleurs) jusqu'aux schémas les plus contrôlés et détaillés, tels des graphiques, calibrages, courbes et autres lignes, formes, lettres, numéros ainsi qu'un arsenal de symboles incluant parfois ceux de la notation conventionnelle*" .

Avant de poursuivre

Nous avons défini l'élément de travail du compositeur dans sa démarche de mise en espace sonore, le document ou la partition comme canevas central de notre interface. En revanche, sa description reste pour l'instant incomplète et ce pour la principale raison que le modèle que nous avons suivi jusque là, celui de la partition musicale "traditionnelle" n'est pas suffisant pour construire une sémantique complète de notre principe. Cela tient particulièrement au fait qu'il est nécessaire de représenter une structure de données complexes dont la dimension est supérieure à deux — la limite d'une partition, du moins pour des phénomènes continus — sans avoir recours à une codification *discrète* dont la notation musicale traditionnelle fait un usage intensif pour les notions de modes de jeu, nuances, etc. Ainsi, la représentation des données selon trois axes telle que définie en figure 4.6 page 61 pose comme nécessaire la création d'un support d'écriture et de visualisation complémentaire à celui que l'on vient de présenter.

C'est à partir de l'espace des paramètres qui, rappelons-le, est défini selon trois axes, le temps, la structure et l'espace, que nous allons effectuer deux projections orthogonales dont l'axe commun est celui de la structure : la première selon l'axe de l'espace proposera une **représentation du temps** tandis que la seconde, selon l'axe du temps constituera une **représentation spatiale**. Nous allons construire ces deux vues au cours des chapitres suivants.

Chapitre 6

Une interface tangible de dessin

Dans le but de pouvoir créer de manière naturelle l'espace dans lequel vont évoluer les sonorités, il nous faut maintenant nous poser la question du type d'interface que l'on va proposer au compositeur pour dessiner des courbes, placer des points, en un mot pour créer et interagir avec l'espace de la scène sonore. L'industrie des jeux-vidéo ainsi que le secteur de la création multimédia ont été à l'origine de nombreuses médiums visant à la manipulation de contenu informatique. Parmi les plus représentatifs, on peut citer la *Kinect*, une manette de jeu originellement destinée à la console *Xbox 360* de Microsoft, sortie en 2008, dont l'utilisation a rapidement été détournée vers d'autres applications comme les installations numériques. En effet, la principale caractéristique de cette manette est d'être immatérielle : il s'agit en fait d'une caméra qui va suivre les mouvements du joueur, mais aussi permettre son interaction par commande vocale. Plus récemment, en 2013, la société Leap Motion a développé un système de captation 3D de la position de la main

appelé *Leap Motion Controller*. Ce périphérique permet de contrôler directement par la main et les doigts un contenu proposé sur un ordinateur, et, à la différence de la Kinect, ne fonctionne qu'en proximité du capteur. Il n'en reste pas moins que ce système est très puissant et promis à un bel avenir.

Ces deux dispositifs peuvent être représentatifs de la nature des capteurs de mouvement en 3D. Du côté des périphériques d'interactions dans un plan, on trouve depuis longtemps les tablettes graphiques, à stylet ou non, et les tablettes à écran tactiles depuis une dizaine d'années. Bien que ne semblant pas convenir à notre espace, que l'on a défini comme 3D, nous allons voir comment de tels systèmes peuvent être très adaptés à la création de notre interface.

Dans les années 1970, Iannis Xenakis invente l'Upic (Unité Polyagogique Informatique du CEMAMu), un outil de composition musicale. Développée au sein du Centre d'Études de Mathématique et Automatique Musicales, la machine est achevée en 1977 et Xenakis l'utilise pour sa pièce *Mycènes Alpha* (1978). D'un point de vue matériel, l'UPIC est une tablette graphique reliée à un ordinateur, avec un affichage vectoriel. L'utilisateur dessine des formes d'ondes et des enveloppes de volume, qui sont ensuite traitées par l'ordinateur. Une fois les formes d'ondes enregistrées, l'utilisateur peut composer avec le matériau en dessinant des "compositions" sur la tablette. L'axe des abscisses représente le temps et celui des ordonnées les fréquences. Les compositions peuvent être étirées sur des durées qui vont de quelques secondes à une heure. Elles peuvent être aussi transposées, retournées, inversées et soumises à plusieurs algorithmes de transformation. Le système permet une performance en temps-réel en déplaçant simplement le stylet sur la tablette.

Le but de notre interface d'édition spatiale est justement de se réapproprier ce qui a fait la force de l'Upic : son aspect tangible et immédiat, tout en y ajoutant certaines fonctionnalités propres à notre système.

La bidirectionnalité

En 1985, Bill Buxton¹ conçoit à l'université de Toronto une technologie tactile multipoint, la *Multi-touch Tablet*, permettant la gestion simultanée de différents objets à l'écran via la position des doigts, mais aussi en détectant la pression d'appui. A l'inverse des autres principaux systèmes tactiles créés jusque ici, la *Multi-touch Tablet* utilisait un système de détection constitué d'un réseau *capacitif*, et non une caméra embarquée réduisant ainsi l'épaisseur de la tablette. Le principe de la tablette tactile telle qu'on la connaît actuellement est née. Il ne lui manque qu'un retour visuel direct pour être complet. En effet les diverses expériences conduites à l'époque en viennent à utiliser des cartons perforés à placer sur la tablette afin d'en délimiter certaines zones attribuées à différents paramètres.

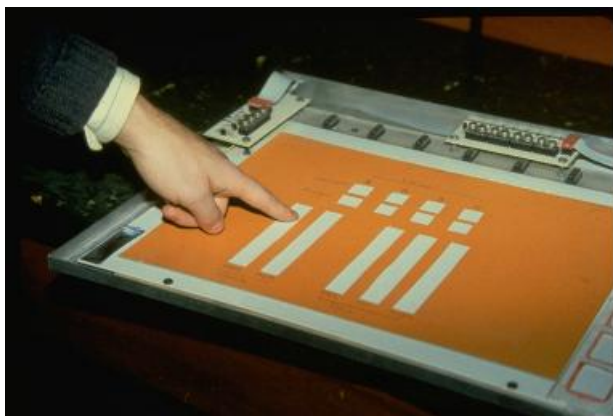


FIGURE 6.1 – la *Multi-touch Tablet* complétée par un patron de carton

1. *A Multi-touch Three Dimensional Touch-Sensitive Tablet*, Bill Buxton, Proceedings of the 1985 Conference on Human Factors in Computer Systems, CHI '85, San Francisco, 1985.

En 1990, le département des sciences informatiques de l'université du Maryland publie une série de recommandations techniques visant à la construction d'écrans tactiles multi-point². Ce principe de gestion d'interface est depuis devenu un standard dans l'élaboration de systèmes informatiques, à la fois professionnels et grands publics. La preuve en est l'importante prolifération actuelle de téléphones, tablettes mais aussi d'ordinateurs à écrans tactiles, à tel point qu'une tendance dans l'élaboration des applications pour Smartphones et tablettes a émergé, notamment du côté de la marque Apple, qui outre le fait d'encourager les concepteurs à fournir des interfaces tactiles très élaborées et naturelles, posant aussi la condition d'une extrême simplification des possibilités et des fonctionnalités de telles applications. Le credo actuel, en terme de contenu est de « faire simple », et les bonnes raisons sont multiples : la taille de l'écran qui reste réduite, l'absence de clavier matériel nécessaire pour la saisie de données quelques peu complexes, la volonté de ne pas fournir de manuel utilisateur, etc. Cela ne pose sans doute pas de problèmes dans la majeure partie des cas d'utilisations d'applications mobiles — utilisations ludiques avec les jeux, à visée informative ou personnelle — mais la question doit être soulevée dans le cas d'applications professionnelles complexes.

Il est justement un cas d'interface tactile non bidirectionnelle très usitée dans le milieu professionnel, particulièrement dans la création de contenu visuel : la tablette graphique. Bien qu'à ranger du côté de la souris, la tablette graphique permet de contenir le mouvement du pointeur dans cette zone géographique délimitée qu'est la surface de la tablette.

2. *A new era for touchscreen application : High precision, dragging icons and refined feedback*, Andrew Sears, Catherine Plaisant et Ben Shneidermann, Human-Computer Interaction Laboratory, Department of Computer Science, University of Maryland, 1990

Bien sûr, il n'y a pas de retour visuel direct, mais la gestion de contenu, et surtout la capacité à dessiner d'une manière plus « traditionnelle » en fait un outil puissant et particulièrement adapté.

Ainsi, la nécessité de sortir du paradigme "clavier/écran/souris", qui impose un intermédiaire entre l'objet que l'on manipule et le mouvement à effectuer, nous conduit à envisager un autre moyen pour construire notre scène sonore. La dimension tangible de la tablette tactile semble très à propos. Dans l'introduction à cette partie nous avons par ailleurs fait état du besoin d'une sémantique adaptée à notre système, dont l'inspiration serait à trouver du côté du geste musical : la plus-value apportée par un contrôleur tactile contribue d'avantage à la création d'une gestuelle forte fondée sur l'hypothèse qu'il est possible de modéliser un mécanisme sous-jacent à la création, une dynamique *pré-réfléchie* dont l'apprentissage relève exclusivement de l'apprentissage d'un mouvement générateur d'une caractéristique perceptive.

Terminologie

Premièrement, la question du nombre de points de contact est primordiale dans la conception d'une interface utilisateur. En effet c'est le centre même du paradigme d'utilisation d'un ordinateur : une souris, qui lorsqu'on la déplace, change la position d'un pointeur à l'écran et qui permet à l'utilisateur de « montrer » au système l'action qu'il souhaite réaliser en cliquant dessus.

Pour des raisons techniques, mais aussi parce que les premiers systèmes tactiles étaient dédiés à un univers informatique centré autour d'un ordinateur standard, la technologie *single-touch* est la première à voir le jour. Avec elle, le système n'est capable d'interpréter qu'un seul

appui sur la surface tactile.

Description	Paramètres
Appui simple	Coordonnées $\{x, y\}$ pointées
Appui maintenu	Coordonnées $\{x, y\}$ pointées temps t d'appui
Appui répété	Coordonnées $\{x, y\}$ pointées Nombre de répétition N_r

FIGURE 6.2 – Les éléments gérés par un système *single-touch*

Rapidement obsolète, il a été remplacé par un autre système, notamment grâce à l'essor des téléphones à écran tactiles : la technologie *multi-touch*.

Description	Paramètres
Pour chaque appui	Coordonnées $\{x, y\}$ pointées temps t d'appui Nombre de répétition N_r
Nombre d'appuis simultanés	N_a

FIGURE 6.3 – Les éléments gérés par un système *multi-touch*

Dans un premier temps, ce système a permis de partager une interface de contrôle entre plusieurs utilisateurs ; mais une tendance a rapidement émergé, celle de la reconnaissance gestuelle. Dès lors, il est possible pour le système de reconnaître la rotation de plusieurs doigts et de définir ainsi des règles d'utilisation qui ne sont plus fondées uniquement sur le principe de « pointer » un élément à l'écran pour l'activer. Par la suite, nous appellerons ce paradigme *détection de gestuelle*. Voici l'exemple de la détection de gestuelle du système *iOS* d'Apple, proposé sur iPad et iPhone :




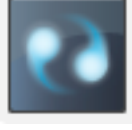
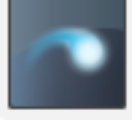
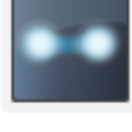
	Description	Paramètres
	<i>Tap</i> : détection d'un ou plusieurs appui	Position de chaque appui, nombre d'appuis
	<i>Long Press</i> : détection d'un appui maintenu	Position de l'appui, temps de maintien
	<i>Pinch</i> : pincement à deux doigts	Distance entre les deux points, vitesse
	<i>Rotate</i> :Rotation à deux doigts	Angle de rotation, vitesse
	<i>Swipe</i> : Déplacement rapide et instantané, à un ou plusieurs doigts	Direction, nombre de doigts utilisés
	<i>Pan</i> : Déplacement continu, à un ou plusieurs doigts	Nombre maximum de points touchés, vitesse, vecteur de translation instantané

FIGURE 6.4 – Les éléments de gestuelle établis pour le système *iOS* d'Apple

Pour résumer, nous avons la possibilité de choisir entre trois paradigmes de gestion tactile d'une interface :

- *Single-touch* : ce système se fonde sur le paradigme des icônes pointées par une souris. Bien que relativement usé, ce système reste d'actualité, notamment pour la gestion de tâches simples de second ordre.
- *Multi-touch* : ce système permet d'imaginer des alternatives au « glisser-déplacer » d'un objet en proposant, par exemple, de toucher l'élément que l'on souhaite déplacer, puis, tout en gardant le premier contact appuyé, de toucher la destination du mouvement.
- Détection de gestuelle : Très riche, ce système ouvre la possibilité de pouvoir manipuler différents éléments directement à l'écran, avec une gestuelle définie pour une tâche donnée.

Il ne s'agit évidemment pas d'avoir à décider lequel de ces systèmes est le meilleur, ni d'en choisir un pour l'interface que l'on souhaite réaliser, mais bien d'en réaliser une synthèse en faisant le choix du paradigme le mieux adapté à la tâche à accomplir.

Le retour visuel

«Everything is best for something and worst for something else»

Bill Buxton

Puisque notre interface est définie comme *bidirectionnelle*, il nous faut maintenant déterminer quelles informations l'utilisateur verra s'afficher à l'écran. On l'a dit, une des principales qualités d'une tablette est sa portabilité ; mais cela impose de proposer un contenu sur un écran de résolution plus réduite, et donc de simplifier la vue qui ne peut avoir la dimension de celle disponible sur un écran d'ordinateur. Il nous faut donc être sélectif et concis dans

l'élaboration de la vue. On introduit donc l'idée d'une hiérarchisation du contenu en fonction du contexte d'utilisation du système par le fait qu'il n'est pas systématiquement nécessaire de "tout montrer" à tout moment.

De plus notre espace de spatialisation a été défini comme "*idéal*" en ce qu'il ne tient pas compte de la réalité de sa réalisation effective par un système de diffusion donnée. Dans le but de proposer un retour visuel sans *indirection*³, c'est-à-dire sans décalage entre ce que l'on voit, ce que l'on manipule et la réalité sonore de ce que l'on perçoit, il est nécessaire proposer au compositeur un ensemble de repères dans l'espace dont la matérialisation pourrait être sous la forme d'icônes représentant les haut-parleurs, ou encore le schéma d'une installation.

La construction de notre interface d'édition spatiale

Voilà que sont définis tous les éléments constitutifs de cette partie de l'interface. Celle-ci se fonde sur un large plan dans lequel il est possible de dessiner des courbes, qui vont servir à la construction des paradigmes de pilotage des atomes. Cette construction sera facilitée par une intégration de la gestuelle propre aux tablettes tactiles, et dont l'aspect dynamique servira son apprentissage pour un utilisateur accoutumé à des gestes que l'on peut qualifier de "musicaux". Cette volonté de rapport direct, sans intermédiaire entre le projet de l'artiste et la réalisation effective de la mise en espace est aussi soulignée par l'emploi d'éléments de syntaxe forts.

3. Terme informatique qui désigne la possibilité d'un intermédiaire sous la forme d'une référence entre un concept à manipuler et l'auteur de cette manipulation.

Le cheminement d'écriture d'un paradigme de pilotage ressemble donc au suivant :

- L'utilisateur définit un groupe dans lequel il place une série d'atomes.
- Il dessine une courbe dans le plan.
- Il a la possibilité de modifier à l'envi cette courbe, d'y ajouter ou supprimer des points, d'en faire une boucle, etc.
- L'utilisateur fait le choix d'un paradigme de pilotage : suivi de la courbe, déplacement aléatoire, boids, etc.

L'ordre proposé ici n'est pas immuable : chacune des actions présentée peut intervenir à n'importe quel moment, et ce dans le but de garantir une souplesse à l'utilisation, surtout dans une phase d'idéation où la recherche compositionnelle peut amener à effectuer de grandes modifications, voire de faire table rase.

Nous allons maintenant voir comment articuler l'autre vue qui s'attache à l'aspect temporel de la partition. Comme nous l'avons précisé dans l'introduction à cette partie, la visualisation de la partition du côté de la tablette concerne les paramètres relatifs à l'espace et à la structure. Nous allons maintenant voir comment articuler l'autre vue, qui accentue l'aspect temporel de la scène sonore.

Chapitre 7

Une interface temporelle

Comme nous l'avons énoncé dans le chapitre 4 relatif à la gestion du temps, il est absolument nécessaire de proposer à l'utilisateur un outil d'articulation temporelle des différents paradigmes de pilotage qu'il aura défini pour sa scène sonore. Cet outil devra privilégier un aspect performatif fort, afin de pouvoir être utilisé en temps réel, de manière souple.

La *Timeline*, un concept qui a fait ses preuves

La *timeline*, ou ligne de temps, est un concept très fort utilisé dans la plupart des logiciels d'édition musicale, video ou autre, dès lors qu'il est nécessaire d'écrire des événements dans le temps. Ce paradigme est en fait très naturel et proche de l'écriture musicale de partition ; en effet on retrouve une hiérarchisation verticale des sources — ou émetteurs — dont l'écriture des différentes parties se fait de manière horizontale selon un axe temporel. Ainsi, il est possible d'un coup d'œil de visualiser les différents états au cours du temps, de mettre en regard différentes parties, en un mot d'embrasser pleinement les évolutions et articulations d'une œuvre.

Ce qui caractérise la notation sur timeline, c'est le concept de *Piste* et de *Clip*. Les pistes s'empilent verticalement et sont meublées de clips, qui vont correspondre à différents événements : la lecture d'une séquence MIDI, d'un fichier audio ou encore d'un fichier video. Tous ces éléments sont repérés par un temps **absolu**, dont la vitesse de défilement peut être variable mais qui reste la même pour toutes les pistes. Dans le contexte de notre système, la notion de *clip* va s'apparenter à celle d'un groupe comme proposée dans le chapitre 2 :

Elément de structure 1.

Un clip est un conteneur pour un objet Group, possédant une coordonnée de départ t_i dans le référentiel temporel absolu ainsi qu'une durée T .

En accord avec la définition de la classe *Group* définie dans le chapitre 2, chaque clip possède un groupe d'atomes, un paradigme associé ainsi qu'une existence délimitée dans le temps.

Dans ce contexte, la notion de piste est plutôt vide de sens. Du moins a-t-elle une importance moindre que dans les séquenceurs audio, son principal but étant de permettre une polyphonie simultanée de différents paradigmes. Par ailleurs, les fonctions de mise en *solo* et de désactivation de chaque piste constituent de bons outils à la gestion d'un projet.

Les limites de la Timeline

Cependant, certains logiciels ignorent ce concept pour ouvrir le champ à d'autres systèmes d'écriture. On peut citer les logiciels issus de *Patcher*¹, *Max* et *Pure Data*, dont l'interface, le *patch*, offre la possibilité de relier des "briques élémentaires" de contrôle ou de traitement du signal via des connexions filaires. La gestion temporelle n'est donc pas intégrée à l'origine dans le système et c'est à l'utilisateur de construire sa propre conduite. De fait, il est assez rare qu'un patch créé avec un tel logiciel gère la temporalité de manière autonome et continue. Au contraire la pratique courante, orientée plutôt temps réel, invite l'utilisateur à envoyer une série de *cues*² au logiciel, déclenchant ainsi des événements selon une séquence définie.

Cette façon de composer et de gérer le temps de manière fragmentée peut être intégrée à une timeline via un système de marqueurs comme on peut déjà le rencontrer dans tous les séquenceurs audio tels que Cubase, Pro-tools, etc. On propose donc à l'utilisateur de placer des repères verticaux sur la timeline afin, d'une part de se repérer dans le projet, mais aussi de "faire des sauts" d'un marqueur à l'autre, brisant l'organisation définie par la succession temporelle des événements et offrant ainsi la possibilité d'arranger différemment la construction globale des parties.

1. Logiciel développé à l'Ircam par Miller Puckette dans les années 1980. Il sera commercialisé d'abord par la société opcode System en 1990, puis en 1999 par la société Cycling '74. Miller Puckette a quant à lui développé une version libre, *Pure Data*.

2. Sorte de marqueur temporel numéroté qui sert à découper une œuvre en parties de taille variable afin de créer un repère à sa conduite. On trouve aussi cette pratique dans les conducteurs d'orchestres, généralement sous forme de lettres (A, B, C, etc.) qui servent de marques de répétition.

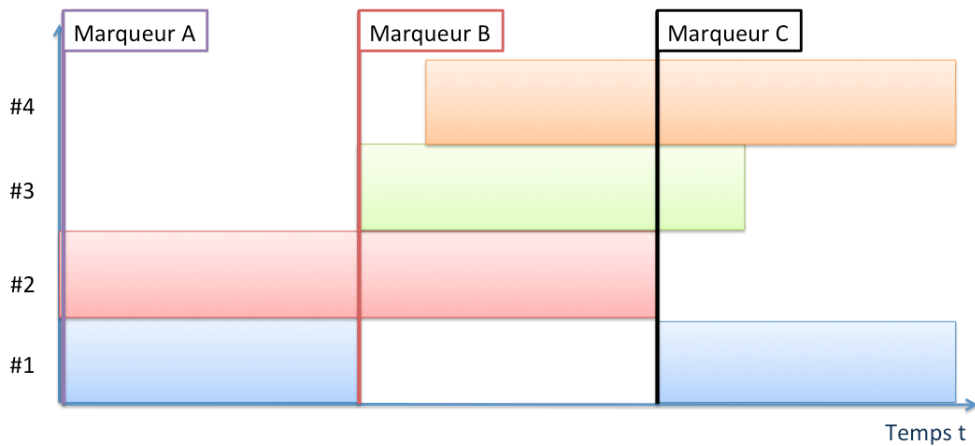


FIGURE 7.1 – représentation des marqueurs temporels sur la timeline

Une fois les différents marqueurs placés, il est alors possible d'en obtenir un agencement linéaire, dans une autre représentation que la timeline afin de contrôler soi-même le temps de passage d'un marqueur à l'autre grâce à un curseur. On peut aussi envisager le passage "brut" d'un marqueur à l'autre sans interpolation. On note par ailleurs que si l'utilisateur n'interagit pas avec le curseur, la lecture nominale, telle que définie dans la timeline, reste en vigueur.



FIGURE 7.2 – agencement linéaire des marqueurs

Le principal problème lié à cette représentation linéaire est qu'il conserve l'ordre des marqueurs tel que défini dans la timeline et empêche donc tout réarrangement. Pour briser cette linéarité, on doit alors en complexifier l'agencement.

La théorie des graphes

Un graphe est un ensemble de points appelés *sommets* qui peuvent être reliés par un ou plusieurs liens, orientés ou non, appelés *arêtes*.

Définition 6. Soient V un ensemble de sommets et E un ensemble d'arêtes reliant les sommets :

On définit le graphe G par le couple $G = (V, E)$.

On dit que le graphe G est simple s'il n'a ni liens multiples entre deux sommets, ni boucles, ni éléments isolés.

Dans notre cas, le graphe G généré par l'ensemble des marqueurs E est d'abord simple, et reste conforme au déroulement de la timeline : d'abord A, puis B puis C. Cependant, l'utilisateur peut aussi définir d'autres arêtes afin d'envisager d'autres arrangements des différentes parties, s'offrant la possibilité de passer du marqueur A au marqueur C, puis d'aller à B par exemple.

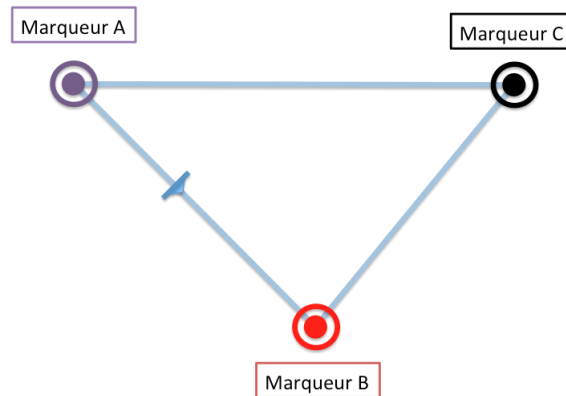


FIGURE 7.3 – agencement des marqueurs selon un graphe

Notre interface temporelle possède deux facettes d'une même temporalité. D'un côté la timeline et son caractère immuable, définitif, possède toutes les qualités d'une écriture musicale dans le temps ; et de l'autre côté, le système de marqueurs temporels offre la gestion temps-réel de l'œuvre, avec ses transitions, ses variations brusques, ses *accelerando* sous la forme d'une interface lisible et tangible.

Chapitre 8

Synthèse du système

Vers une interface hybride

L'introduction de la nouvelle console de jeux de salon Wii U par Nintendo en 2012 a apporté un nouveau concept dans le monde du multimédia : le *gameplay asymétrique*. Cette console propose une manette de jeu qui intègre un écran tactile, permettant ainsi la diffusion de contenu alternatif à celui destiné à l'écran principal, mais offrant aussi la possibilité de jouer de deux manières à un même jeu. Il devient ainsi possible à plusieurs joueurs de bénéficier d'un gameplay différent et individuel selon des phases de jeu bien déterminées.

L'immense intérêt apporté par la tablette graphique réside dans sa portabilité, sa légèreté et son potentiel de création et de contenu alternatif par rapport à ces systèmes, que l'on pourrait maintenant qualifier de « préhistoriques », que sont l'ordinateur, son clavier et sa souris. Ces atouts doivent cependant être nuancés car ils sont certainement insuffisants pour l'élaboration de systèmes complexes. En effet, la taille et la résolution des écrans proposés par les fabri-

quants de tablettes tactiles sont bien souvent très réduites, sans compter le fait que la main de l'utilisateur en masque très souvent une grande partie lorsqu'il touche des objets à l'écran.

Il semble alors plus opportun, et c'est ce que semble proposer Nintendo avec sa console Wii U, de considérer l'écran de la tablette comme un affichage alternatif dont le rôle serait de proposer, en premier lieu une visualisation des données jugées les plus pertinentes dans le contexte actuel, puis d'une manière générale, de proposer une visualisation complémentaire à l'interface utilisateur.

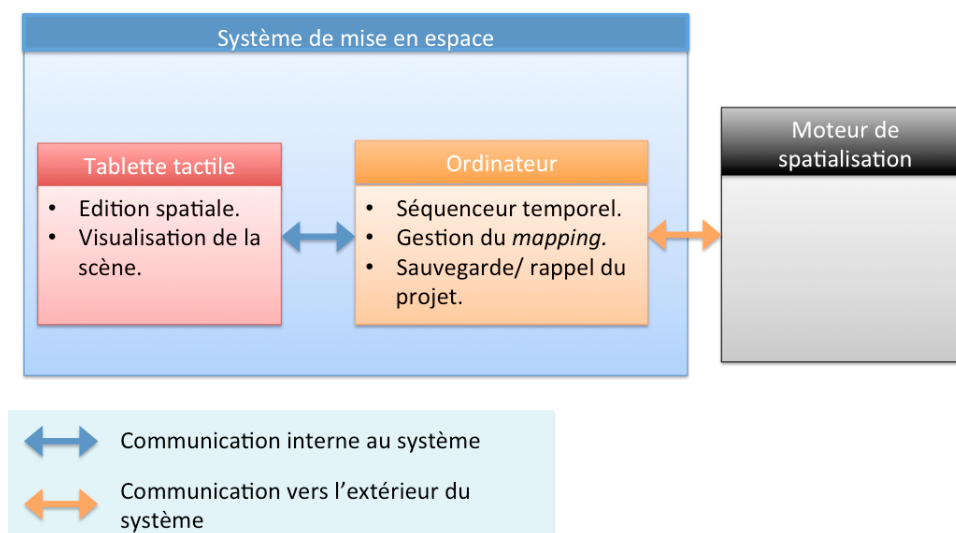


FIGURE 8.1 – Synoptique général du système

Ainsi la réalisation de notre système se fait sous la forme d'une application logiciel pour ordinateur, *augmentée* d'une surface graphique permettant l'édition et la visualisation des paradigmes par le geste. Afin d'optimiser au mieux son utilisation, le contenu du système est hiérarchisé en fonction du contexte de la manipulation à effectuer.

Comme nous l'avons évoqué dans le chapitre 6, l'introduction d'une interface d'édition sous la forme d'une tablette tactile suppose la réduction de notre espace, au départ 3D, à un simple plan. Cependant, il nous semble que cela soit la seule solution, à notre sens, au problèmes d'indirections induits par des périphériques dont l'interaction se fait par un intermédiaire gestuel : de la souris en passant par le *Leap Motion Controller*, le mouvement effectué par l'utilisateur n'est pas effectué dans l'espace de l'écran, il lui est donc nécessaire d'ajuster constamment ses mouvements pour les faire correspondre à ceux du pointeur affiché à l'écran. Une tablette tactile, de part son retour visuel direct, semble donc être un très bon moyen de manipulation *directe* de l'espace sonore, quitte à devoir perdre une dimension. Nous verrons plus tard comment remédier à ce problème.

Dans la première partie, nous avons définis le principe de notre système, qui repose les notions d'atomes, de groupes et de paradigmes articulés de manière temporelle et spatiale dans le but de composer la mise en espace d'une scène sonore. Puis, dans cette partie, nous avons esquissé une interface reposant d'une part sur la série de principes énoncés en introduction à ce travail, et surtout sur un aspect moteur et gestuel fort. Nous allons maintenant compléter cette esquisse en réalisant une série de prototypes visant à valider notre principe.

Troisième partie

Développement logiciel
d'une série de prototypes

Dans cette partie, nous entrons dans la phase de développement de prototypes afin de mettre en pratique le système construit précédemment. Chaque élément décrit fait l'objet d'une intégration au sein de programmes de taille ou d'ambition différentes, le point final étant la mise en place du système complet.

Les différents prototypes que nous allons évoquer sont les suivants :

- Une télécommande du logiciel *IanniX* pour tablette tactile précédemment réalisée dans le cadre d'un stage durant l'été 2012 au sein de la société *Buzzing Light*.
- Une interface de dessin de courbes pour tablette tactile.
- Une interface de contrôle de la trajectoire de plusieurs sources sonores pour tablette tactile.
- Une interface de séquenceur temporel pour ordinateur.
- Une implémentation de notre principe, incluant la partie tablette tactile couplée au séquenceur temporel.

Comme nous l'avons énoncé dans l'introduction, l'objectif de ce travail est de proposer un outil qui se détache de tout environnement de spatialisation, tout en conservant une interaction nécessaire forte. Il ne s'agit aucunement de se passer des différents organes constituant le studio de création, mais de trouver "une place" dans laquelle pouvoir insérer notre système de mise en espace.

Le premier constat est qu'il est primordial de faire en sorte que notre logiciel de mise en espace soit accueilli par les systèmes d'exploitation les plus en usage : *Windows*, *Mac OS X*, et *Linux*. De la

même manière, la multitude des moteurs de spatialisation se décline aussi en une multitude de formats : du patch *Max/Pd* au *plugin*, en passant par l'application *stand alone*, il est nécessaire de prendre en compte tous ces formats et de trouver un système fédérateur qui permette de les utiliser indifféremment.

Les interfaces de programmation

Concernant le développement logiciel sur ordinateur, notre choix s'est porté sur l'API¹ C++ *Qt* (à prononcer "cute"), développé par l'entreprise *Qt Development Frameworks* qui en propose une licence libre *GNU*. L'immense intérêt de *Qt* réside dans le fait qu'il est possible de porter très facilement l'application produite vers les environnements *Windows*, *Mac OS X*, et *Linux*.

En ce qui concerne l'application pour tablette, notre choix a été d'utiliser un *iPad* d'Apple pour la simple raison que cette tablette nous était facilement accessible. En termes d'ouverture à d'autres systèmes, ce choix n'est pas le plus pertinent dans la mesure où une application écrite pour les appareils d'Apple n'est malheureusement pas portable et doit donc être codée à nouveau. En revanche, l'environnement fourni par la société pour la programmation des applications est très documenté et le langage employé, l'*Objective-C* est très puissant.

1. API : Application Programming Interface ou interface de programmation. Système proposant à un utilisateur une série de *routines* visant à faciliter le développement ou l'utilisation d'une librairie logicielle.

Un protocole de communication

Dans un environnement de création, ce sont les protocoles MIDI² et O.S.C³ qui sont le plus couramment utilisés comme supports de communication entre les différents organes matériels ou logiciels du studio.

L'OSC, emploie le protocole UDP⁴ du réseau. Il s'agit donc d'une technique de communication totalement compatible TCP⁵ qui peut utiliser n'importe quel réseau Ethernet, câblé ou non. Il n'a pas besoin d'infrastructure dédiée. De plus, il peut servir en interne, dans un même ordinateur, à communiquer entre différentes applications.

Par ailleurs, la connexion MIDI est de type sérieelle et nécessite l'emploi d'un support de communication physique spécifique et dédié. Elle s'adapte très bien à la liaison USB, elle aussi sérieelle, mais requiert la mise en place de pilotes matériels sur l'ordinateur afin d'émuler un port MIDI traditionnel.

Nous avons donc choisi l'OSC comme protocole principal, autant pour les communications entre la tablette et le séquenceur que pour la communication vers le moteur de spatialisation. L'implémentation de ce protocole est explicitée en annexe D.

2. Ou *Musical Instrument Digital Interface*, la liaison de référence pour les appareils audio de tous types depuis le début des années 1980.

3. Ou *Open Sound Control*, un format de transmission de données temps-réel plus rapide et fiable que la norme MIDI.

4. Ou *User Datagram Protocol*, un des principaux protocoles de télécommunication employé par les réseaux informatiques.

5. Ou *Transmission Control Protocol*, autre protocole majeur de télécommunication employé par les réseaux informatiques.

Le support de communication

Comme on l'a dit, le protocole OSC nécessite l'emploi d'un réseau compatible. Dans le cas de la communication *interne* au système, celle se situant entre l'ordinateur et la tablette, nous avons fait le choix d'une connexion sans fil wifi. L'avantage immédiat est celui de la liberté de mouvement, de l'absence de "fil à la patte" pour l'utilisateur. Cependant, l'emploi de cette technologie pose un certain nombre de contraintes, qui peuvent aller à l'encontre d'une bonne expérience utilisateur :

- La communication par protocole UDP nécessite, dans notre cas, une connexion de type client-serveur, et donc de devoir spécifier à chaque instance du système l'adresse IP de l'autre. Il est aussi nécessaire de fournir au système deux numéros de *port*, un pour chaque sens de communication.
- Une connexion wifi peut être sujette à des décrochages, rendant son utilisation dans un contexte de performance assez risqué. L'utilisation pour le développement d'un iPad a d'ailleurs montré que cet appareil avait une tendance assez mystérieuse à se déconnecter du réseau wifi de manière assez fréquente.

La première contrainte concerne directement l'utilisateur du système car on présuppose qu'il possède un minimum de connaissances en déploiement et en configuration d'un réseau wifi. Il faut donc trouver un moyen de lui faciliter la tâche dans la saisie des paramètres d'adresse IP et de ports réseaux.

Depuis le début des années 2000, il existe un système appelé *Zeroconf*⁶, qui permet de proposer des services au sein d'un réseau

6. Apple appelle ce service *Bonjour*.

de manière automatique. En d'autres termes, les machines s'auto-détectent afin de se configurer elles-mêmes. Plus récemment, grâce à l'apparition des smartphones, il est possible d'avoir recours à un Qr-Code⁷, qui affiché à l'écran de l'ordinateur, peut être pris en photo avec la tablette et ainsi décrypter l'ensemble des informations nécessaires à une bonne connexion, qui s'effectue alors de manière automatique.

En revanche la seconde contrainte pose une question : de quelle manière se comporte le système en cas de perte de connectivité réseau ? Dans le cas d'une utilisation en temps réel, il est absolument impensable que le déroulement de la partition s'arrête. La réponse à cette question sort du sujet de ce mémoire, en revanche, il faut aussi notifier l'utilisateur de la perte de connectivité. Le principe utilisé est explicité en annexe E à la fin du mémoire.

7. Sorte de code-barre en deux dimensions.

Chapitre 9

Une télécommande pour

IanniX

L'intégration du logiciel *IanniX* dans une œuvre sous-entend bien souvent que celle-ci est très dispersée dans l'espace, à l'exemple de l'installation de Charles De Meaux à Yeosu en Corée pour l'exposition internationale de 2012. Il s'avère alors très fastidieux de devoir faire des aller-retour incessants entre la régie et les différents pôles du dispositif afin d'en vérifier le bon fonctionnement et de peaufiner les derniers réglages. Le premier intérêt d'une application *IanniX* pour tablette graphique n'est donc pas d'émuler point pour point l'application native proposée sur ordinateur, mais plutôt de réaliser une télécommande proposant une vision différente de la partition graphique jouée. Si l'immense intérêt de *IanniX* est de proposer une répartition graphique des objets, il n'en demeure pas moins que l'accès et la gestion de ces objets, mais aussi des groupes, n'est pas facile dans le paradigme de gestion actuel.

Ainsi la principale fonction de la télécommande est-elle de per-

mettre l'édition à distance des paramètres de chaque objet présent dans la partition Iannix, *via* une présentation non plus spatiale, mais selon les groupes et le type des objets. De cette proposition découle immédiatement une autre, qui permet la complémentarité entre l'application native et son interface pour tablette : cette dernière doit pouvoir être utilisée comme support d'information supplémentaire lors de travaux sur la partition, comme une sorte d'écran supplémentaire à l'ordinateur permettant d'émuler la zone de gestion des objets. Enfin l'application pour tablette propose une émulation des fonctions de transport du projet ainsi que la capacité de changer l'état des *triggers* ainsi que des *curseurs*. Ceci est particulièrement pratique pour pouvoir tester rapidement les actions émises par chaque objet, en les décontextualisant de la partition. Ainsi, si un *trigger* déclenche une séquence sonore, il suffit simplement de le toucher sur la tablette pour vérifier le bon déroulement de l'événement.

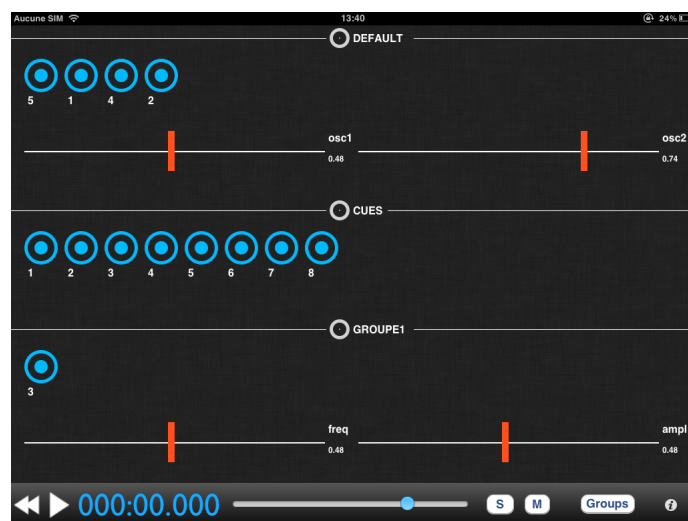


FIGURE 9.1 – Un aperçu du logiciel IanniX pour *iPad*

Test in situ

Lors de l'installation *Cosmogonium Domestique*¹ réalisée dans le cadre de la majeure "scénographie" à l'ENS Louis Lumière et présentée au public du 9 au 11 novembre 2012 à la Mix Box à Paris, nous avons utilisé le logiciel IanniX comme séquenceur principal des différents événements sonores de l'installation. Il a donc été possible de tester l'intérêt, ainsi que le bon fonctionnement, de cette application en situation réelle.

Les points développés

Le développement de cette application a permis la réalisation d'un *socket* OSC pour l'iPad ainsi qu'une prise en main générale de l'environnement de développement *Xcode* proposé par Apple.

1. Une installation interactive réalisée par Christophe Da Cunha, Manuel Deneu, François Heller, François Mallebay et Yoan Vincent.

Chapitre 10

Une interface de dessin de courbes pour tablette tactile

Nous allons maintenant nous pencher sur la question du dessin de courbes sur une tablette tactile. Le but de ce prototype est de dégager une méthode pour définir une trajectoire plane, puis de l'éditer afin d'en modifier les points, d'en ajouter ou d'en supprimer.

Le schéma d'utilisation est donc le suivant :

- L'utilisateur pose le doigt à l'endroit où il souhaite commencer la courbe.
- L'utilisateur déplace son doigt, toujours appuyé sur l'écran, pour dessiner la trajectoire de la courbe. Chaque point de l'écran touché par le doigt vient s'ajouter à une liste.
- La saisie de la courbe s'achève dès lors que l'utilisateur lève le doigt de l'écran.

- Une fois la saisie achevée, le système convertit la série discrète de points en une courbe de Bézier (Algorithme explicité en annexe B).
- L'utilisateur peut alors déplacer la courbe dans le plan ou encore lui faire effectuer une homothétie en utilisant les gestes multi-points de *pincement* ou de *rotation à deux doigts*.
- L'utilisateur peut aussi éditer la position de chaque point de la courbe, en ajouter ou en supprimer par un *appui long* sur la courbe.
- L'utilisateur peut fermer la courbe en faisant coïncider le premier et le dernier points. Ceux-ci deviennent alors confondus. Il peut à tout moment les dissocier en effectuant un *double appui*.

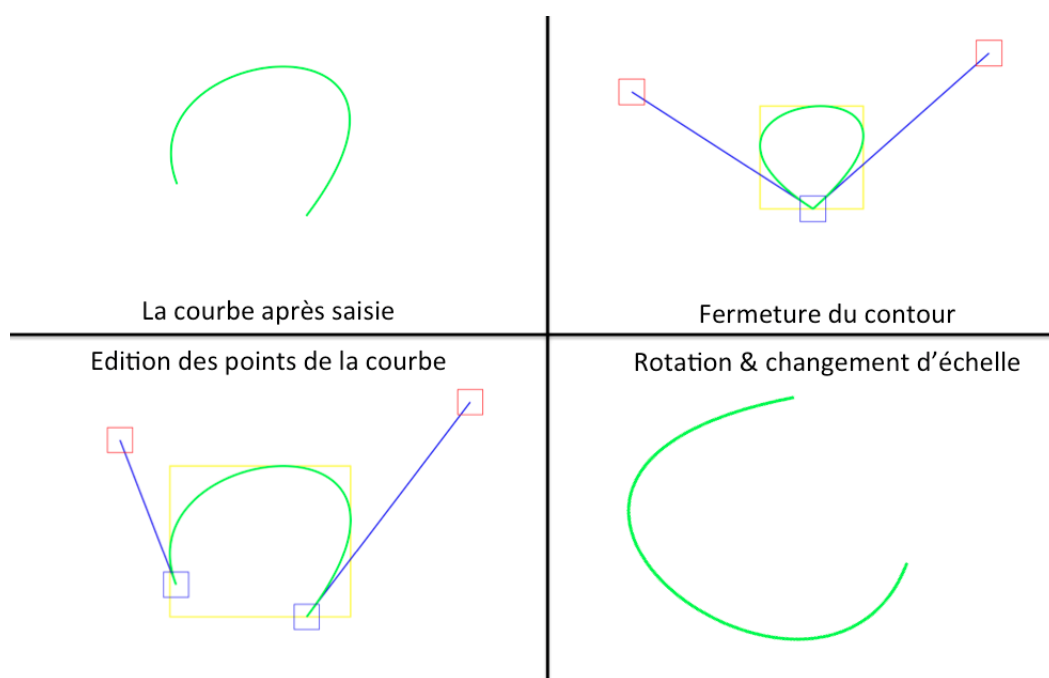


FIGURE 10.1 – Les différentes opérations sur une courbe





Description	
	Déplacement d'un point
	Ajout d'un point
	Redimensionnement global de la courbe
	Rotation globale de la courbe autour du point d'ancrage

FIGURE 10.2 – La gestuelle employée pour la manipulation d'une courbe

Les points développés

Le développement de ce prototype a permis l'implémentation des classes *Point* et *Courbe* respectivement définies au chapitre 5 page 23 et chapitre 4 page 37. La réalisation logicielle des transformations de l'espace définies chapitre 1 a été aussi nécessaire.

Chapitre 11

Une interface de contrôle de la trajectoire de sources

Le chapitre précédent a posé les bases d'une interface de contrôle de la trajectoire de sources dans l'espace. Nous allons donc maintenant combiner la fonction de dessin expliquée dans le chapitre 10 avec la fonction de contrôle des objets à distance via une liaison OSC définie dans le chapitre 9.

Le but de ce prototype va être de créer une interface simple pour tablette tactile dans laquelle il sera possible de dessiner une série de courbes qui serviront de trajectoires à des sources sonores. Les données sont ensuite envoyées *via* une liaison OSC vers un hôte hébergeant un moteur de spatialisation, ici l'objet *Max hoa.map* de la librairie ambisonique HOA.

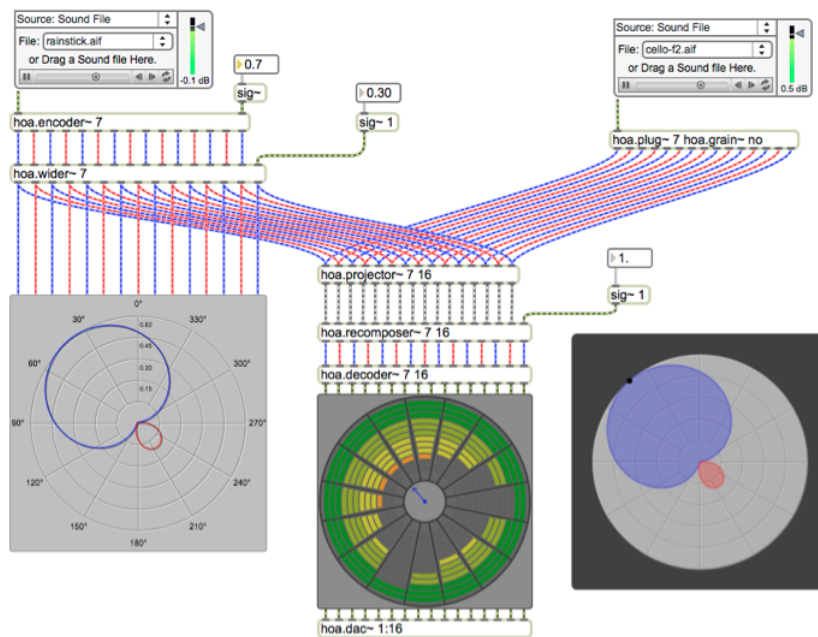


FIGURE 11.1 – La librairie HOA dans *Max*

Développée au sein du Labex Arts-H2H, Laboratoire d'excellence des Arts et Médiations humaines, et inscrit dans le contexte du CICM, centre de Recherche Informatique et Création Musicale situé entre l'Université Paris VIII et la Maison des Sciences Humaines de Paris Nord, la librairie HOA "*tente de mettre à la portée des musiciens des modèles de spatialisation du son basés sur l'ambisonie d'ordre supérieur et sur la synthèse de champs acoustiques*".¹ Offrant des codes sources C++ ainsi que des objets pour *Max*, la librairie permet au musicien de créer une synthèse d'un champ sonore, d'en modifier les caractéristiques *via* une série de transformations qui utilisent l'ambisonie mais aussi la *synthèse de fronts d'ondes*, l'ensemble dans un contexte d'utilisation simplifié, notamment grâce à l'emploi d'interfaces graphiques repensées pour

1. <http://www.mshparisnord.fr/hoalibrary/>

l'application au modèle ambisonique.

Dans le cadre des "Journées d'Informatique Musicale"² de 2013, la librairie HOA a fait l'objet d'une présentation au public suivie de la démonstration d'une interface de pilotage pour iPad. Cet atelier a été l'occasion de mettre en pratique notre système, et surtout d'en faire l'application directe à un moteur de spatialisation.

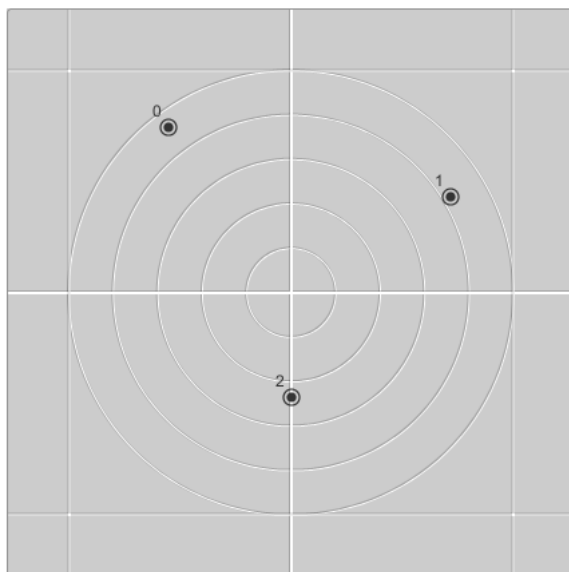


FIGURE 11.2 – L'objet *Max hoa.map* de la librairie HOA

Nous nous trouvons donc dans un cas de figure où un patch *Max* proposant une série d'objets de la librairie HOA est hébergé sur un ordinateur, que l'on souhaite piloter à distance *via* un iPad. Le premier point à considérer dans la construction de l'interface est celui de l'*indirection* (comme nous l'avons déjà évoqué dans le chapitre 6 pour ce qui concerne l'extension de notre principe à d'autres contrôleurs physiques) c'est à dire les décalages spatio-

2. Événement annuel réunissant des chercheurs en informatique musicale et différents acteurs de la vie musicale utilisant l'informatique comme moyen d'expression, comme aide à la composition, ou dans leur démarche musicologique.

temporels qui s'inscrivent entre la représentation graphique du logiciel à contrôler et le périphérique utilisé pour ce contrôle. Dans la mesure où notre périphérique de contrôle possède un écran, il faut donc que la vue du logiciel, ici l'objet *hoa.map* soit recopiée sur l'iPad partiellement, afin d'établir une visualisation unique du placement des sources. Sur la figure 11.2, on peut voir une série de cercles concentriques. Le cercle le plus à l'extérieur correspond au diamètre formé par les haut-parleurs, tandis que ceux qui se situent à l'intérieur sont de simples guides visuels, comme le sont aussi les axes verticaux et horizontaux. Du côté de la tablette, nous avons fait le choix ici de ne conserver que le cercle principal ainsi que le système d'axes (figure 11.10).

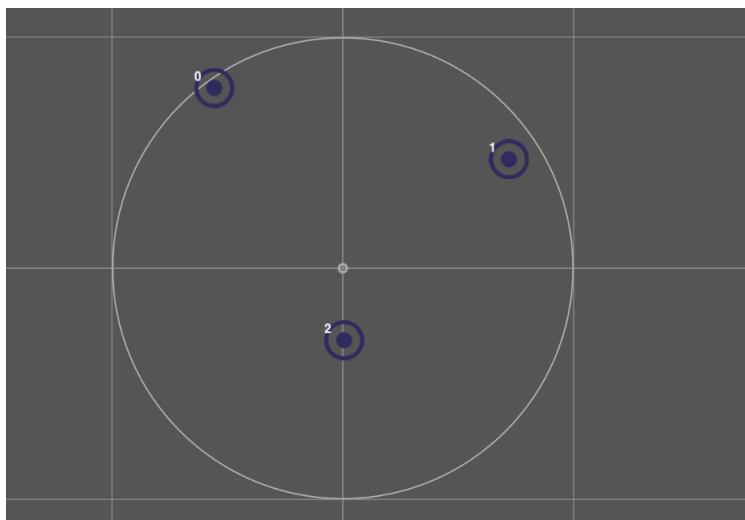


FIGURE 11.3 – La représentation de l'objet *hoa.map* sur iPad

La figure 11.10 nous montre aussi la représentation que nous avons choisie pour les *atomes* : la forme d'un disque entouré d'un anneau. Pour l'instant le système est très simple : on peut déplacer des atomes un à un ou plusieurs à la fois et il est possible de se

déplacer dans la scène et de la redimensionner.

Nous allons maintenant ajouter à cette interface la fonctionnalité de dessiner des trajectoires telle que nous l'avons définie au cours du chapitre précédent. Il est donc nécessaire de définir plusieurs états de l'interface, chacun correspondant à une catégorie de tâche à effectuer :

- Un mode de déplacement "libre" des atomes : cet état correspond à celui décrit précédemment.
- Un mode de dessin des courbes : ce mode va permettre à l'utilisateur de tracer une courbe avec le doigt afin qu'elle soit ensuite convertie en courbe de Bézier (Cf. Annexe B).
- Un mode d'édition point-à-point des courbes : cet état de l'interface est tel que nous l'avons défini dans le chapitre précédent.



FIGURE 11.4 – Les vues "Free" et "Edit"

On peut maintenant constater l'apparition d'une ligne de texte dans la scène. Il s'agit d'une aide contextuelle permettant de décrire brièvement à l'utilisateur la vue dans laquelle il se trouve.

La gestion de la structure

De la même manière que dans notre système, l'objet *Max hoa.map* propose à l'utilisateur de pouvoir ajouter des sources à piloter, mais aussi de les grouper pour en effectuer un contrôle simultané. Nous allons donc rendre accessible la modification de la structure à l'utilisateur par le biais de notre interface. Est alors introduite une vue secondaire, l'*inspecteur*.

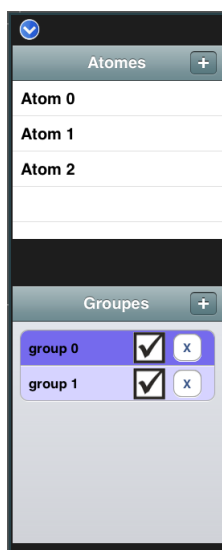


FIGURE 11.5 – La vue secondaire de l'inspecteur

Cette vue permet, d'une part de gérer la liste des atomes dans le projet, en permettant l'ajout ou la suppression d'une entité, et d'autre part, de gérer des groupes ainsi que l'affectation d'atomes en leur sein. Comme nous l'avons précisé dans le chapitre 6, il est important de hiérarchiser les vues dans une interface, ainsi la vue de l'inspecteur doit-elle être *flottante*, c'est-à-dire que l'on doit pouvoir la cacher afin qu'elle n'encombre pas l'écran si elle n'est pas nécessaire.

La gestion des trajectoires

Maintenant qu'il est possible de créer des atomes, de les grouper, de dessiner des courbes et de les éditer, nous allons mettre en place un séquenceur très rudimentaire, qui va permettre la réalisation des paradigmes, qu'il nous faut d'ailleurs aussi implémenter.

Pour ce prototype, le séquenceur est très simple : il s'agit d'un simple *timer* qui va venir incrémenter un compteur temporel toutes les 50 millisecondes. Chacun des paradigmes de pilotage viendra alors y lire l'avancement global dans le temps. En ce qui concerne l'interface, on ajoute une barre de "transport" en bas de l'écran qui propose les fonctions de *lecture/pause*, *stop*, ainsi qu'un curseur de *vitesse d'avancement globale* et un affichage de l'avancement temporel via un compteur au format *Heures :Minutes :Secondes.Millisecondes*.



FIGURE 11.6 – La barre de transport de la tablette

Grâce à la méthode de calcul de la position d'un point sur une courbe de Bézier fournie dans l'annexe C, il est maintenant possible d'assigner une trajectoire à un groupe d'atomes.

L'implémentation du paradigme de la *Cinétique des gaz*

Nous allons, de la même manière, intégrer à notre prototype la possibilité de piloter un groupe d'atomes selon le paradigme de pilotage modélisant le comportement de la cinétique des gaz décrit

dans le chapitre 3. L'implémentation n'est pas explicitée ici mais peut être consultée dans le code source fourni en annexe.

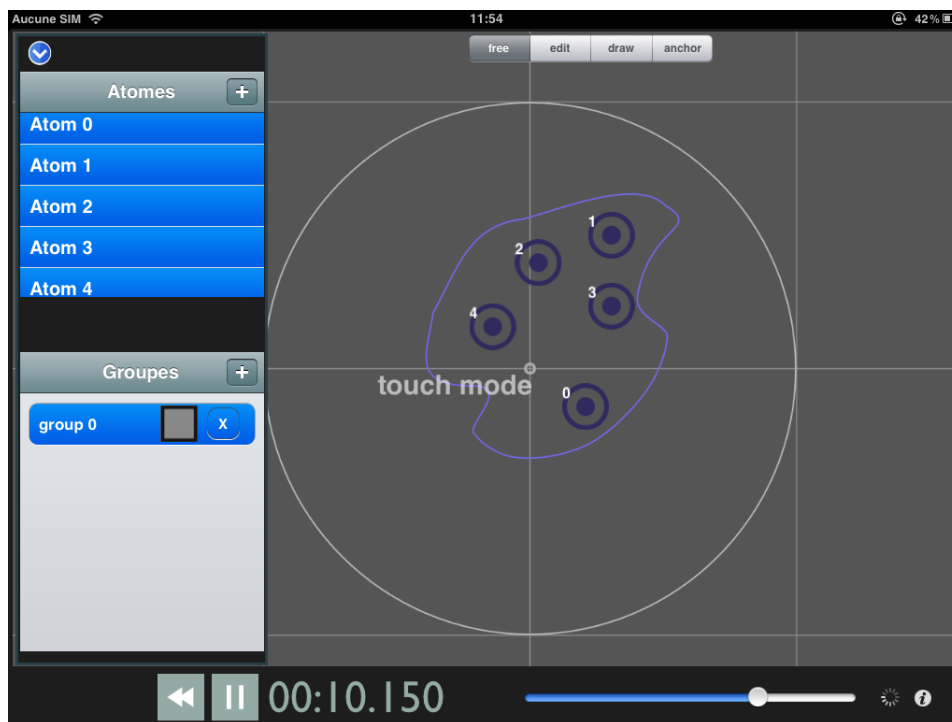


FIGURE 11.7 – La mise en place du paradigme de Cinétique des gaz

Nous nous trouvons maintenant dans le cas de figure où le compositeur doit faire le choix de piloter un groupe, soit par le suivi d'une trajectoire, ou par la modélisation d'un déplacement aléatoire au sein d'un contour fermé. Il doit aussi pouvoir faire le choix de n'appliquer aucun pilotage aux sources et les laisser statiques ou les déplacer manuellement. Ainsi est-il nécessaire de définir un moyen pour l'utilisateur de faire le choix du paradigme à appliquer. La première solution consiste à expliciter une liste des paradigmes disponibles dans l'inspecteur. L'utilisateur pourra alors, pour le groupe sélectionné, choisir s'il désire que les atomes suivent une trajectoire,

qu'ils se déplacent de manière aléatoire ou encore qu'ils soient libre ; respectivement les modes *path*, *aléa* et *free* comme montrés dans la figure suivante.

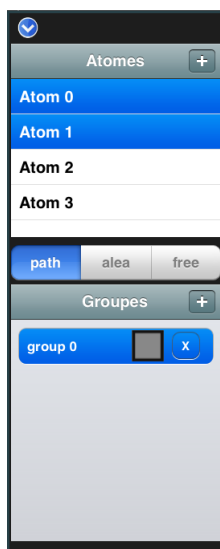


FIGURE 11.8 – L'inspecteur, complété par le choix du paradigme

La seconde solution consiste à exploiter la gestuelle proposée pour l'utilisation d'un iPad. Ainsi peut-on proposer le geste "*swipe*" (le déplacement rapide et instantané, ici à un doigt) sur un atome pour appliquer un paradigme. Si aucun paradigme n'est défini pour un groupe, un geste de *swipe* y appliquera le suivi de courbe, un second geste changera le paradigme pour celui du déplacement aléatoire, etc. Ceci n'est qu'une ébauche de la gestuelle de l'interface ; il serait possible de la compléter afin de la rendre plus performante.

Vers d'autres paradigmes

L'application de notre système à la librairie ambisonique HOA ouvre la perspective d'autres paradigmes de pilotage de l'espace sonore. Par exemple, l'objet *hoa.space* permet, via la décomposition

du champ ambisonique en ondes planes, de transformer la topologie de l'espace sonore, c'est-à-dire de mettre en valeur certaines portions spatiales au détriment d'autres.

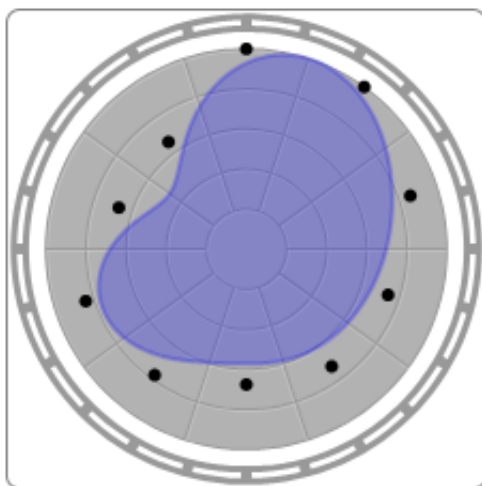


FIGURE 11.9 – L'objet *Max hoa.space* de la librairie HOA

Le principe de cet objet est simple : on place un certain nombre de microphones virtuels dans l'espace et il est possible pour chacun, de faire varier sa distance au centre, tout en conservant cependant sa position angulaire par rapport au cercle ambisonique. Il est en revanche possible d'effectuer une rotation à l'ensemble via la couronne extérieure. Nous avons donc défini tous les outils de mise en place de ce pilotage dans le chapitre 3, à savoir :

- La description d'un contour, ici fermé.
- La description d'un lien spatial entre différents points.
- La possibilité d'effectuer une série de transformations sur les points (rotation, translation, échelle).

La description des liens faite précédemment nécessite une légère reformulation, ou du moins, doit-elle être généralisée à tout type de

point pour ne pas être réservée aux seuls atomes. En effet dans ce cas, le paradigme de pilotage ne nécessite pas d'atomes, au sens de sources sonores, mais seulement des points de la courbe.

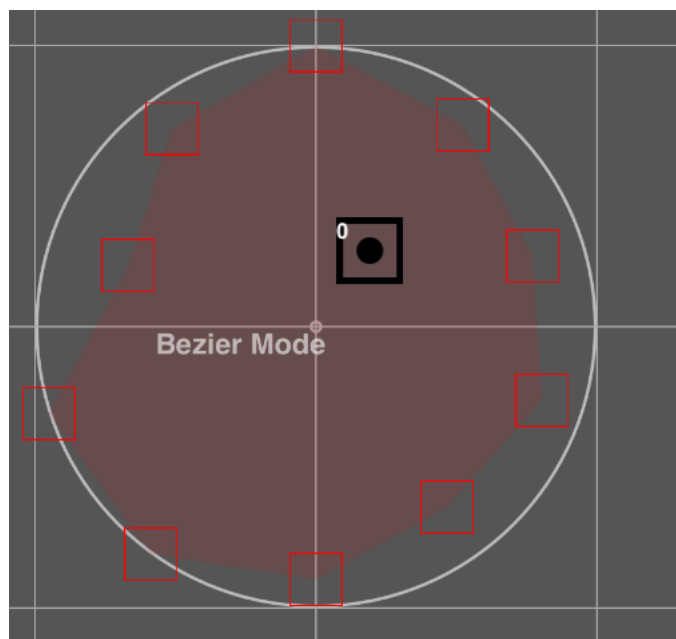


FIGURE 11.10 – La représentation de l'objet *hoa.space* sur iPad

L'ajout d'une interface de contrôle externe à la librairie HOA permet une réelle plus-value qui va dans le sens des objets de contrôle de l'espace sonore déjà proposés pour *Max*. Ce que l'on gagne à utiliser un tel système, c'est d'abord la portabilité offerte par la tablette mais aussi une amélioration substantielle des tâches nécessaires à la mise en espace, et surtout celles qui concernent l'évolution au cours du temps des événements sonores.

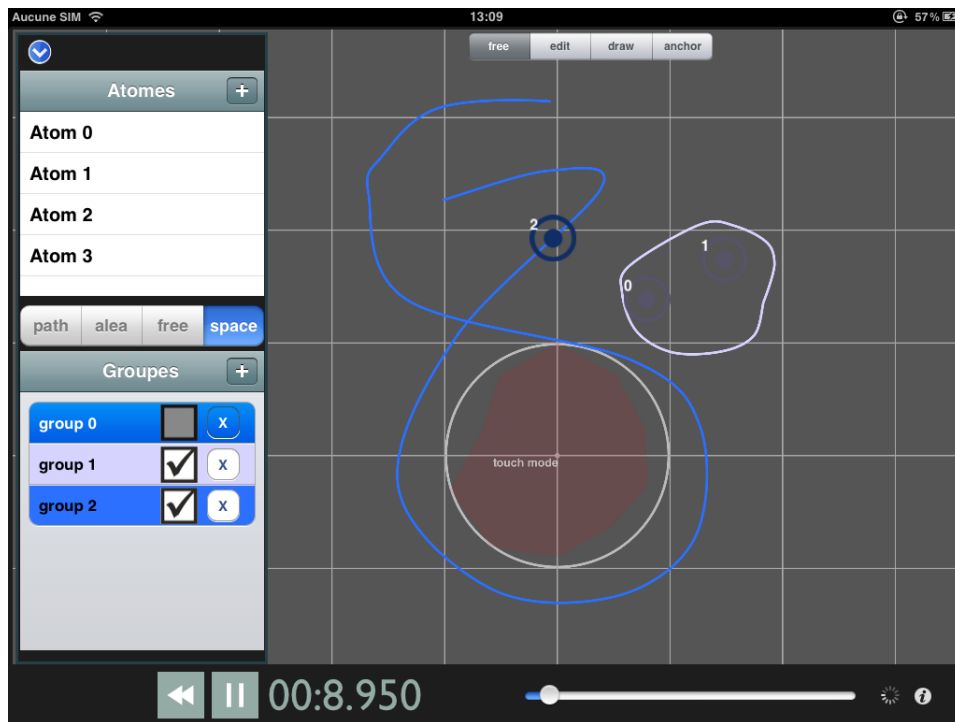


FIGURE 11.11 – L’interface de mise en espace complète présentée aux JIM 2013 pour la librairie HOA

Les points développés

Le développement de ce prototype a permis, d’une part l’implémentation de la classe *paradigme* définie dans le chapitre 3 et d’autre part la définition de la gestuelle à employer dans le cas d’une manipulation graphique des objets *Point*, *Atome* et *Groupe*.

D’un point de vue purement ergonomique, nous avons aussi décrit le comportement ainsi que la représentation des différences vues, c’est à dire celle de la scène sonore définie comme vue principale, sur laquelle viennent se superposer les vues du transport et de l’inspecteur, cette dernière pouvant être masquée pour ne pas trop encombrer l’écran.





Description	
	Déplacement d'un atome
	Redimensionnement de la scène
	Déplacement au sein de la scène
	Choix d'un paradigme de pilotage

FIGURE 11.12 – La gestuelle employée pour la manipulation des atomes et de la scène

Pour conclure

Cette interface constitue donc le premier pas vers l'élaboration complète de notre système. La partie relevant de son aspect spatial et graphique a été en grande partie définie. L'utilisation de cette interface a très vite montré les limites du séquenceur temporel que nous avons implémenté de manière très rudimentaire : il nous faut donc maintenant définir la manière dont le compositeur va articuler au cours du temps les différents paradigmes de pilotage des sources sonores. Ce sera l'objet du chapitre suivant.

Chapitre 12

Une interface de séquenceur temporel pour ordinateur

Partant des définitions posés dans les chapitres 4 et 7 relatives à la question de la gestion du temps, nous allons maintenant construire un prototype de l'interface temporelle de notre système. Le paradigme d'interface sous-jacent est celui de la *timeline* dont on va ici détailler l'appropriation pour notre principe de mise en espace de sons. Dans un souci de cohésion avec l'interface d'édition proposée dans le chapitre précédent, nous allons adopter la même organisation générale des données et des vues, afin d'harmoniser au mieux notre système.

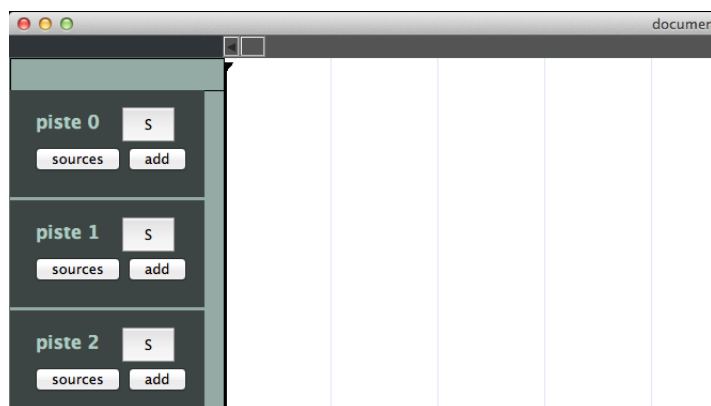


FIGURE 12.1 – Détail du séquenceur : les pistes

les pistes constituent l'élément d'organisation verticale de la mise en espace d'une scène sonore. Elles vont accueillir ce que l'on a défini dans le chapitre 7 comme étant un *clip*, c'est-à-dire un conteneur pour un groupe piloté par un paradigme dans une section temporelle délimitée.

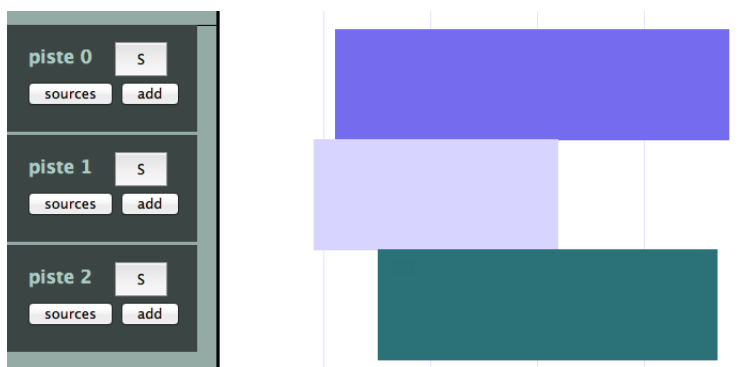


FIGURE 12.2 – Détail du séquenceur : les clips

Chacun des clips constituant la timeline peut être déplacé temporellement et verticalement ; il peut aussi être redimensionné. Dans le cas d'un paradigme proposant une vitesse d'avancement comme définie dans le chapitre 3, cette dernière est directement liée à la

mesure temporelle du clip. Dans le cas d'un suivi de courbe, la vitesse de parcours d'un atome sera d'autant plus petite que le clip sera large.

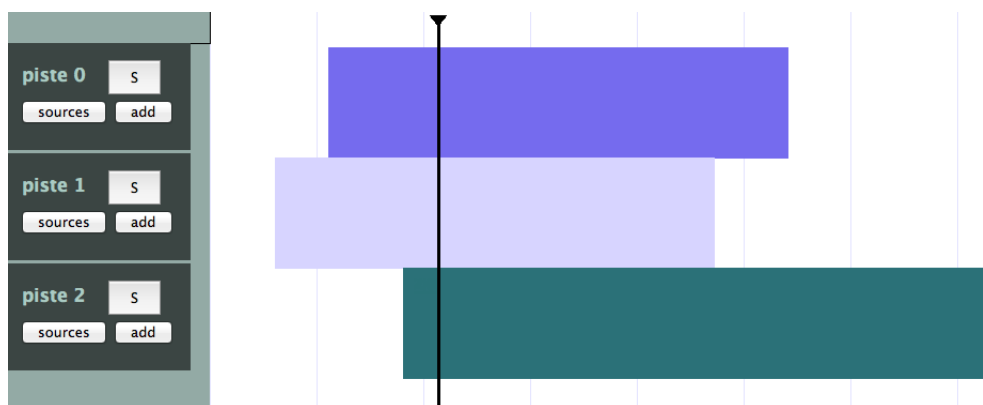


FIGURE 12.3 – Détail du séquenceur : la tête de lecture

Comme le montre la figure 12.5, la lecture des événements temporels se fait selon un curseur unique : la tête de lecture. Il nous faut donc réserver une partie de l'interface dédiée au *transport* dans la partition.

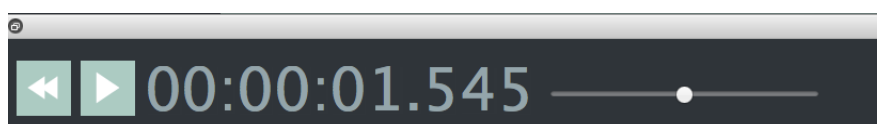


FIGURE 12.4 – Détail du séquenceur : la barre de transport

Dans un souci de hiérarchisation du contenu proposé au compositeur, la barre de transport peut être masquée.

La gestion de la structure

Nous allons maintenant nous attacher à la construction des vues annexes de celle de la timeline. En effet, comme nous l'avons précisé dans l'introduction de la seconde partie de ce travail, notre

représentation du temps dépend de la structure du projet, selon les mêmes modalités que l'interface graphique présentée dans le chapitre précédent. Ainsi ajoutons-nous à l'interface une vue secondaire, l'*inspecteur*, permettant la création d'atomes, leur assignation à des groupes ainsi que les choix de pilotages *via* les paradigmes.

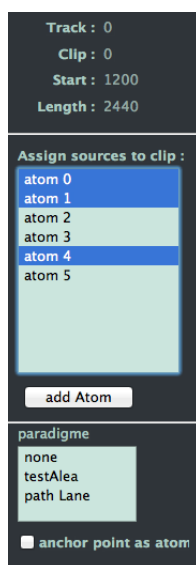


FIGURE 12.5 – Détail du séquenceur : l'inspecteur

On peut noter que certaines données présentées dans l'inspecteur dépendent du clip sélectionné dans la timeline :

- les données de *Track*, *Clip*, *Start*, *Length*, respectivement les numéros de piste, puis de clip, puis la date de départ et la durée du clip.
- Les atomes sélectionnés sont ceux qui sont assignés au groupe du clip. La liste reste commune à chaque groupe.
- Le paradigme sélectionné dans la seconde liste est celui qui va piloter le groupe du clip.
- Il est possible de définir le point d'ancrage du groupe comme un

atome, conformément à ce que l'on a énoncé dans le chapitre 3. Si la case "anchor point as atom" est cochée, le point d'ancrage du groupe apparaîtra dans la liste des atomes.

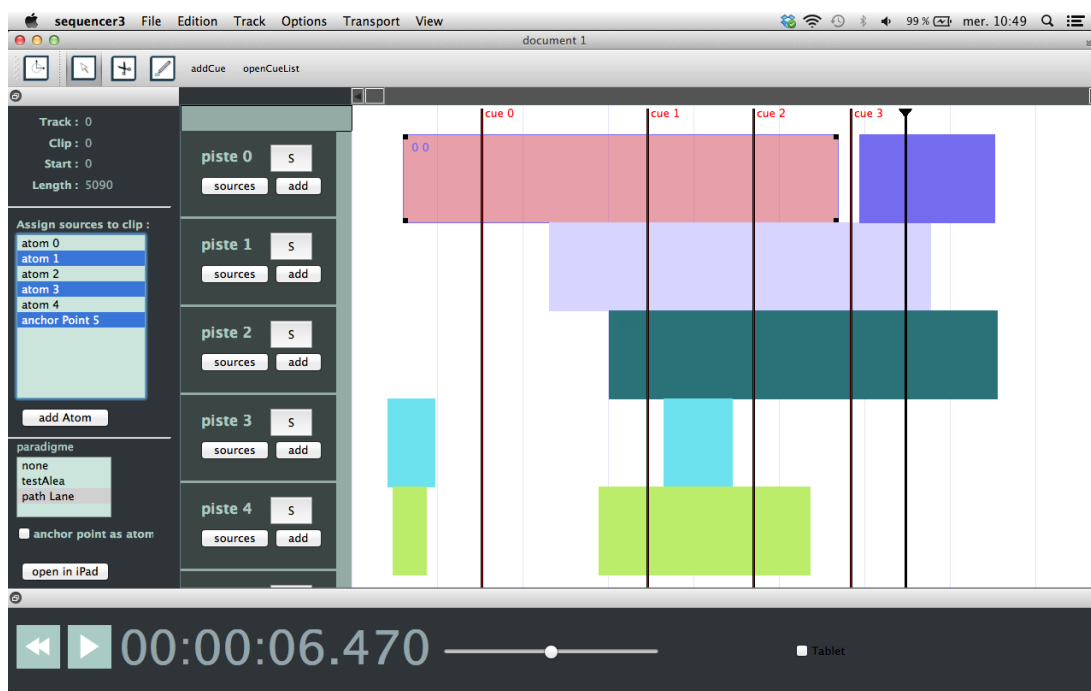


FIGURE 12.6 – Vue complète du séquenceur temporel

On peut noter dans l'interface (figure 12.6) la présence de marqueurs temporels — les vues — que l'on peut placer, déplacer, de manière souple dans la timeline.

Les points développés

Le développement de ce prototype a permis d'appliquer à notre système le principe de la timeline. On y retrouve toutes les opérations courantes de manipulation de segments temporels mais aussi de gestion d'un projet que l'on peut attendre de ce type d'interface. L'utilisation du framework *Qt* facilite d'ailleurs grandement

la tâche de développement des fonctions annexes — et pourtant essentielles — telles que le *copier/coller*, la sauvegarde d'une partition : en un mot les fonctions qui rendent l'expérience utilisateur fluide, naturelle et aisée.

Pour conclure

Pour l'instant, ce prototype est une "coquille vide" : l'édition spatiale des paradigmes n'est pas possible, car comme nous l'avons énoncé dans la seconde partie, il nous semble nécessaire de segmenter les parties temps et espace, dans l'optique d'une clarification de l'interface du système. Dans le chapitre suivant, nous allons définir les passerelles entre cette interface temporelle, et l'interface spatiale pour tablette tactile proposée dans le chapitre précédent afin de rendre notre système complet.

Chapitre 13

L'implémentation de notre système

Dans ce chapitre, nous allons utiliser les éléments des prototypes déjà réalisés pour construire une implémentation de notre principe qui, rappelons-le, va employer à la fois l'interface spatiale et la timeline de manière coordonnée.

Lors de l'élaboration de l'interface de notre système, nous en avons défini le synoptique général, page 89 figure 8.1, dans lequel nous avons proposé deux systèmes de communication. Le premier concerne le dialogue *interne* entre la tablette et l'ordinateur, le second, *externe*, va permettre le pilotage effectif d'un ou plusieurs moteurs de spatialisation. Ici, c'est la partie logicielle hébergé par l'ordinateur qui est *centrale*. C'est elle qui va gérer les fonctions de sauvegarde de la partition et de communication avec le "monde extérieur". La tablette est considérée comme périphérique, c'est à dire un satellite *client*.

La communication interne au système

Lors de l'élaboration du séquenceur temporel, nous avons défini l'existence de groupes, d'atomes et de paradigmes dans le temps, sans pour autant en permettre l'édition spatiale. Nous allons donc ajouter la possibilité d' "ouvrir" sur la tablette un clip créé dans la timeline. Ainsi, nous ajoutons à l'inspecteur du séquenceur l'action d' *ouvrir le clip dans l'iPad* via un bouton (figure 13.1).

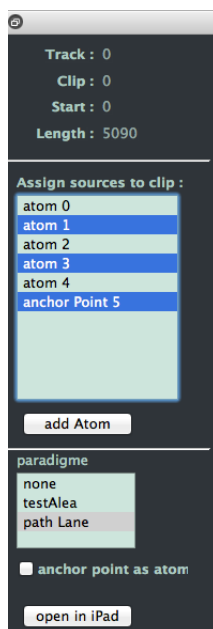


FIGURE 13.1 – L'inspecteur complété de l'interaction avec l'iPad

Cette action va permettre d'ajouter le clip à la liste des groupes dans la tablette, dans l'optique d'en éditer l'extension spatiale, à savoir son paradigme de pilotage.

Dans le sens inverse, dès lors que le compositeur créera une courbe, l'éditera ou fera toutes sortes d'actions visant à modifier l'état d'un des objets de la partition, la tablette notifiera le séquenceur de la modification afin de la prendre en compte.

L'explicitation de la communication interne au système est détaillée dans l'annexe E, page 149.

La visualisation de l'évolution de la scène sonore

Dans notre système, c'est l'écran de la tablette qui sert de retour visuel principal de la partition. Nous avons donc ajouté la possibilité, à la manière des *calques* des logiciels de retouches photo, de masquer un ou plusieurs groupes, grâce à l'ajout d'une icône dans l'entête de chaque groupe. De la même manière, il est possible de fermer un groupe depuis la tablette, une fois son édition terminée, sachant qu'à la fermeture une synchronisation avec le séquenceur est effectuée afin d'en sauvegarder le contenu. Il est alors possible de l'ouvrir à nouveau en le sélectionnant dans la timeline.



FIGURE 13.2 – Les actions de masquage et de fermeture d'un clip

Le contrôle du temps en direct

Comme nous l'avons énoncé dans le chapitre 4, notre système propose une gestion du déplacement temporel au sein de la partition, sous la forme d'un graphe, dont les sommets correspondent aux marqueurs placés sur la timeline (figure 12.6 page 120).



FIGURE 13.3 – Une série de marqueurs temporels dans le séquenceur

La figure 13.4 montre une série de marqueurs placés dans la timeline. Cette liste est transmise à l'iPad qui va la présenter sous forme de graphe, dans une vue différente de celle qui représente l'espace.

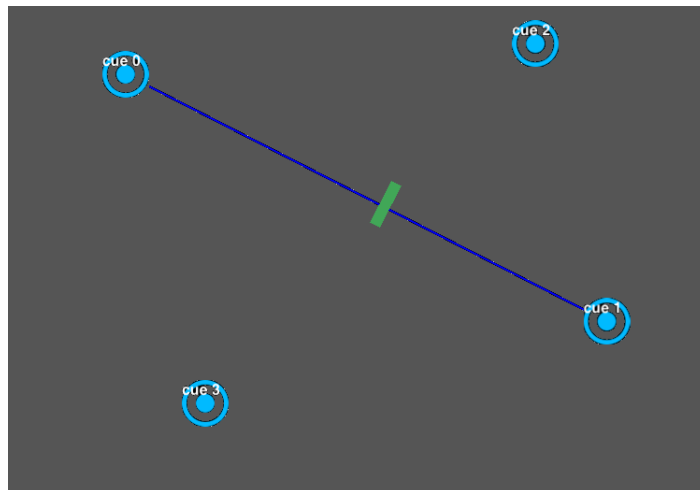


FIGURE 13.4 – Une série de marqueurs disposée dans un plan

Dans ce prototype, le choix du "chemin" à suivre s'effectue en sélectionnant deux marqueurs. Une ligne apparaît alors, sur laquelle peut évoluer un curseur qui va définir le taux d'avancement entre les deux marqueurs. Nous n'avons pas souhaité dessiner chaque chemin possible entre chaque trajet, car le graphe deviendrait trop vite illisible et donc inutilisable.

La poursuite du développement

A ce stade du développement, il y a un point dont nous n'avons pas pu tenir compte. C'est celui de la capacité des tablettes à réagir à des mouvements d'inclinaison et à l'accélération selon les trois axes spatiaux. C'est un aspect pourtant très prisé des concepteurs de jeux-vidéos pour mobile, et on dénombre quantité de jeux, notamment de courses automobiles dans lesquels c'est la tablette elle-même qui fait office de volant. De la même manière, Apple a un temps tenté de normer l'action *Annuler* par l'emploi d'une secousse sur l'appareil, à la manière des ardoises magiques qu'il fallait secouer pour effacer le contenu. La question sera donc d'étudier si ces mouvements peuvent s'appliquer à notre système, et quelles fonctionnalités ils peuvent déclencher.

Conclusion

Ce mémoire nous a donné l'occasion de construire une nouvelle approche de la mise en espace sonore. Tout au long de ce projet, nous nous sommes efforcé de garder une approche *systemique* et interdisciplinaire, qui a été fructueuse puisqu'elle nous a permis de chercher — et de trouver — des éléments de réflexion alternatifs, dans d'autres domaines de connaissances que ceux du son ou de la musique. Maintenant qu'une grande partie de l'implémentation logicielle de notre principe est réalisée et qu'une série de tests avec la librairie ambisonique HOA a été effectuée, il ne nous reste plus qu'à "éprouver" le système avec la réalisation d'une pièce composée par, et aussi pour ces nouveaux paradigmes de mise en espace. On pourra également en faire évoluer la liste, car il est finalement très aisé et porteur de sens de trouver des moyens alternatifs de "faire vivre" et d'animer des sons dans un espace. Et la présentation au JIM 2013 du logiciel GASPR¹, qui propose de donner un comportement "vivant" à des entités sonores, en a été un très bon exemple.

Sur le plan technique, la version logicielle de notre système propose maintenant une implémentation, certes minimale, mais effective, des différents principes de mise en espace. Elle nous a permis de souligner l'importance, souvent négligée dans d'autres logiciels, d'une gestion temporelle efficace des événements de la mise en espace.

Par ailleurs, dans un schéma plus traditionnel de production d'un logiciel, la question de l'ergonomie des interfaces est souvent tardive, car considérée comme secondaire. Nous constatons

1. *GASPR, Graphical Audio Spatialization Program With Real Time Interactions*, Thierry Dilger in *Actes des Journées d'Informatique Musicale 2013*.

cependant que certains produits, parfois *alternatifs* ou ne bénéficiant pas d'une production commerciale, délaissent souvent l'aspect *user-friendly* d'une interface, les écartant ainsi du "grand public", au profit d'un cercle d'utilisateurs initiés. Ainsi, nous avons choisi d'investir très tôt une part importante de recherche et de développement dans la production d'une interface qui intègre un maximum d'éléments d'aide au compositeur, lui permettant ainsi d'utiliser l'outil de manière plus intuitive et plus aisée.

Dans ce projet, nous nous sommes donné comme objectif la construction d'un principe de mise en espace d'une scène sonore, selon une démarche que nous avons voulu expérimentale, en proposant une série de cinq principes fondateurs d'un système d'écriture. C'est ce qui nous a conduit à introduire une hypothèse de travail justifiant la nécessité d'intermédiaires à la composition modélisés par des comportements que l'on peut qualifier de *pré-réfléchis*. Nous avons alors défini une classe d'éléments, les paradigmes de pilotage, comme constituants d'un système faisant l'union entre un *concept* — la volonté d'une mise en espace — et un *signifiant*, la perception effective de cette mise en espace. Cette union a été par ailleurs matérialisée par un lien, une représentation à la fois iconographique et gestuelle. Selon la théorie du linguiste Ferdinand De Saussure, toutes les conditions sont réunies pour parler d'un *Signe Linguistique*, c'est à dire le lien entre « *non pas un nom et une chose, mais un concept et une image acoustique* »². Selon Saussure encore, le signe linguistique montre quatre caractéristiques différentes. La

2. *Cours de linguistique générale*, Ferdinand De Saussure, Payot 1976

première concerne le caractère *arbitraire* du signe et nous conduit à construire un système de signes, et donc à définir un corpus codifié mettant en relation une représentation avec une gestuelle unique. Un signe linguistique se caractérise aussi par son aspect *linéaire*, du fait qu'«*étant de nature auditive, [il] se déroule dans le temps* »³, apportant alors la dimension temporelle à notre principe, et consécutivement une dimension *causale* entre le concept et son "image acoustique". Enfin les deux dernières caractéristiques concernent la dimension "*synchronique et diachronique*" du signe linguistique, c'est à dire la capacité d'un ensemble de signes à pouvoir évoluer au cours du temps, tout en conservant une unité et une cohésion arbitraire, mais nécessaire et partagée.

Ces critères légitiment finalement les différents principes énoncés en introduction : la nécessité de proposer un système générateur de sens justifie l'appellation de "langage", lui conférant alors sa dimension *performative* nécessaire à l'expression artistique d'une créativité spontanée et dont la forme n'est jamais contrainte (**Principes 2 et 3**). Finalement, c'est ce besoin de liberté face à un système codifié que pose l'**hypothèse heuristique** à laquelle nous avons eu recours tout au long de ce travail et qui a justifié l'introduction des paradigmes, dont la fonction est bien de faire le lien entre l'intention artistique, le "quoi", et sa réalisation effective, le "comment" (**Principe 1**). Dans son *Cours de linguistique générale*, Ferdinand De Saussure énonce aussi un "*principe d'étude*" des signes dans lequel il pose leur définition de manière systémique, impliquant une relation de dépendance de chaque signe aux autres, de chaque

3. Ferdinand De Saussure, *op. cit.*

signe à l'ensemble. C'est finalement cette "relation aux autres" qui marque notre système (**Principes 4 et 5**) et dont se sont servi Boulez et Xenakis, chacun de leur côté, dans l'élaboration de leurs systèmes d'écriture de la musique.

Un langage étant nécessairement vivant et évolutif, il n'est pas question de faire de notre principe un principe *mort*. Il faut que chaque utilisateur puisse se l'approprier, le faire évoluer, et surtout l'adapter à son discours artistique ainsi qu'à sa vision des choses. Le cadre de référence que nous avons proposé pour les paradigmes dans le chapitre 3, page 34, permet aux utilisateurs d'en étendre la liste, comme nous l'avons fait au cours de la réalisation des prototypes, en proposant d'autres moyens de pilotage des sources sonores. Il devient alors judicieux d'adjoindre à notre système le principe de *script* pour permettre la définition de nouveaux paradigmes.

Sur un plan plus pratique, l'intégration de notre système à un environnement de travail déjà constitué a montré ses limites, notamment en ce qui concerne les problèmes de communication entre les différentes entités et surtout en imposant la répétition de différentes actions dans différentes interfaces. Finalement, notre principe de mise en espace peut servir à d'autres tâches dans la gestion d'un projet musical que la seule spatialisation sonore. L'Upic de Xenakis servait entre autres à tracer des échelles de hauteur ; il en est de même pour notre système dont le principe peut, et *doit*, s'étendre à d'autres paramètres sonores pour en élargir et compléter les potentialités.

D'autres perspectives

L'élaboration de ce travail aura finalement été l'occasion de construire un système de mise en espace sonore en faisant abstraction des solutions déjà existantes dans ce domaine. En soi, les paradigmes proposés ne sont pas nouveaux : on aura d'ailleurs beaucoup emprunté à la *Conception Assistée par Ordinateur* (CAO) car c'est un domaine qui a su fortement évoluer face à une demande et un savoir-faire en perpétuelle évolution. De la même manière, alors que le principe des dispositifs tactiles tel qu'on le connaît aujourd'hui existe depuis la fin des années quatre-vingt, il n'a été appliqué de manière forte que depuis une dizaine d'années. Mais c'est un délai plutôt courant dans les domaines de la technologie et de l'informatique. Il nous faut alors prédire quel pourrait être le prochain "tournant", la prochaine scansion dans l'élaboration d'interfaces de travail. La réponse est peut-être à trouver du côté de l'intelligence artificielle. En effet ce domaine d'étude est en plein essor depuis les années quatre-vingts et un de ses champs d'applications concerne son extension aux interfaces utilisateurs. Il peut s'agir de rechercher les moyens d'organiser différemment les éléments d'interactions à l'écran, mais aussi d'anticiper l'action suivante de l'utilisateur grâce à un *apprentissage* de la part de la machine. Notre volonté revendiquée d'avoir recours à des éléments linguistiques trouve ici une nouvelle justification, car c'est justement eux qui vont permettre à la machine de devenir intelligente, en offrant une réelle assistance à un compositeur, en comprenant sa sensibilité artistique, ses habitudes, son langage, etc. , en d'autres termes son "style". Si l'on considère que les "réflexes de création" d'un artiste sont en par-

tie guidés par des mécanismes *pré-réfléchis*, on peut alors suivre la démarche de Paul Jorion exposée dans son ouvrage *Principes des systèmes Intelligents* qui consiste à faire simuler à une machine un comportement intelligent en prenant comme point de départ les préceptes de la "métapsychologie" freudienne, et plus précisément celui d'une "*dynamique d'affect appliquée à un univers de mots*".

« On peut supposer que tout ce que nous éprouvons par nos sens, en particulier dans les modalités visuelles et auditives, même après élaboration dans le néocortex temporal, doit finalement être transmis aux structures limbiques pour conférer à l'expérience son caractère immédiat. La contribution limbique propre à ce processus pourrait être d'attribuer un rôle d'affect ou de motivation à ce percept. Cette continuité limbique est probablement la condition préalable permettant à un percept d'être vécu ou évoqué consciemment ; et cela pourrait impliquer que tout événement consciemment perçu possède une sorte de dimension affective, aussi petite soit-elle. »⁴

Alors, si la partie créative de l'intellect possède une dimension *affective*, il sera possible d'adjoindre à notre principe une fonction "*d'assistant*" , via une interface intelligente. Ainsi commence véritablement notre travail.

4. *The role of the limbic system in experimental phenomena of temporal lobe epilepsy*, P. Gloor A. Olivier A L.F. Quesney F. Andermann S. Horowitz, *Annals of Neurology* vol.12 1982.

Annexes

Annexe A

Le fichier de sauvegarde d'un projet

Dans la mesure où il est nécessaire de permettre à l'utilisateur de sauvegarder une partition afin de pouvoir la recharger ultérieurement, il nous faut définir un moyen d'enregistrer dans un fichier l'état de chaque objet. Afin de décrire chaque objet à sauvegarder, nous avons fait le choix d'une description dans le langage XML, un format rapide, ouvert et extensible dont le principe est le suivant : pour chaque description que l'on souhaite ajouter au fichier, on utilise une *balise*, c'est-à-dire un mot clef entouré par des chevrons "<" et ">" pour la balise de départ, "</" et ">" pour la balise marquant la fin de la description :

```
<Balise1>
```

```
    Une liste de paramètres
```

```
<Balise2>
```

```
    Une seconde liste de paramètres
```

```
</Balise2>
```


`</Balise1>`

Il est aussi possible d'ajouter des paramètres à l'intérieur d'une balise pour en expliciter la description :

`<Balise1 un_paramètre="une valeur" >`

 Une liste de paramètres

`</Balise1>`

Pour notre système, la liste des objets à sauvegarder est donnée plus loin, figure A.2. Voici la forme générale du fichier XML généré lors de la sauvegarde de la partition :

```

<Document name="partition3">
  <AtomPool>
    <Atom>0</Atom>
    <Atom>1</Atom>
  </AtomPool>
  <CueList>
    <Cue Id="0" Start="480" Length="0" Label="intro"/>
    <Cue Id="1" Start="3185" Length="0" Label="part1"/>
  </CueList>
  <TrackPool>
    <Track>
      <TrackId>0</TrackId>
      <ClipPool>
        <Clip R="117" G="107" B="238">
          <ClipId>0</ClipId>
          <ClipStart>600</ClipStart>
          <ClipLength>2790</ClipLength>
          <ClipParadigm>2</ClipParadigm>
          <ClipAtomPool>
            <ClipAtom>2</ClipAtom>
            <ClipAtom>0</ClipAtom>
          </ClipAtomPool>
          <ClipPoints>
            <Point>
              <X>300</X>
              <Y>450</Y>
            </Point>
            <ControlPoint>
              <X>600</X>
              <Y>450</Y>
            </ControlPoint>
          </ClipPoints>
        </Clip>
      </ClipPool>
    </Track>
  </TrackPool>
</Document>

```

FIGURE A.1 – La structure du fichier XML dans lequel est sauvegardé la partition

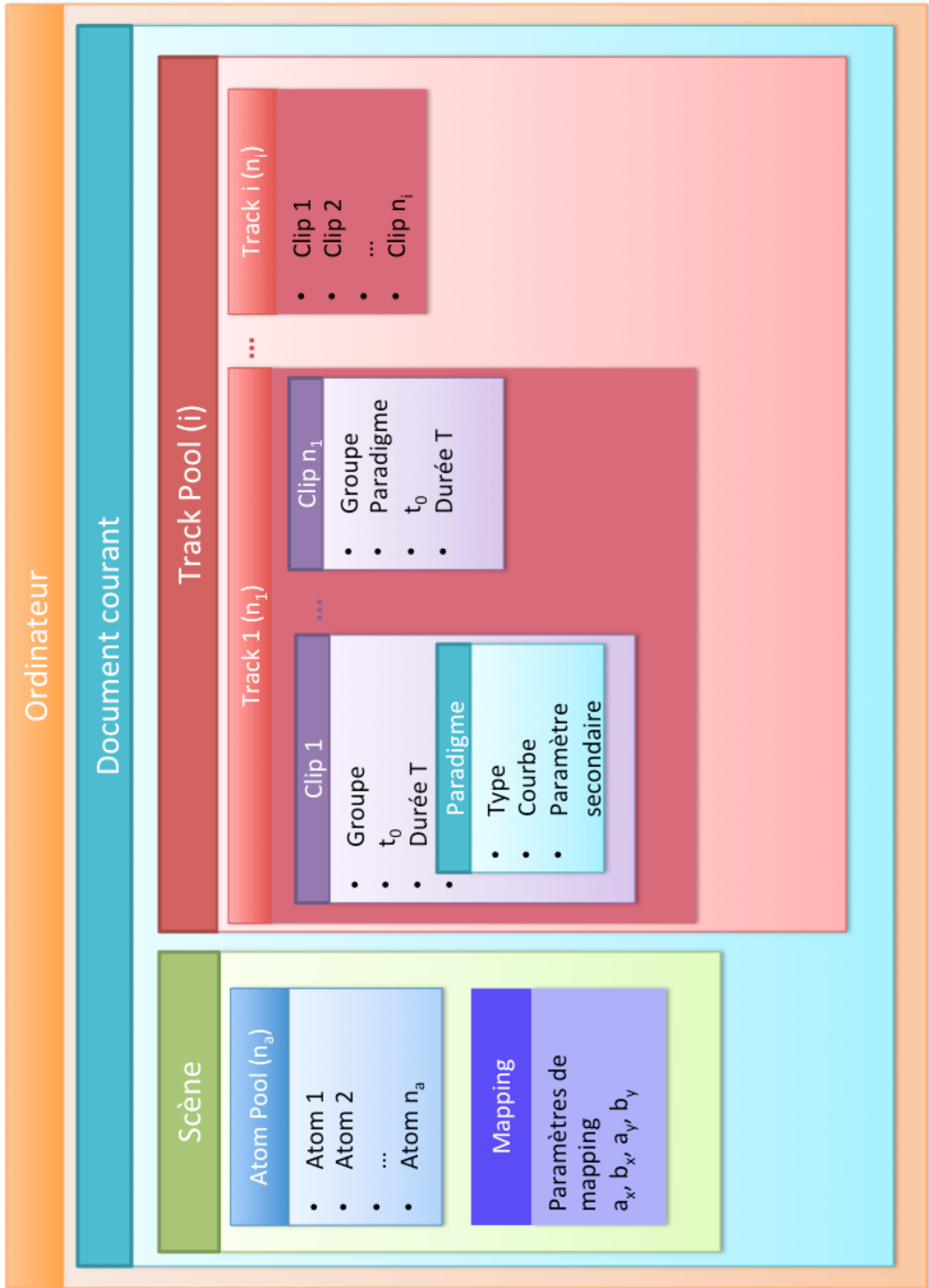


FIGURE A.2 – Synoptique de construction d'un document

Annexe B

La paramétrisation des courbes

La plupart des logiciels intégrant des graphismes vectoriels utilisent un système de courbes de Bézier. Cette famille de courbes dites polynomiales paramétriques est décrite par un ensemble de points de contrôle, n'appartenant pas forcément à la courbe, qui satisfont à la définition suivante :

Définition 7. Soient $(n+1)$ points de contrôle (P_0, P_1, \dots, P_n) . On définit une courbe de Bézier par l'ensemble des points :

$$\sum_{i=0}^n B_i^n(t) \cdot P_i$$

où B_i^n sont les polynômes de Bernstein.

D'une manière plus géométrique, cette propriété peut se résumer au fait qu'un morceau de courbe peut être décrit avec quatre points (P_0, P_1, P_2, P_3) comme le montre l'exemple suivant :

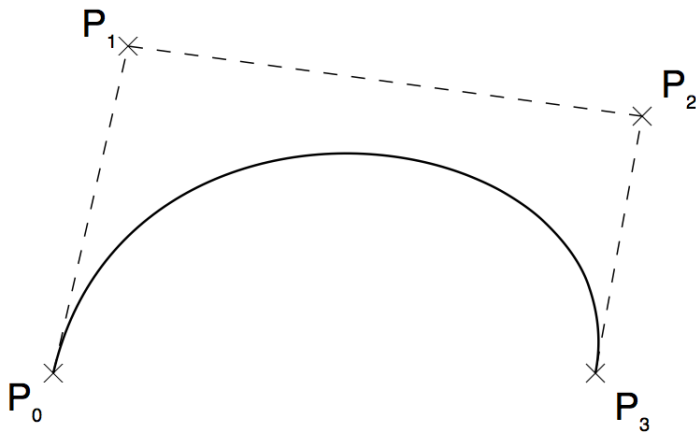


FIGURE B.1 – Une courbe de Bézier cubique

Ainsi, pour paramétrer une courbe plus complexe, il "suffit" de raccorder plusieurs courbes de Bézier cubiques, le point P_3 de la première courbe étant confondu avec le point P_0 de la courbe suivante et ainsi de suite, tout en s'assurant que la courbure est continue aux points de jonction.

Comme on peut le constater sur la figure B.1, il n'est pas du tout garanti que les points de contrôle appartiennent à la courbe que l'on souhaite décrire. Cela n'arrive d'ailleurs quasiment jamais dans la mesure où la force de ce modèle est précisément que ses points de contrôles sont à l'extérieur. Le problème est donc que lorsque l'on trace une courbe avec le doigt, on génère en fait un ensemble discret de points que l'on peut assimiler aux coordonnées des pixels de l'écran touchés par le doigt et qu'il est très difficile d'en interpoler directement un ensemble de courbes de Bézier. Il est donc nécessaire de trouver un modèle mathématique intermédiaire entre l'ensemble discret de points dessinés par l'utilisateur et le

modèle des courbes de Bézier : C'est ici que les *Splines de Catmull-Rom*¹ interviennent.

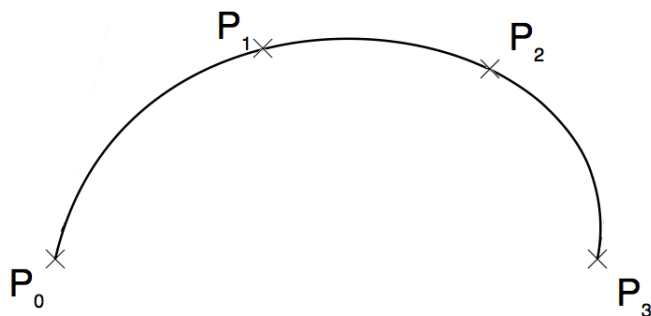


FIGURE B.2 – La même courbe, cette fois-ci avec les points de contrôle d'une Spline de Catmull-Rom

Ce modèle a pour immense intérêt pour nous d'avoir des points de contrôles sur la courbe. En partant de la liste de points créée par l'utilisateur en dessinant sur l'écran tactile, on peut donc construire la spline, puis en déduire la courbe de Bézier par la routine mathématique suivante :

1. Ce modèle porte le nom des physiciens Raphael Rom et Edwin Catmull. Ce dernier est un des fondateurs et directeur actuel du studio d'animation Pixar. Le modèle des splines de Catmull-Rom a en outre été développé à l'origine pour rendre plus lisses les courbes de déplacement des caméras virtuelles en animation graphique assistée par ordinateur.

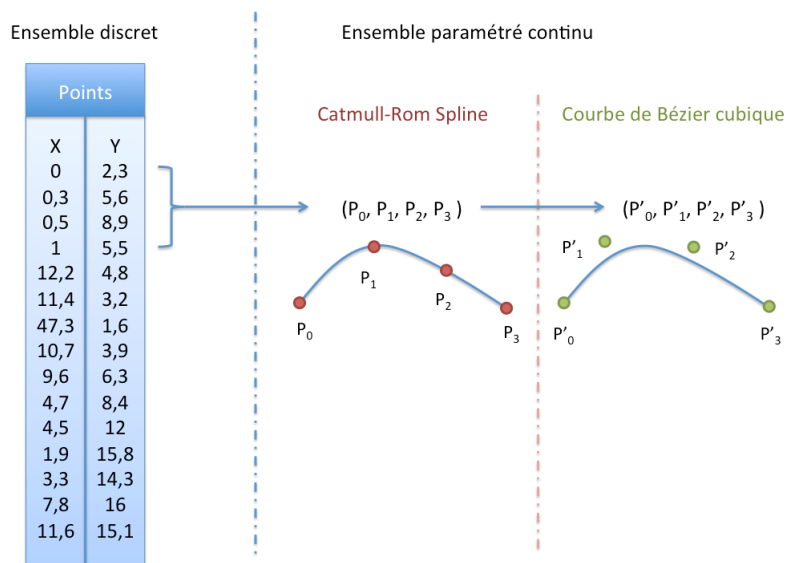


FIGURE B.3 – Etapes de l'interpolation d'une courbe

Il ne reste plus qu'à répéter ces étapes en avançant de trois le groupe de quatre points, à chaque itération.

Annexe C

Le calcul de la position d'un point sur une courbe de Bézier

Dans le but de pouvoir déplacer un atome, et donc un point de l'espace, le long d'une courbe de Bézier, il est nécessaire d'en avoir une description paramétrique. Soient quatre points P_0, P_1, P_2, P_3 définissant une courbe de Bézier cubique quelconque. On a alors la description paramétrique suivante :

$$\forall t \in [0, 1],$$
$$P(t) = P_0(1 - t)^3 + 3P_1t(1 - t)^2 + 3P_2t^2(1 - t) + P_3t^3$$

Dans notre cas, chaque courbe est en fait une série de courbes de Bézier mises "bout-à-bout". Il faut donc ajuster un peu le calcul. Nous allons généraliser le paramètre t en une autre variable T qui tiendra compte de la longueur de la courbe, ou en d'autres termes

du nombre de courbes de Bézières cubiques qui la constitue.

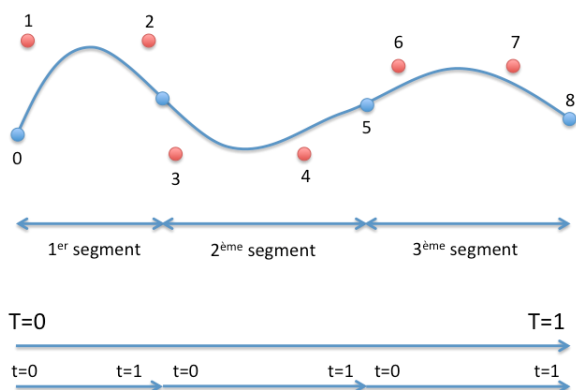


FIGURE C.1 – Un ensemble de courbes de Bézier cubiques

Voici l'algorithme de calcul de la position d'un point sur la courbe, en fonction d'un paramètre T :

```

Données : une liste de points L, un paramètre T
Sortie : un point de coordonnées x,y
Tmax= (L.count - 1)/3;
t = reste de la division de T par Tmax;
inc=arrondi inférieur de t;
if (inc * 3) + 3 < L.count then
    Point a = L.point(inc*3);
    Point b = L.point(inc*3)+1;
    Point c = L.point(inc*3)+2;
    Point d = L.point(inc*3)+3;
    décrémenter inc;
else
    | sortir de la fonction
end
resultat.x = pow(t,3) * (-a.x + 3*(b.x-c.x) +d.x) + pow(t,2)* (3*(a.x+c.x) - 6*b.x) +
3*t*(b.x-a.x) + a.x;
resultat.y = pow(t,3) * (-a.y + 3*(b.y-c.y) +d.y) + pow(t,2)* (3*(a.y + c.y) - 6*b.y) +
3*t*(b.y-a.y) + a.y;
renvoyer resultat ;

```

FIGURE C.2 – Algorithme de calcul de la position d'un point sur une courbe

Annexe D

La liaison OSC

La mise en forme, ainsi que la lecture d'un message au format OSC fait l'objet d'une spécification technique de la part du CN-MAT, disponible sur le site *www.opensoundcontrol.org*.

La syntaxe d'un message

Un message OSC peut véhiculer les types de données suivants :

- **int32** : un nombre entier codé en complément à deux sur 32 bits (big-endian).
- **OSC-timetag** : un marqueur temporel codé sur 64 bits (big-endian).
- **float32** : un nombre à virgule flottante codé sur 32 bits (big-endian) selon la norme IEEE 754.
- **OSC-string** : une chaîne de caractères ASCII qui se termine par un *null*, et dont le nombre de bits doit être un multiple de 32. Le cas échéant, on ajoute des caractères "vides" à la fin de la chaîne pour en ajuster la taille.

La taille en bits de chaque donnée est un multiple de 32, et ce dans le but de garantir un alignement constant des données, par multiples de 32.

Pour construire un paquet OSC, il faut fournir une chaîne d'adresse commençant par le caractère "/", puis une liste commençant par le caractère ",", contenant le type de chaque donnée que l'on souhaite transmettre, puis chaque donnée.

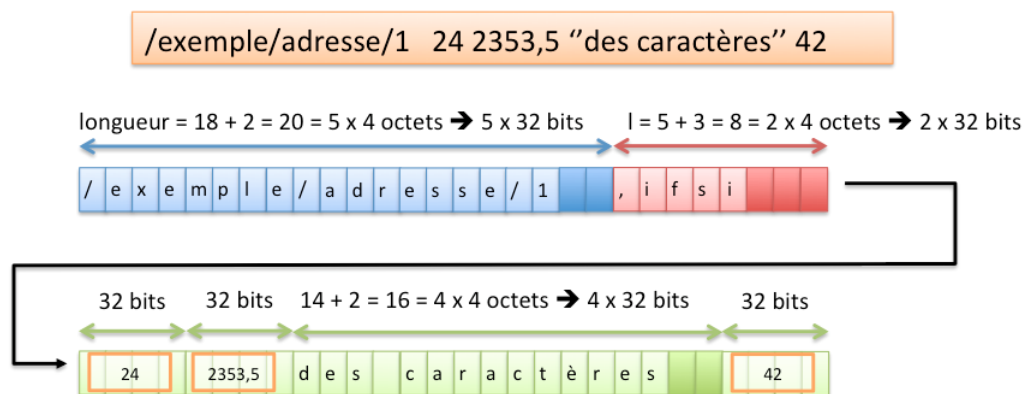


FIGURE D.1 – Un exemple de message OSC

Dans le but d'émettre et de recevoir un message OSC, le logiciel doit créer un *socket*¹ UDP par le biais duquel il communique.

1. Terme informatique désignant un élément logiciel qui gère l'interface d'une connexion réseau.

```

Données :
A : une adresse ="/exemple/adresse",
L : une liste de valeurs ={23; 353,5; "bla"}
S : un socket UDP
while A.length %4 ≠ 0 do
| A->append(0x00);
end
S->addData(A);
S->addData(',');
for i = 0, i < L.count, i ++ do
| if type de L[i]=int then
| | S->addData('i');
| end
| if type de L[i]=float then
| | S->addData('f');
| end
| if type de L[i]=string then
| | S->addData('s');
| end
end
while S->data.length %4 ≠ 0 do
| S->addData(0x00);
end
for i = 0, i < L.count, i ++ do
| S->addData(L[i]);
| if type de L[i]=string then
| | while S->data.length %4 ≠ 0 do
| | | S->addData(0x00);
| | end
| end
end
end
S->sendData;

```

FIGURE D.2 – Algorithme de mise en forme d'un message OSC

Données :
D : les données brutes reçues par le socket UDP
Sorties :
L : une liste de paramètres
A : une adresse

```

index = 0 ;
d = D->readChar(index) ;
tableau types ;
while d≠ 0x00 do
    A->append(d) ;
    index+1;
    d = D->readChar(index) ;
end
while d≠ ',' do
    index+1;
    d = D->readChar(index) ;
end
index+1 ;
while d≠ 0x00 do
    if d='i' then
        | types->append(int);
    end
    if d='f' then
        | types->append(float);
    end
    if d='s' then
        | types->append(string);
    end
    index+1;
    d = D->readChar(index) ;
end
while d= 0x00 do
    index+1;
    d = D->readChar(index) ;
end
for i = 0, i < types.count, i ++ do
    if types[i]=int then
        | value = D->readCharInterval(index,index+4);
        | L->addInt(value);
        | index+4;
    end
    if types[i]=float then
        | value = D->readCharInterval(index,index+4);
        | L->addFloat(value);
        | index+4;
    end
    if types[i]=string then
        while d= 0x00 do
            | d = D->readChar(index) ;
            | value->append(d);
            | index+1;
        end
        L->addString(value);
        index = index + index /
    end
end

```

FIGURE D.3 – Algorithme de réception d'un message OSC

Annexe E

Le protocole de communication séquenceur-tablette

Cette annexe a pour but d'établir la liste des messages OSC à établir en vue de la communication entre la partie logicielle hébergée par l'ordinateur, le séquenceur temporel, et le programme d'édition spatiale de la tablette.

Le ping

L'utilisation d'un réseau wifi peut parfois poser certains problèmes, notamment celui d'une déconnexion impromptue de la tablette. Nous avons donc ajouté au système un dispositif permettant de vérifier de façon régulière la bonne connexion entre le séquenceur et la tablette par l'envoi d'une requête "ping" à l'iPad, à laquelle il doit répondre dans un temps imparti. Le cas échéant, un message avertissant l'utilisateur d'un problème de connexion s'affiche dans

le séquenceur, mais aussi sur la tablette.

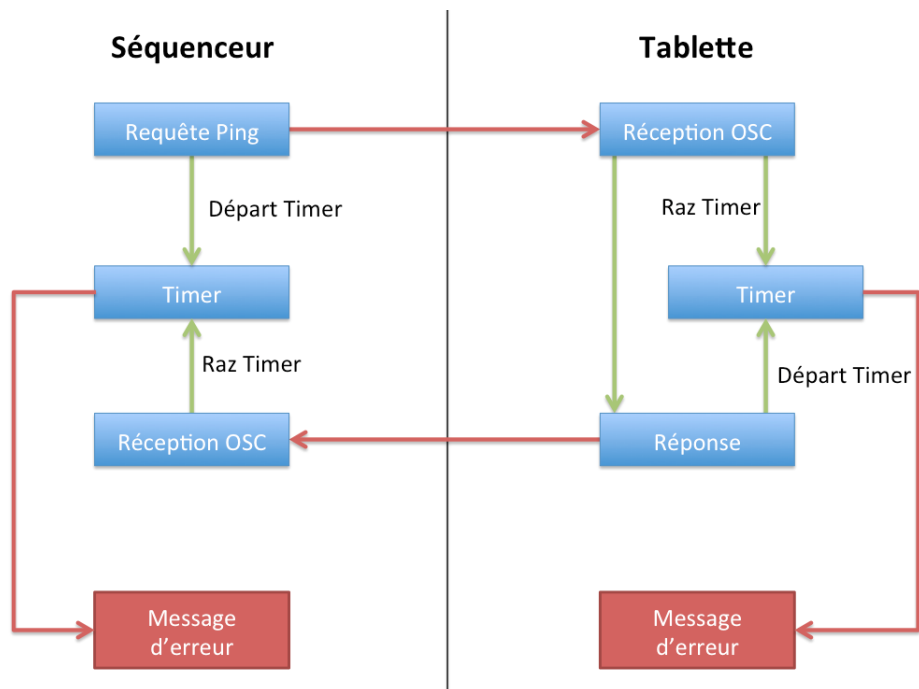


FIGURE E.1 – Schéma de contrôle de la connexion entre l'ordinateur et la tablette

Le contrôle du transport

Les deux interfaces proposent les mêmes fonction de transport dans la partition. Il est ainsi possible d'émuler les fonctions "Play/pause" ainsi que le "retour à zéro" de la tête de lecture depuis la tablette. A l'inverse, il faut aussi notifier la tablette de changements dans la barre de transport du séquenceur afin de conserver la cohérence des informations affichées dans les deux interfaces.

Chartre de contrôle en OSC

Voici la liste des messages OSC employés pour la communication entre le logiciel hébergé sur l'ordinateur et celui de la tablette. Cette

liste n'est pas exhaustive et devra être complétée par l'ajout futur de fonctionnalités.

Description	Emetteur	Adresse	Arguments
Ping	Ordinateur/Tablette	/.../ping	/
Play/pause	Ordinateur/Tablette	/.../play	1 : lecture 0 : pause
Retour à zéro	Ordinateur/Tablette	/.../raz	/
Modification d'une courbe	Ordinateur/Tablette	/.../contourChanged	int : <i>id</i> de piste int : <i>id</i> de clip liste de points : "C" si point de contrôle "P" si point float : coordonnée X float : coordonnée Y
Changement de paradigme	Ordinateur/Tablette	/.../paradigmChanged	int : <i>id</i> de piste int : <i>id</i> de clip int : <i>id</i> du type de paradigme
Editer Clip	Ordinateur	/.../editClip	int : <i>id</i> de piste int : <i>id</i> de clip
Modification de la liste d'atomes dans un groupe	Ordinateur	/.../atomListChanged	int : <i>id</i> de piste int : <i>id</i> de clip "list" liste de int : <i>id</i> de chaque atome
Position d'un atome dans l'espace	Ordinateur	/.../atomPos	int : <i>id</i> de l'atome float : coordonnée X float : coordonnée Y
Compteur temporel	Ordinateur	/.../timeCode	"HH :MIN :SEC.MS"
interpolation entre deux Cues	Tablette	/.../inter	int : <i>id</i> du marqueur 1 int : <i>id</i> du marqueur 2 float : curseur d'interpolation [0,1]

Annexe F

Les codes sources

Sur le CD-ROM fourni en annexe¹ à ce mémoire figure les fichiers sources des différents projets effectués au cours de ce travail.

On y retrouve :

- La télécommande du logiciel *Iannix* pour iPad dans le dossier "Iannix". Projet *cocoa-touch* pour Xcode, écrit en objective-C.
- Le prototype de dessin de courbes pour iPad dans le dossier "test-Curve". Projet *cocoa-touch* pour Xcode, écrit en objective-C.
- L'interface de contrôle de la trajectoire de sources pour iPad dans le dossier "hoaMap LITE". Projet *cocoa-touch* pour Xcode, écrit en objective-C.
- l'interface tactile de notre système dans le dossier "hoaMap". Projet *cocoa-touch* pour Xcode, écrit en objective-C.
- Le séquenceur temporel de notre système dans le dossier "sequenceur3". Projet Qt 5.0.1, écrit en C++.

1. L'ensemble du code est aussi accessible à l'adresse suivante : <https://github.com/manu88>

Le contenu du code source est régie par la *Licence Publique générale GNU* et peut donc être librement utilisé, modifié et redistribué.

Enfin, l'ensemble du projet comporte des parties empruntées à d'autres logiciels ou bibliothèques, qu'il convient de citer.

- La bibliothèque HOA, license GNU LGPL.
- Le logiciel IanniX, license GNU GPL.
- la bibliothèque OscPack, *open source*.
- la bibliothèque GCDAsyncUdpSocket, domaine public.

Annexe G

Spatialisation : Etat de l'art

Voici une liste non exhaustive des moteurs de spatialisations existants. La sélection donnée ici se veut représentative des grandes tendances qui existent dans ce domaine.

le Spat - Ircam

Developpé par l'Ircam, le SPAT est un outil avancé de spatialisation sonore qui combine un principe de placement de la source dans l'espace à une simulation de l'acoustique d'un lieu. Le logiciel, développé et commercialisé par la société *Flux* gère jusqu'à huit canaux d'entrées/sorties, le rendant ainsi compatible pour la plupart des formats de diffusion du son.

Music Space - Olivier Delerue

MusicSpace est une interface de contrôle de la spatialisation du son combinée à un algorithme de contraintes permettant d'établir des relations entre les différentes sources sonores présentes dans une scène. Il a d'abord été réalisé au laboratoire SONY CSL à Paris, puis intégré à l'environnement *Max*, contrôlant ainsi un ensemble de paramètres beaucoup plus importants que la simple position spatiale de sources sonores.

Spatial Audio Workstation 2 Upmix! - IO-SONO

Contrairement aux autres logiciels cités, *Spatial Audio Workstation 2 Upmix!* n'est pas un moteur de spatialisation en soi : il s'agit d'un système de pilotage pour le logiciel *Nuendo* de Steinberg. Son utilisation est très intéressante car elle permet de rester dans l'environnement de production audio de qualité qu'est *Nuendo*, tout en bénéficiant d'une interface de spatialisation dédiée.

Les DAW

Depuis plusieurs années, la plupart des stations de travail audio numériques (*DAW*) intègrent les formats audio multicanaux. Ils proposent donc tous naturellement des outils, dont la complexité peut varier d'un fabricant à l'autre, positionnement dans l'espace par *intensité*, c'est-à-dire de simples *pan-pots* étendus aux formats multicanaux.

Bibliographie

- [1] Samuel Adler. *The Study Of Orchestration*. W W Norton, 3e edition, 2002.
- [2] Ben Shneidermann Andrew Sears, Catherine Plaisant. A new era for touchscreen application : High precision, dragging icons and refined feedback. Technical report, Human-Computer Interaction Laboratory, Department of Computer Science, University of Maryland, 1990.
- [3] John Langshaw Austin. *Quand dire c'est faire*. Editions du Seuil, 1970.
- [4] François Bayle. *Musique acousmatique : propositions... positions*. Buchet/Chastel, Paris, 1993.
- [5] Hector Berlioz. *Mémoires*. C. Lévy, 1878.
- [6] Paul A. Booth. *An introduction to human-computer interaction*. Psychology Press, 1898.
- [7] Pierre Boulez. *Penser la musique aujourd'hui*. Gallimard - Editions Gonthier, 1963.
- [8] Bill Buxton. A multi-touch three dimensional touch-sensitive tablet. In *Proceedings of the 1985 Conference on Human Factors in Computer Systems, CHI '85, San Francisco*, 1985.

- [9] Bill Buxton. The natural language of interaction : A perspective on non-verbal dialogues. *INFOR : Canadian Journal of Operations Research and Information Processing*, 26, 1988.
- [10] Michel Chion. *La Musique concrète, art des sons fixés*. Entredeux/Môméludies - CFMI de Lyon, 2010.
- [11] Alan Cooper. The myth of metaphor. *Visual Basic Programmer's Journal*, June 1995.
- [12] Thierry Dilger. Gaspr, graphical audio spatialization program with real time interactions. *Actes des Journées d'Informatique Musicale*, Mai 2013.
- [13] Madeleine Arnold en collaboration avec Claude Lebrun. Représentation linguistique et spatiale d'une scène tridimensionnelle. *Sémiotiques - Centre National de la Recherche Scientifique et Institut National de la Langue Française. Paris : Didier*, 4, juin 1993.
- [14] Adrian Freed et Andy Schmeder. Features and future of open sound control version 1.1 for nime. In *NIME*, juin 2009.
- [15] Peter Gorniak et David Poole. Predicting future user actions by observing unmodified applications. Department of Computer Science University of British Columbia Vancouver, B.C., Canada.
- [16] Ingrid Zukerman et David W. Albrecht. Predictive statistical models for user modeling. *User Modeling and User-Adapted Interaction*, 11, mai 2001.
- [17] Paul Guillaume. *La psychologie de la forme*. Flammarion, 1937.

- [18] Louis Hjelmslev. *Prolégomènes à une théorie du langage*. Les Editions de minuit, 1968.
- [19] P. Gloor A. Olivier A L.F. Quesney F. Andermann S. Horowitz. *The role of the limbic system in experimental phenomena of temporal lobe epilepsy*, volume 12. *Annals of Neurology*, 1982.
- [20] Paul Jorion. *Principes des systèmes intelligents*. Editions du Croquant, 2012.
- [21] Emmanuel Kant. *Critique de la faculté de juger*. Flammarion Poche, 2000.
- [22] Bill Buxton Ken Hinckley, Michel Pahud. Direct display interaction via simultaneous pen + multi-touch input. *Society for Information Display (SID) Symposium Digest of Technical Papers*, 41, 2010.
- [23] Richard Kostelanetz. *Conversations avec John Cage*. Editions des Syrtes, 2000.
- [24] Claude Levi-Strauss. *Les structures élémentaires de la parenté*. Mouton, 1967.
- [25] Claude Levi-Strauss. *Anthropologie structurale, tome 2*. Pocket Agora, 1997.
- [26] Claire Petitmengin. La dynamique pré-réfléchie de l'expérience vécue. *Alter : revue de phénoménologie*, (18), 2010.
- [27] Craig W. Reynolds. Flocks, herds, and schools : A distributed behavioral model. *Computer Graphics*, 21(4), 1987.
- [28] Ferdinand De Saussure. *Cours de linguistique générale 1916*. Payot, 1975.

- [29] Pierre Schaeffer. *Traité des objets musicaux*. Paris Seuil, 1966.
- [30] Kurt Stone. *Music Notation in the Twentieth Century : A Practical Guidebook*. W W Norton, 1980.
- [31] Eero Tarasti. L'espace dans le discours musical. *Les cahiers de l'IRCAM*, 5, 1994.
- [32] Anne Veilt. Notation écrite et musique contemporaine : quelles grandes caractéristiques des technologies numériques d'écriture musicale. *Journées d'Informatique Musicales*, 2007.
- [33] Iannis Xenakis. *Musiques formelles*. La Revue musicale, 1963.

Table des figures

1	Extrait du conducteur de la <i>Symphonie Fantastique</i> .	16
2	L' <i>Acousmonium</i> du GRM salle Olivier Messiaen. <i>Photo :</i> <i>François Bayle</i>	17
3	Description du point P dans l'espace de spatialisation	21
4	Exemple de modélisation d'une classe en UML	22
5	Modélisation de la classe Point	23
1.1	Modélisation de la classe <i>Atom</i>	26
1.2	Une description de l'espace dans le logiciel <i>IanniX</i> .	26
2.1	Principe du point d'ancrage en C.A.O	29
2.2	Simple translation des atomes d'un groupe	30
2.3	Rotation et homothétie combinées des atomes d'un groupe	30
2.4	Modélisation de la classe <i>Group</i>	31
2.5	L'interaction des classes <i>Atom</i> et <i>Group</i>	31
2.6	Exemple de structuration des sources selon une écriture musicale standard	32
3.1	Modélisation minimale de la classe <i>Courbe</i>	38
3.2	Représentation du suivi d'une courbe par un groupe	39

3.3	Représentation d'un mouvement aléatoire au sein d'une enceinte close	42
3.4	Une implémentation des <i>boids</i> dans le logiciel <i>PureData</i>	43
3.5	Représentation du suivi d'un point par un essaim d'atomes	45
3.6	Tableau récapitulatif des caractéristiques propres à chaque paradigme	46
3.7	Combinaison d'un mouvement aléatoire avec un dé- placement global selon une trajectoire	48
3.8	Modélisation de la classe <i>Paradigm</i>	49
3.9	Synoptique intermédiaire du pilotage de plusieurs atomes	50
4.1	Exemple d'articulation de différents paradigmes . . .	53
4.2	Description du point <i>P</i> dans l'espace de spatialisation	54
4.3	Exemples d'interpolation linéaire et polynomiale . .	56
4.4	Synoptique complet du système	57
4.5	Arbre sémantique du système	58
4.6	Les trois dimensions de la représentation des données.	61
5.1	La structure d'une partition dans notre système. . .	67
5.2	Exemples d'icônes pour les paradigmes	68
5.3	Vers une notation plus complète	69
5.4	<i>Artikulation</i> , Ligeti 1958. Partition réalisée par Rai- ner Wehringer	70
6.1	la <i>Multi-touch Tablet</i> complétée par un patron de carton	74
6.2	Les éléments gérés par un système <i>single-touch</i> . . .	77

6.3	Les éléments gérés par un système <i>multi-touch</i>	77
6.4	Les éléments de gestuelle établis pour le système <i>iOS</i> d'Apple	78
7.1	représentation des marqueurs temporels sur la timeline	85
7.2	agencement linéaire des marqueurs	85
7.3	agencement des marqueurs selon un graphe	86
8.1	Synoptique général du système	89
9.1	Un aperçu du logiciel IanniX pour <i>iPad</i>	98
10.1	Les différentes opérations sur une courbe	101
10.2	La gestuelle employée pour la manipulation d'une courbe	102
11.1	La librairie HOA dans <i>Max</i>	104
11.2	L'objet <i>Max hoa.map</i> de la librairie HOA	105
11.3	La représentation de l'objet <i>hoa.map</i> sur iPad	106
11.4	Les vues "Free" et "Edit"	107
11.5	La vue secondaire de l'inspecteur	108
11.6	La barre de transport de la tablette	109
11.7	La mise en place du paradigme de Cinétique des gaz	110
11.8	L'inspecteur, complété par le choix du paradigme . .	111
11.9	L'objet <i>Max hoa.space</i> de la librairie HOA	112
11.10	La représentation de l'objet <i>hoa.space</i> sur iPad	113
11.11	L'interface de mise en espace complète présentée aux JIM 2013 pour la librairie HOA	114
11.12	La gestuelle employée pour la manipulation des atomes et de la scène	115

12.1	Détail du séquenceur : les pistes	117
12.2	Détail du séquenceur : les clips	117
12.3	Détail du séquenceur : la tête de lecture	118
12.4	Détail du séquenceur : la barre de transport	118
12.5	Détail du séquenceur : l'inspecteur	119
12.6	Vue complète du séquenceur temporel	120
13.1	L'inspecteur complété de l'interaction avec l'iPad	123
13.2	Les actions de masquage et de fermeture d'un clip	124
13.3	Une série de marqueurs temporels dans le séquenceur	124
13.4	Une série de marqueurs disposée dans un plan	125
A.1	La structure du fichier XML dans lequel est sauvegardé la partition	137
A.2	Synoptique de construction d'un document	138
B.1	Une courbe de Bézier cubique	140
B.2	La même courbe, cette fois-ci avec les points de contrôle d'une Spline de Catmull-Rom	141
B.3	Etapas de l'interpolation d'une courbe	142
C.1	Un ensemble de courbes de Bézier cubiques	144
C.2	Algorithme de calcul de la position d'un point sur une courbe	144
D.1	Un exemple de message OSC	146
D.2	Algorithme de mise en forme d'un message OSC	147
D.3	Algorithme de réception d'un message OSC	148

E.1 Schéma de contrôle de la connexion entre l'ordinateur et la tablette 150