



ÉCOLE NATIONALE SUPÉRIEURE LOUIS LUMIÈRE

Mémoire de fin d'études :
Le module d'improvisation algorithmique interactif dans
les performances de musique *Techno*

Auteur :
Nicolas NARBONI

Directeur interne :
Jean ROUCOUSE
Directeur externe :
Jérôme NIKA
Rapporteur :
Laurent MILLOT

25 août 2020

Résumé

Ce travail de mémoire a pour objectif, en s'inspirant des divers travaux sur la musique algorithmique, de réaliser un modèle théorique de ce que nous appelons un "module d'improvisation algorithmique interactif". Ce module, qui a pour visée d'interagir avec un musicien effectuant une performance musicale, se base sur des procédés issus de la musique procédurale, aléatoire et des différents logiciels qui permettent d'accompagner des musiciens de manière indépendante. Nous en réaliserons un prototype sur *Max/MSP*, intégré à *Ableton Live* via *Max for Live* qui puisse respecter un ensemble de règles musicales spécifiques à la musique *Techno*. Il nous est apparu qu'il serait plus pertinent de se concentrer sur la théorisation et la réalisation d'un "module" qui puisse être intégré dans une performance plutôt que de s'intéresser à la construction d'une performance entièrement basée sur l'algorithme, étant donné que la démarche est fondamentalement différente. Il s'agit ici de trouver des outils et d'ouvrir de nouvelles possibilités plutôt que d'explorer une manière spécifique de créer et de jouer de la musique.

Mots-clefs : musique algorithmique, techno, house, performance, musique procédurale, improvisation.

Abstract

This master's thesis, drawing from a range of works on algorithmic music, aims to make a theoretic framework for an "Interactive Algorithmic Module for Improvisation". This module, which purpose is to interact with a musician during a live music performance, will be based on process developed in procedural music, algorithmic music, and softwares such as automatic accompaniment systems that are designed to play along a musician in a somewhat independent way. We realised it could be more helpful to make a framework and a prototype for a "module" that could easily be integrated in live performances than to build a live performance entirely based on the use of algorithms. This work tries to find tools that can be used to open new musical possibilities and assist musicians, rather than exploring a specific way to create and play music.

Keywords : algorithmic music, techno, house, procedural music, live performance, improvisation.

Remerciements

Je remercie chaleureusement le personnel encadrant de l'École Nationale Supérieure Louis Lumière, plus particulièrement Jean Rouchouse, mon directeur de mémoire interne, Corsin Vogel, qui s'est occupé de la coordination des mémoires et Laurent Millot, mon rapporteur de mémoire. Je remercie les professeurs qui m'ont accompagné le long de ma scolarité notamment Pascal Spitz, Sylvain Lambinet, Alan Blum et Eric Urbain, ainsi que Nicolas Mongermont et Thierry Coduys qui m'ont donné l'envie de travailler sur des algorithmes musicaux. Je remercie aussi sincèrement Jérôme Nika, mon directeur de mémoire externe, qui m'aura à plusieurs reprises redonné confiance dans mon travail et aiguillé vers de nombreuses idées et questions.

Je remercie aussi tous les musiciens et acteurs de la scène *Techno* et *House* parisienne qui m'ont donné le goût de cette musique et l'envie d'inscrire mon mémoire dans le thème des performances, à savoir Rafael et l'équipe d'Increase the Groove, William, Lorenzo et l'équipe de Latence, Pierre, Tom, Abel, Samy, Bruno, Swan et l'équipe d'Electronic Feeling, Lucas et Alex des Groove boys Project, Ralph et l'équipe de Parallèle et enfin Gaspard, Orion et l'équipe du Bercail.

Je remercie enfin mes amis, Adrien, Pierre, Octave, Simon, Fanny, Florine, Denys, Paco, Alma, et Mathilde dont le soutien moral m'aura été fort utile, mes camarades de classe dont la gentillesse n'a eu d'égal que l'assiduité aux raclettes hivernales, et enfin mes parents et mon frère, donc les relectures fréquentes de mon travail m'auront évité de nombreux contresens.

Table des matières

1	Introduction	1
I	Partie théorique : Contexte	4
2	La musique Techno	5
2.1	L'Electronic Dance Music	6
2.2	Historique de la Techno et de la House	7
3	Les Performances de Techno	13
3.1	Les deux approches de la performance	16
3.1.1	L'approche de "contrôle"	16
3.1.2	L'approche "performative"	17
3.2	Les machines utilisées dans les performances de Techno	19
3.2.1	Les séquenceurs	19
3.2.2	Les consoles de mixage	21
3.2.3	Le matériel musical	22
3.3	Le modèle général de la performance	24
3.4	Exemples de dispositifs	25
4	La musique algorithmique	29
4.1	Les origines de la musique algorithmique	30
4.1.1	La musique algorithmique avant l'ordinateur	30
4.1.2	La musique algorithmique assistée par ordinateur	35
4.2	Les deux approches de l'algorithme	42
4.3	Les logiques algorithmiques dans la musique Techno	45
5	L'improvisation algorithmique	47
5.1	L'improvisation	48
5.2	Les logiciels d'improvisation algorithmiques	50

II	Partie théorique : Apports	53
6	Motivations	54
6.1	L'automatisation des tâches	55
6.2	Surprise, danse et performance	56
6.3	Règles et création	58
6.4	La logique modulaire	61
6.5	Objectifs	63
7	La matière musicale	65
7.1	La musique préalablement créée	67
7.2	La musique générée durant la performance	70
7.2.1	La musique générée en temps réel : l'approche transformative . .	70
7.2.2	La musique générée en temps réel : L'approche combinatoire . .	78
7.2.3	La musique générée en temps réel : L'approche générative	79
8	Interaction, adaptativité et communication	81
8.1	Communication avec le musicien	83
8.1.1	Les informations entrantes	83
8.1.2	Les informations sortantes	84
8.2	Compréhension du contexte	87
9	Les choix du module	91
9.1	Les systèmes de choix	91
9.1.1	Le hasard	92
9.1.2	Le système par liste	93
9.1.3	Les arbres décisionnels	94
9.2	Intentions et Contexte	96
9.2.1	Systèmes rule-based et case-based	96
9.3	Variables externes et mise en scène	98
9.4	L'équilibre des choix	99
10	Modèle général d'un module d'improvisation algorithmique interactif	100
10.1	Ecoute	101
10.2	Intention	103
10.3	Traduction	105
10.4	Génération	108
11	Constitution d'un corpus de règles	109
11.1	Définition des différentes règles	109

11.2 Les corpus de règles	111
11.2.1 Le corpus de règles rythmiques	111
11.2.2 Le corpus de règles tonales, harmoniques et mélodiques	115
11.2.3 Le corpus de règles sonores	118
11.2.4 Le corpus de règles narratives	119
12 Conclusion	123
13 Bibliographie	128

Chapitre 1

Introduction

Les performances de musique électronique, et plus particulièrement de musique *Techno*, sont souvent exécutées par un nombre de musiciens réduit. La *Techno* se caractérise par des critères musicaux précis, notamment de nombreux traitements sonores sophistiqués et un grand nombre d'éléments. Ces critères amènent les musiciens à prévoir de nombreuses choses à l'avance en raison de la difficulté de la gestion des différentes machines et des nombreux éléments rythmiques et harmoniques à manipuler.

Les musiciens effectuant des performances sont souvent programmés aux côtés de *Disc Jockeys*¹ passant des morceaux mixés et le plus souvent masterisés² au préalable. Il y a donc un besoin important de fournir une performance se rapprochant le plus possible dans sa sonorité et sa complexité d'un morceau qui a pu être produit à l'avance. En conséquence, les musiciens exécutent souvent les mêmes performances, malgré quelques légères différences. Le travail de préparation d'un changement dans la performance est long et fastidieux.

1. Les *Disc Jockeys*, ou *DJs*, sont des musiciens qui jouent des morceaux entiers les uns à la suite des autres dans des événements musicaux.

2. Le terme "masterisé" renvoie à l'étape de mastering. Il s'agit de l'étape de la production musicale où l'on applique des effets sonores sur le morceau lui-même. Cette étape consiste principalement en une compression et une égalisation audio. Dans le cas de la *Techno*, elle conduit souvent à une baisse de la dynamique et une augmentation du niveau sonore du morceau.

Les musiciens peuvent en partie composer à la volée en utilisant divers outils et instruments. Mais ils sont relativement limités dans leurs possibilités créatives durant la performance, en raison du grand nombre d'éléments à manipuler.

Nous avons observé qu'une partie de ces éléments suivent souvent les mêmes logiques pour coller à des impératifs musicaux liés au style, et donc que certaines manipulations et logiques de compositions se retrouvent de manière quasiment systématique dans toutes les performances. Nous nous sommes ainsi demandés s'il y avait un moyen d'automatiser ces manipulations.

Il existe une branche entière de la musique informatique qui propose différentes manières de palier à ces problèmes et d'offrir des pistes créatives pour enrichir les performances : la musique algorithmique. La grande variété de travaux qui ont été effectués à son sujet, qu'il s'agisse du domaine de la musique expérimentale ou de celui de l'accompagnement musical, fournissent en effet un grand champ théorique et de nombreuses façons de répondre à des besoins simples d'assistantat dans la composition et la performance musicale.

Il semble intéressant de s'inspirer de ces travaux pour tenter de trouver des solutions simples permettant de soulager le musicien d'une partie du travail tout en apportant une part d'imprévu dans la performance. La solution que nous proposons ici prend la forme d'un programme que nous avons nommé "Module d'improvisation algorithmique interactif"

Nous développerons ainsi dans ce mémoire, à travers un inventaire et un historique des différentes techniques de composition musicale en temps réel, issues de la musique adaptative, interactive et algorithmique, l'idée d'un élément simple que l'on pourrait intégrer dans une performance.

Cet élément aurait en charge un ou plusieurs instruments ou parties rythmique, et pourrait s'en occuper de manière autonome en interaction avec le musicien.

L'intérêt est double : d'une part, cela permet de libérer l'esprit du musicien qui peut se concentrer sur d'autres tâches, et de l'autre, cela offre un aspect unique à chaque représentation. Le fait d'être épaulé et d'avoir des garanties de rendu musical dans la performance permet plus de liberté et nécessite moins de préparation préalable. Cela permet aussi d'ajouter une part non négligeable d'imprévu et d'offrir des possibilités créatives en temps réel.

Nous commencerons par contextualiser ce travail vis-à-vis de la musique algorithmique et des performance de musique *Techno* dans les chapitres 2 à 5. Nous exposerons ensuite plus en détails nos motivations dans le chapitre 6, avant de réaliser un modèle théorique et un inventaire des outils et méthodes que l'on peut utiliser dans les chapitres 7 à 11, pour enfin présenter plusieurs prototypes de modules, leur réalisation, et leur mise en œuvre dans une performance dans les chapitres 13 et 14.

Première partie

Partie théorique : Contexte

Chapitre 2

La musique Techno

Le terme *Techno* au sens large tel qu'il est utilisé de nos jours par les médias et les chercheurs regroupe en pratique deux sous-familles très liées l'une à l'autre : la *Techno* et la *House*. Elles sont toutes deux rattachées à une famille plus large apparue dans les années 70 que l'on nomme l'*Electronic Dance Music*.

L'esthétique entière de ces musiques est basée sur l'utilisation intensive de machines électroniques diverses, ce qui leur forge un lien très fort avec l'informatique musicale. Les musiciens qui en composent utilisent depuis longtemps certains procédés algorithmiques de composition. Ces procédés, à l'origine cantonnés aux machines électroniques utilisées dans la composition, sont devenus d'autant plus courants et riches depuis le début des années 2000 avec le développement de l'informatique musicale chez les particuliers. On notera qu'il est complexe, étant donné la nature très vivante de ces familles de musique, de définir précisément les contours d'un genre ou d'un autre, tant ces genres se créent et se combinent à une fréquence élevée [1]. Nous tenterons tout de même de définir les contours des genres dont nous traiterons ici à l'aide d'une approche historique combinée à l'usage général actuels des termes que *Techno* et *House*. Nous présenterons ainsi une synthèse de différentes recherches sur l'histoire de ces musiques [2][3][4].

2.1 L'Electronic Dance Music

Si nous nous intéressons spécifiquement dans ce mémoire à la *Techno*, il nous paraît pertinent de revenir sur la famille plus large qui les englobe, l'*Electronic Dance Music*, ou *EDM*. L'*EDM* désigne ainsi tous les genres musicaux majoritairement composés à l'aide d'ordinateurs et/ou d'instruments électroniques et réalisés avec pour but final d'être dansés. Le pionnier de cette famille de musique est généralement considéré comme étant le groupe allemand Kraftwerk. Ses membres intègrent dès les années 70 des instruments électroniques tels que des boîtes à rythmes ou des synthétiseurs à des compositions se rapprochant de la musique *pop*. Ils popularisent alors l'usage de ces instruments.

Cette famille comporte une forte dimension historique : les genres qui s'y rattachent sont souvent, à l'origine, associés au mouvement des *raves* et à celui des discothèques. Ces genres, aussi divers soient-ils, sont liés les uns aux autres par divers liens de parenté et liens géographiques. Leur racines se mélangent et chaque genre paraît issu d'un autre, au point que l'on peut facilement établir un arbre généalogique aux branches très imbriquées qui permet de rattacher un genre à cette famille.

Par exemple, si le *Reggaeton*¹ moderne est souvent réalisé avec des instruments électroniques, et qu'il est joué dans le but d'être dansé, il ne partage pas la même filiation d'origine que la *Techno*, la *House*, le *Dubstep*, la *Big Room*, etc... et n'est donc pas rattaché à la famille de l'*EDM*.

Il s'agit aussi de faire une distinction entre les musiques électroniques que l'on peut qualifier de populaires (la *Techno*, la *House*, etc...) et celles que l'on pourra qualifier de savantes ou expérimentales (la musique électro-acoustique par exemple).

1. Le *Reggaeton* est un style de musique originaire d'Amérique latine qui mélange des influences *Hip Hop* et latino-américaines.

Cette distinction permet de marquer que la plupart des codes musicaux de l'*EDM* se rapportent à des habitudes de consommation musicale, des contextes d'écoutes et des codes sociaux festifs, à contrario de la musique savante moderne dont les œuvres s'inscrivent dans des contextes plus particuliers.

Si cette famille rassemble des genres composés à l'origine pour un contexte d'écoute orienté vers la danse, tous ne le sont pas nécessairement. Ils restent néanmoins rattachés à cette dernière par un lien de parenté musicale. On pourra citer à cet égard l'*IDM* (pour *Intelligent Dance Music*), une branche plus expérimentale de l'*EDM* dont la complexité rend difficile son utilisation dans un contexte festif et orienté vers la danse. Enfin si les genres qui la composent sont très variés, ils sont souvent pensés d'une manière similaire. De nombreuses idées évoquées dans ce mémoire pourront facilement être appliquées dans d'autres genres de cette famille, comme les genres de la sous-famille de la *Bass Music* (*Dubstep, Drum and Bass, Jungle...*) ou ceux de la sous-famille *Trance* (*Psytrance, Trance, Tribe...*). L'auteur de ce mémoire reste cependant plus familier du genre *Techno*, ce qui motive le choix de se restreindre à cette famille.

2.2 Historique de la Techno et de la House

La *House* apparaît au début des années 80 dans la ville de Chicago. Les années 70 sont marquées aux Etats-Unis par l'avènement de la musique *Disco* dans les discothèques. Ce style de musique, à l'origine circonscrit aux communautés afro-américaines et "LGBTQIA+"², devient très populaire et perd ses racines communautaires en se démocratisant. C'est ainsi que certains membres de ces communautés décident de

2. "Lesbian, Gay, Bi, Trans, Queer, Intersex, Asexuals and more". Si le mot que l'on retrouve le plus souvent dans la littérature de l'époque est "LGBT", le terme "LGBTQIA+" est aujourd'hui utilisé pour inclure plus de personnes.

revenir à un mouvement plus "underground"³ et créent la *House*, un genre musical à contre-courant de la *Disco* dans laquelle les ensembles musicaux sont de plus en plus fournis en musiciens et les performances de plus en plus extravagantes.

Frankie Knuckles et Marshall Jefferson, deux *DJs* de Chicago que l'on considère comme les pères fondateurs de la *House*, prennent l'habitude au début des années 80 d'utiliser des boîtes à rythmes par dessus les titres de *Disco* qu'ils passent sur des platines, et donnent naissance à la *House*, que l'on nomme ainsi en référence au club dans lequel ils jouent le plus souvent, le *WareHouse*. Cette origine musicale est d'ailleurs à rapprocher du *Hip Hop*, qui se développe à la fin des années 70 à New York, et qui naît de la mise en boucle des "breaks" de batterie des disques de disco par des *DJs*.

À partir de l'année 1983, des musiciens se mettent à enregistrer eux-même leurs propres disques de *House*, en combinant des "samples"⁴ issus de disques de *Disco* à des sons issus de boîtes à rythmes et de synthétiseurs.

Ces premiers titres, dont le tout premier est généralement considéré comme étant "On and On" de Jesse Saunders (1983) partagent les codes originaux de la musique *Disco* (faite pour danser, avec un rythme typique dit "four on the floor", etc.), mais beaucoup plus électroniques, légèrement plus rapides, et avec une esthétique musicale simplifiée qui laisse de côté les breaks⁵ complexes et les orchestrations riches de l'âge d'or de la *Disco*. Les titres sont souvent réalisés en "home-studio"⁶. Les années 80 représentent le moment du développement des disques vinyles maxi 45 tours sur lesquels sont pressés des singles de *Disco*. Les ingénieurs du son de l'époque sont alors invités à effectuer des "remix"⁷ des titres pour occuper la face B du disque. Ces

3. Le mot "underground", littéralement "sous-terrain", désigne une musique qui n'est pas populaire ni répandue, par opposition à une musique "mainstream", c'est à dire populaire.

4. Un "sample" est un échantillon audio.

5. Un break est une partie d'un morceau qui marque une rupture dans la progression.

6. Un "home-studio" est un studio d'enregistrement et de production musicale réalisé chez un particulier.

7. Un "remix" est une version retravaillée d'un morceau.

"remix", dont la sonorité s'inscrit dans une esthétique *House* contribuent à populariser la *House* chez les auditeurs de *Disco*. La *House* se répand rapidement dans les clubs des Etats-Unis, et donne naissance dans la fin des années 80 à la *Deep-House*, qui se base sur un travail plus précis de l'aspect mélodique et harmonique, permis par l'amélioration des technologies de séquenceurs et de synthétiseurs. Le terme *Techno*, s'il est souvent utilisé de manière impropre pour désigner les bandes musicales jouées dans les *raves* parties, désigne plus précisément un genre de musique qui apparaît au milieu des années 80 à Détroit [5]. Cette musique se développe dans les communautés afro-américaines vivant dans un contexte économique et social difficile (Détroit, surnommée la "Motor City" en référence aux nombreuses industries automobiles qui y étaient installées, fait face à un déclin rapide de son industrie, ce qui cause un taux de chômage et de pauvreté élevé), et certains musiciens comme Kevin Saunderson ou Derrick May posent ainsi les bases de la musique *Techno* à proprement parler en développant l'*Acid House*. L'esthétique se démarque de la *House* par un aspect plus sombre, plus brut, et des emprunts à la science-fiction dans la scénographie, les visuels ou encore les noms des titres. Vers l'année 1988, certains labels changent le nom sous lequel ce style de musique est commercialisé pour le différencier de la *House* et pour éviter le terme *Acid* qui renvoie à l'imaginaire américain négatif sur l'usage des drogues dans les *raves*, et commercialisent les premiers titres officiellement rattachés à la *Techno*.⁸

Vers la fin des années 80 la *Techno* arrive à Berlin après la chute du mur, et est réappropriée par la jeunesse berlinoise qui utilise les nombreux bâtiments en ruine pour organiser des *raves*. Elle adopte alors une esthétique plus sombre et donne naissance à un style de *Techno* encore plus agressif et rapide, la *Techno industrielle*.

8. On notera cependant que le terme *Acid House* existe toujours et sert à désigner précisément les morceaux de *Techno* utilisant le synthétiseur de basse TB-303.

En parallèle, le Royaume-Uni voit naître les *raves party* en 1988, dans un contexte où les fêtes sont fortement réprimées par le gouvernement. Les soirées doivent se terminer à 2h du matin, et s'organisent ainsi, pour continuer la fête, des *raves party*, c'est à dire des rassemblements illégaux dans des prairies, des hangars et des usines désaffectées. Le gouvernement britannique durcit rapidement les lois pour lutter contre le phénomène, qui s'exporte alors partout en Europe, notamment en France vers 1991, ce qui fait découvrir la *Techno* au public français. La *Techno* se développe ainsi partout dans le monde occidental en combinant des influences très variées. Certains compositeurs de *Techno*, comme Richie Hawtin ou The Orb, citent des références très expérimentales, se réclamant de l'esthétique répétitive et hypnotique d'artistes de musique contemporaine comme Steve Reich ou Philip Glass. D'autres, comme les artistes du collectif américain Underground Resistance, se réclament plutôt d'un héritage jazz et de rythmiques afro-américaines.

Si la *Techno* et la *House* peuvent être différenciées en partie par leur histoire et leurs codes musicaux, un des problèmes récurrents dans les écrits sur les genres relevant de l'*Electronic Dance Music* est le fait que les frontières entre les genres sont floues. Les distinctions relèvent parfois de la distinction sociale, étant donné que les deux genres continuent à coexister. La *House* est ainsi rarement vue comme une simple parente de la *Techno*, et le classement d'un morceau ou d'un artiste dans un genre ou l'autre dépend fortement du contexte géographique et social dont il est issu. La *House* est ainsi souvent vue comme moins revendicatrice que la *Techno*, ayant été moins associée au phénomène des *raves parties* et plus associée aux clubs et donc à une forme légale de marchandisation de la musique. La *Techno*, elle, a rapidement été associée à l'explosion de la scène *rave* en Europe, et des fêtes illégales aux États-Unis dans un Détroit en friche. Cette distinction reste présente en France, où la *House*

à souvent été vue comme moins "underground" que la *Techno*. Ce phénomène tend en revanche à s'inverser depuis quelques années, les événements se revendiquant de la *Techno* devenant majoritaires dans la scène française d'*EDM*, rassemblant des dizaines de milliers de festivaliers à des prix parfois prohibitifs, qui restreignent le public à une certaine classe sociale aisée⁹. Comme ces deux genres musicaux appartiennent aux ramifications de ce qu'on appelle la culture *club* et *rave*, le fait de choisir de se ranger dans un style ou dans un autre peut renvoyer plus aux critères sociaux d'appartenance à un groupe qu'à un réel choix musical, ce qui contribue à rendre les contours effectifs de chaque genre musical difficiles à tracer. Cet aspect social est à la source d'un changement permanent de termes utilisés pour désigner des genres, et nous choisissons ici d'utiliser le terme *Techno* au sens très large pour simplifier la compréhension. On notera cependant que des distinctions musicales reviennent souvent, qu'il s'agisse d'une fourchette de tempos, d'une manière de poser le rythme, ou encore des sonorités des instruments utilisés. Les morceaux situés à la frontière entre les deux sont d'ailleurs rattachés à un genre hybride nommé *Tech-House*. Une grande part des codes musicaux de la musique *Techno* trouvent leur source dans le rapport important de cette musique à la danse. Elle est décrite souvent comme une musique qui existe avant tout pour être jouée dans des *raves* et pour qu'un public puisse danser dessus. Ce lien étroit se retrouve dans le fait qu'elle ait pu créer diverses sous-cultures autour de la danse.

On peut penser au *voguing*, une danse nommée en rapport au magazine américain *Vogue*. Ce style de danse, qui est lié à une culture à part entière, est né aux États-Unis, parmi des danseurs et danseuses issus de la communauté "LGBTQIA+" . Cer-

9. Il n'existe pas de chiffres précis concernant ce phénomène, mais un rapide coup d'oeil au site *Resident Advisor* qui rassemble les événements musicaux de la scène française permet de rendre compte de la sur-représentation du genre *Techno* dans les événements les plus chers et les plus pourvus en public.

taines soirées et communautés sont construites entièrement autour du *voguing*, les danseurs se rassemblant en groupes nommés *Houses* et s'affrontant dans diverses compétitions nommées *Bowl*.

On notera aussi que la scénographie des *raves-parties* a fortement influencé la vision de la *Techno* comme faite pour être dansée en communion. Les *DJs* y sont souvent placés à côté des enceintes (et non pas au centre du système de diffusion comme dans un concert de musique actuelle), voire parfois au centre du public. Le système de diffusion musicale est ainsi presque plus mis en scène que les musiciens eux-mêmes, et il n'est pas attendu du public qu'il fasse face à la scène.

Cette vision est en rupture avec les concerts plus traditionnels de musiques actuelles, où le groupe de musiciens est mis en avant sur une estrade et le public censé lui faire face. L'idée se retrouve dans la scénographie lumineuse mise en place lors des événements *Techno* : si la scène est éclairée, une attention particulière est prêtée à la mise en scène de la piste de danse, souvent parcourue de lasers et inondée de fumée.

Chapitre 3

Les Performances de Techno

La musique *Techno* est, depuis ses débuts, en grande partie composée dans le but d'être écoutée lors d'évènements musicaux. Cela se retrouve d'ailleurs dans les structures des morceaux qui sont pensés pour qu'un *DJ* puisse facilement les jouer. On peut dire que la *House* est née de l'introduction de la performance dans un *DJ Set de Disco*. Avant même que les premiers disques de *House* soient gravés, la *House* était déjà "performée" en quelque sorte par les *DJs* jouant avec des boîtes à rythme par dessus les disques. La légende veut, d'ailleurs, que le premier disque de *House* ait été réalisé car Jesse Saunders s'était fait voler un disque qu'il avait pour habitude de jouer et avait gardé des "samples" dans un échantillonneur.

On peut aussi voir la performance dans la musique *Techno* comme un déplacement du "Home-Studio" du compositeur sur scène, qui amène ses machines.

Certains *DJs* de *House* des années 90, comme le duo Master at Work, adoptent une approche de la performance qui préfigure d'une certaine façon les concerts de musique actuelle d'aujourd'hui. Ils passent les versions instrumentales de disques et en invitent les chanteurs et chanteuses à venir chanter les paroles originales par dessus.

La scène techno est souvent partagée entre deux types de représentation musicale : Le *DJing*, et la performance. On les distingue ainsi, souvent, dans les programmes distribués au public, en *DJ set* et "Live set". On distingue donc la pratique du *DJing* dans laquelle divers morceaux entiers sont joués, et où le *DJ* applique des effets et s'occupe des transitions de l'un à l'autre, et la performance, dans laquelle le musicien utilise des machines ou des logiciels pour jouer divers éléments musicaux qu'il combine ensemble.

On notera que cette distinction n'est pas absolue : Certains *DJs*, comme vu précédemment, s'accompagnent de machines (il n'est pas rare de voir des *DJs* avec des boîtes à rythmes) et certaines pratiques du *DJing* peuvent relever de la performance (on pourra citer l'exemple du scratch, une pratique instrumentale à l'aide de platines issue du *Hip Hop*, peu utilisée dans la *Techno*, mais parfois présente en *House*).

Il nous semble cependant important d'effectuer cette distinction car l'approche de la représentation musicale par un *DJ*¹ rend compliquée l'introduction de modules musicaux et l'interactivité avec ceux-ci. Le *DJ* jouant des morceaux entiers, il ne dispose pas vraiment de moyens de réagir aux actions d'un système interactif autrement qu'en changeant le morceau joué, ce qui demande du temps.

Enfin, si les *DJ sets* ont longtemps été mis en avant en priorité par rapport aux performances, pour des raisons pratiques (les platines et la table de mixage des discothèques ou des festivals peuvent servir à tous les *DJs* d'un évènement sans changement de plateau), on assiste en France, ces dernières années, avec l'augmentation de l'accessibilité des machines dédiées à la performance et l'augmentation de la puissance de calcul des ordinateurs portables, à un regain de popularité de la performance dans les scènes *Techno* et *House*.

1. La plupart des platines utilisées par les *DJs* affichent l'intégralité de la forme d'onde du morceau pour rendre compte de l'avancée de sa lecture.

Ces performances peuvent être vues comme les véritables concerts de *Techno*, ce qui se retrouve souvent dans la mise en scène des musiciens qui l'exécutent. S'il est courant que les *DJs* se succèdent simplement en effectuant une transition musicale de l'un à l'autre (le *DJ* entrant effectuant une transition avec un morceau de son choix sur le dernier morceau du *DJ* set du *DJ* sortant), les performances commencent généralement par un silence, et se terminent généralement par un silence.

Il nous faut enfin insister sur un problème récurrent des performances de musiques électroniques : le besoin de réduire le plus possible la puissance de calcul nécessaire. D'une part, car les ordinateurs portables utilisés lors des performances ont des puissances de calcul limitées. D'autre part, car il est crucial de réduire le plus possible la latence, c'est-à-dire le temps entre l'entrée et la sortie du signal, qu'il soit audio ou MIDI². Il paraît important que les actions du musicien (appuyer sur un bouton ou jouer une note) aient des effets immédiats. Or, cette latence dépend de la taille du tampon que le processeur utilise pour effectuer ses calculs : plus la taille est réduite, plus la latence est basse, mais plus il est difficile pour le processeur d'effectuer les calculs dans le temps imparti. Cet enjeu de puissance de calcul, combiné au problème de la synchronisation des différentes machines, est au cœur de la mise en place d'un dispositif de performance *Techno*. Une des spécificités de ces performances est aussi qu'elles sont réalisées le plus souvent par des musiciens ayant une formation de producteur³ et de *DJ*. Cela se retrouve dans la volonté de mettre la table de mixage sur scène, et l'étape de mixage entre les mains des musiciens. Il s'agit d'une rupture importante avec le concert traditionnel pour lequel c'est le plus souvent un ingénieur du son qui s'occupe du mixage des différents instruments.

2. Le MIDI, ou Musical Instrument Digital Interface, est un protocole de communication créé pour la musique. Il permet l'échange d'informations entre différents instruments, séquenceurs et logiciels.

3. Nous entendons ici par "producteur" un musicien qui utilise des machines et des logiciels pour composer et jouer de la musique.

3.1 Les deux approches de la performance

Nous distinguons donc ici deux approches de la performance pour envisager la réalisation d'un module. Il ne s'agit pas de deux approches s'excluant mutuellement, mais de deux extrêmes formant entre eux un spectre sur lequel la plupart des performances peuvent être placées.

3.1.1 L'approche de "contrôle"

L'approche de "contrôle" est une manière de concevoir la performance "par le haut", où l'on se concentre sur le contrôle, le lancement, le mixage et l'application d'effets. L'exemple le plus typique de l'approche contrôle est le *DJing*, où l'on joue des morceaux complets.

Mais cette approche se décline de plusieurs manières. Nous avons par exemple, dans le cadre de ce mémoire, interrogé plusieurs musiciens sur la manière dont ils conçoivent leurs performances. Une des façons de voir cette approche revient souvent surtout dans le cadre de la *House*. Il s'agit de préparer à l'avance de nombreux "clips"⁴ audios et MIDI qui vont être déclenchés les uns après les autres et mixés en temps réel. Ce besoin de préparation s'explique par la complexité musicale que le genre exige de nos jours, et par l'utilisation très répandue de "samples" venus d'autres morceaux qui demandent un important travail sonore. Cette grande préparation préalable à pour avantage, d'une part, de libérer une part du travail à effectuer en temps réel et, d'autre part, de permettre au musicien de jouer d'un instrument pour agrémenter la performance de solos ou de mélodies jouées en temps réels (un aspect que nous rangerons dans l'approche performative).

4. Un clip est un morceau d'audio ou de MIDI. Il peut contenir des données en rapport avec son contenu.

Elle est aussi motivée par un raisonnement simple : si les machines et les logiciels peuvent être vus comme des instruments, leur jeu ne comporte pas le même taux d'incertitude que celui des instruments acoustiques, numériques et électroniques. Composer une séquence déjà prévue à l'avance sur un séquenceur en temps réel produira exactement le résultat attendu, et l'intérêt de refaire ce travail soi-même sur scène est limité. Il paraît ainsi logique de réserver l'exécution en temps réel à des actions qui ne sont pas complètement prévues à l'avance, ou ne dépendent pas du jeu du musicien qui les exécute.

Cette préparation permet de jouer de la musique très "produite", c'est-à-dire avec de nombreux instruments et une complexité qu'il serait difficile de reproduire en temps réel.

3.1.2 L'approche "performative"

L'approche performative est une manière de concevoir la performance "par le bas", où chaque motif musical ou rythmique est joué ou composé en direct. L'exemple le plus absolu est celui des groupes comme Playin' 4 the city, un groupe français du début des années 2000 toujours actif aujourd'hui, ou *Secret Value Orchestra* un autre groupe français des années 2010. Les membres du groupe jouent chaque instrument comme dans une performance "traditionnelle" de musique actuelle. On peut aussi penser au collectif précédemment évoqué Underground Résistance, dont les performances réunissaient parfois des dizaines de musiciens sur scène.

Le nombre de musiciens reste cependant souvent réduit, car la *House* et la *Techno* amènent souvent à développer une approche individuelle de la composition. Il est ainsi fréquent de voir des performances en vue desquelles ces motifs musicaux sont enregistrés puis mis en boucle, afin d'ajouter ces boucles et de libérer le musi-

ciens pour qu'il puisse jouer d'autres motifs. C'est le cas de musiciens comme DubFX, un artiste qui utilise sa voix et des pédales de "loop" pour construire des morceaux complexes en additionnant les prises successives. Certains vont même jusqu'à intégrer des logiques issues de la musique concrète, à l'instar de Jacques, un musicien français, qui effectue des performances en utilisant des objets que le public lui apporte pour en tirer des bruits.

D'autres musiciens, comme Kink, un producteur de *Techno* et de *House* bulgare, composent des motifs mélodiques à la volée sur des séquenceurs, proposant parfois au public d'intervenir (une vidéo d'une performance pour *Boiler Room*⁵ montre par exemple Kink en train d'inviter un membre du public à appuyer sur les touches du séquenceur pour composer un motif mélodique). Il est possible de composer à la volée des séquences rythmiques (sur des boîtes à rythmes, comme le font de nombreux musiciens) ou des séquences mélodiques (en utilisant des arpégiateurs ou des séquenceurs).

Cette approche a pour avantage d'être bien plus lisible par le public : il est plus simple de comprendre ce que font les musiciens lorsqu'ils exécutent un motif musical en temps réel, que lorsqu'ils contrôlent la performance d'une manière plus générale. Elle permet aussi plus d'improvisation : les musiciens peuvent ainsi jouer des motifs musicaux différents à chaque performance. Elle demande en revanche bien plus de concentration et de rigueur. En outre, quand les motifs sont enregistrés en direct pour être mis en boucle, il devient crucial de jouer le "bon motif", ce qui ne laisse que peu de place à l'erreur (contrairement au solo au cours duquel un motif osé peut être joué momentanément).

5. Boiler room est un évènement qui consiste à filmer des *DJs* et des performances devant un public.

3.2 Les machines utilisées dans les performances de Techno

On distinguera plusieurs types de machines utilisées dans les performances de musique *Techno* selon leurs fonctions. Certaines machines combinent en outre plusieurs fonctions. On effectue aussi souvent la distinction entre les machines et les instruments "hardware", qui sont des machines dédiées, et les instruments et séquenceurs "software", qui fonctionnent sur un ordinateur (et plus précisément dans un "DAW" ⁶).

3.2.1 Les séquenceurs

Il y a en premier lieu le séquenceur, qui prend le plus souvent la forme d'un "DAW" mais qui peut aussi être une machine dédiée, et qui a pour fonction de contrôler les autres machines. Il constitue le point central de la performance. Ces séquenceurs peuvent aussi être utilisés pour effectuer le mixage des différents éléments. Il est fréquent de voir différentes machines être routées vers un "DAW" pour une dernière étape de mixage avant que le signal soit envoyé vers la sonorisation.

Le principal logiciel utilisé dans les performances de musiques actuelles, et donc d'*Electronic Dance Music*, est le logiciel *Ableton Live*. Il s'agit d'un "DAW", qui comporte, en plus de la vue "arrangement" traditionnelle des "DAWs", une vue nommée session, qui permet un mode de travail adapté à la performance.

Il s'agit simplement d'une vue du logiciel comme d'une matrice dont les colonnes sont les pistes, et les lignes sont des "scènes". On peut ainsi disposer des clips audios

6. Un "DAW", ou "Digital Audio Workstation", est un logiciel qui permet de manipuler et de monter de l'audio et du MIDI. Il intègre le plus souvent une partie "arrangement", qui permet d'effectuer un montage, une partie "mixage", qui permet d'effectuer le mixage et d'ajouter des effets. Certains "DAW" intègrent une partie "session", qui fonctionne avec une matrice et permet de composer facilement des morceaux. Les exemples les plus connus dans le milieu musical sont *Pro Tools* d'Avid, *Ableton Live* d'Ableton, *Logic Pro* d'Apple ou encore *Reaper* de Cockos.

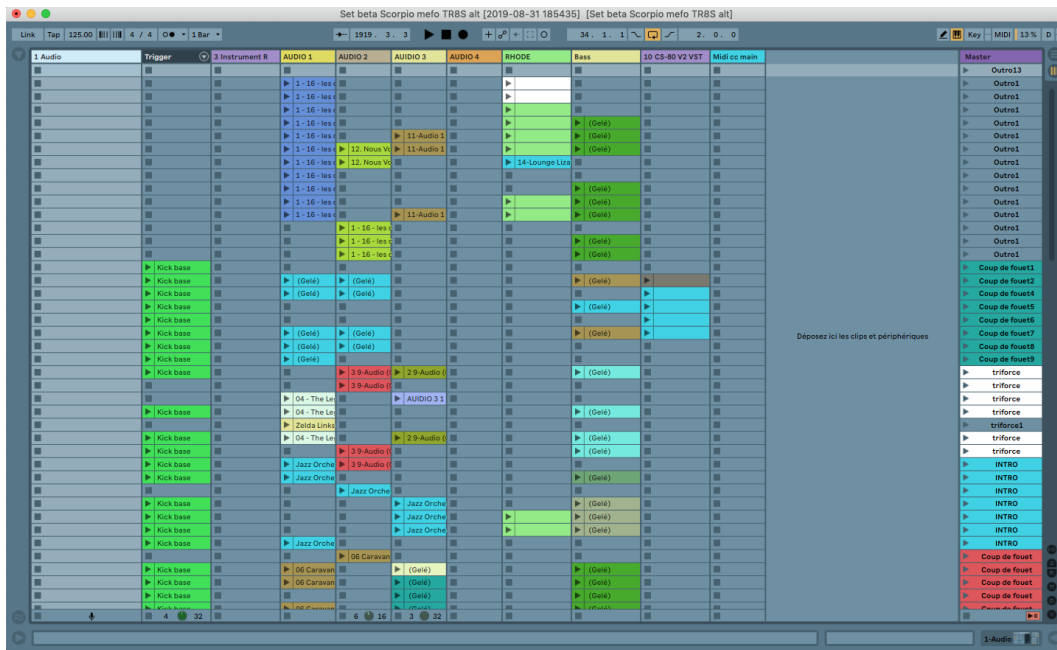


FIGURE 3.1 – Capture d'écran de la vue session d'un "set" ableton préparé pour une performance musicale.

et MIDI sur les pistes et les lancer manuellement, ou encore les lancer de manière groupée en lançant une ligne entière (une scène). Ces clips peuvent ensuite tourner en boucle ou être joués une fois, et peuvent suivre des instructions spécifiques à la fin de leur lecture : il est possible d'indiquer au logiciel de jouer le clip suivant sur la piste à la fin de la lecture, et on peut même ajouter une part aléatoire dans la survenue de cette instruction.

Ableton comporte en outre d'autres manières de simplifier le jeu en temps réel avec des musiciens : il est possible de ralentir ou d'accélérer momentanément le tempo, et aussi de "capturer" un motif audio ou MIDI joué sur une piste même s'il n'a pas été enregistré. Le logiciel intègre aussi nativement un "looper", c'est-à-dire un effet qui capture une séquence audio et la joue en boucle, pour faciliter l'enregistrement à la volée de musique, ce qui est crucial dans les performances où l'on joue des motifs les uns par dessus les autres.

Ableton a enfin pour intérêt d'intégrer le programme *Max/MSP* via un système nommé *Max4Live*. *Max/MSP* est un environnement de programmation graphique qui permet de programmer simplement des "patches", c'est-à-dire des programmes qui permettent de traiter des données, de l'audio ou encore de la vidéo et d'en produire. Cette intégration native de *Max/MSP* dans le programme *Ableton Live* facilite l'intégration et l'utilisation d'éléments programmés par l'utilisateur lui-même et constitue donc une piste intéressante pour la conception et l'intégration de modules dont nous traitons au cours de ce mémoire.

Certaines performances utilisent en outre des séquenceurs "hardware" pour contrôler l'ensemble de la performance, comme le *Tracker* de Polyend ou le *Beatstep* d'Arturia.

3.2.2 Les consoles de mixage

Il y a ensuite les machines dont la fonction est le mixage des éléments, si celui-ci n'est pas effectué sur un "DAW" ou dans un séquenceur. Il est courant de voir des musiciens utiliser des petites consoles de mixage, afin d'éviter les soucis de latence inhérents au fait d'utiliser un "DAW" comme dispositif de mixage. Certains musiciens utilisent parfois des tables de mixages à deux ou quatre voies utilisées par les *DJs* pour appliquer des filtres durant la performance.

3.2.3 Le matériel musical

On peut enfin distinguer un groupe de machines dont la fonction est de générer ou d'exécuter du matériel musical. On en distingue trois types : les boîtes à rythme, les synthétiseurs, et les échantillonneurs.

Une boîte à rythme est une machine qui permet de créer des séquences rythmiques en utilisant divers éléments rythmiques. Une des familles de boîtes à rythme les plus influentes dans le domaine de l'*EDM*, et de la musique actuelle en général, est celle des *TRs* de *Roland* (*TR* pour *Transitor Rythm*).

Le premier modèle, la *TR-808*, conçue à la fin des années 70 et commercialisée pour la première fois en 80, est pensée à l'origine pour permettre de réaliser des maquettes en studio d'enregistrement. Sa sonorité, ainsi que celle de la *TR-909*, est toujours très utilisée aujourd'hui dans un nombre important de productions *Techno* et *House*.

Le fonctionnement de ces boîtes à rythmes, relativement simple, permet au musicien de placer des éléments rythmiques sur une grille en activant ou non des boutons afin de construire un rythme, et de stocker différentes séquences. L'élément qui permet cette manière de composer le rythme se nomme un "step sequencer", ou séquenceur par pas. Elle inclut un potentiomètre de "groove" qui permet de changer la position des doubles croches dans la séquence.

D'autres boîtes à rythmes reprennent le même concept, comme la *Tanzbär* de M.F.B., qui présente une interface plus simple et épurée que les modèles *TR* de Roland. D'autres, comme les *Analog Rythm* d'Elektron, font la démarche inverse, et proposent plus de possibilités au prix d'une interface plus complexe d'utilisation.

On utilise souvent des synthétiseurs, qu'ils soient hardware ou software. Certains ont des arpégiateurs et des fonctions aléatoires. L'arpégiateur est un dispositif qui permet à un synthétiseur d'effectuer automatiquement un arpège à partir des notes



FIGURE 3.2 – Le modèle le plus récent des boîtes à rythmes de Roland : La TR-8S. On voit le séquenceur par pas en bas, avec des lignes qui fournissent des repères rythmiques.

jouées par le musicien, en suivant plusieurs paramètres comme le rythme et la direction du mouvement de l'arpège. L'arpégiateur représente une manifestation très réduite de composition algorithmique, puisque le synthétiseur compose seul l'arpège à partir d'une entrée simple.

Certains synthétiseurs sont conçus comme des séquenceurs, comme la *TB-303* de Roland, à l'origine développée pour servir de basse pour que les musiciens puissent jouer chez eux, et détournées quelques années après être tombée en désuétude pour servir de basse principale de nombreux morceaux de *Techno*, créant un genre à part entière, la *Acid House* précédemment évoquée. Ces synthétiseurs permettent au musicien d'écrire manuellement un motif mélodique plutôt que de le jouer en temps réel, parfois en intégrant des modes restreints pour faciliter la composition.

Les échantillonneurs permettent de stocker des échantillons audio et de les jouer soit manuellement, soit avec une logique de séquenceur. Un des modèles les plus

notables est le *MPC* de Roland, qui se décline en de nombreuses versions

3.3 Le modèle général de la performance

On peut diviser les outils de la performance en trois catégories : le contrôle, le matériau musical, et le mixage. On retrouve souvent un nœud central qui permet la synchronisation des différentes machines et le déclenchement des différentes séquences, voire parfois le contrôle des machines assujetties. Il est de plus en plus courant que ce rôle soit joué par un ordinateur avec un "DAW", et avec une interface audio et MIDI pour le lien avec les autres machines. Le matériau musical renvoie aux synthétiseurs, échantillonneurs et boîtes à rythmes précédemment évoquées. Le mixage concerne deux aspect : le mixage à proprement parler, c'est-à-dire le réglage des volumes et du placement spacial des éléments, et l'ajout d'effets comme des réverbérations ou des filtres.

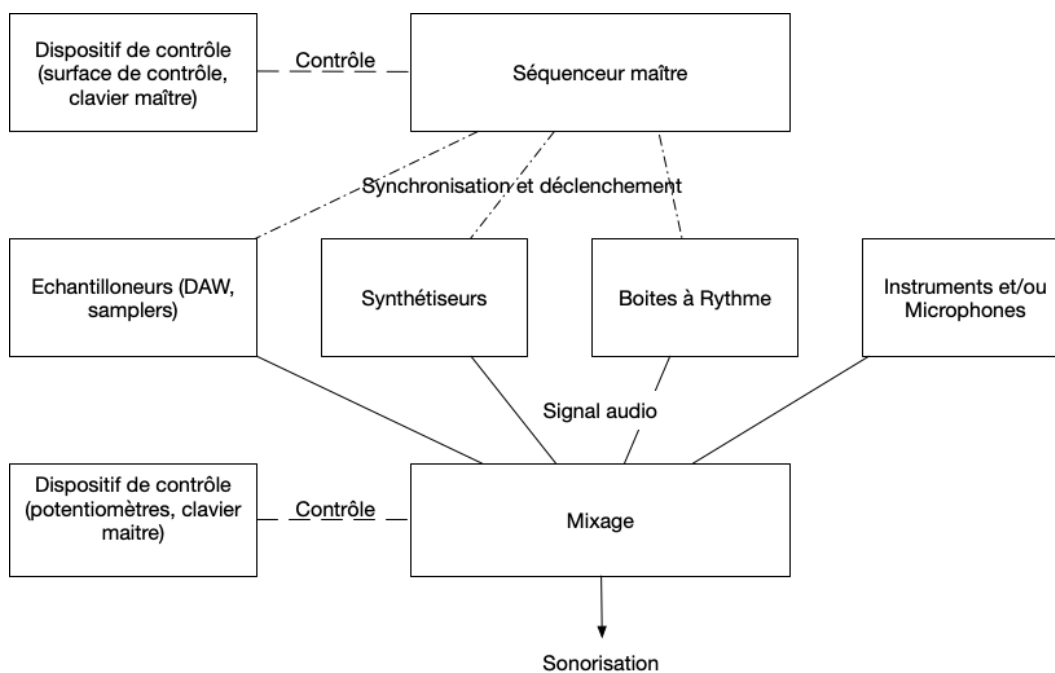


FIGURE 3.3 – Synoptique général d'un dispositif de performance musicale.

3.4 Exemples de dispositifs

Nous avons interrogé plusieurs musiciens afin de montrer quelques exemples de dispositifs utilisés dans la performance.

Nous avons interviewé Lucas Moinet, qui forme avec Alex Valard le duo français de *House Groove Boys Project*. Leur approche de la performance se range du côté "performatif" : leur objectif est de jouer le plus possible d'instruments et de réduire le plus possible la place que prend l'ordinateur dans la performance.



FIGURE 3.4 – Les Groove Boys Project en pleine performance. Crédit photo : Leviet photography.

Cette idée se retrouve dans leur dispositif : l'ordinateur qu'ils utilisent comme séquenceur pour jouer des boucles préparées en amont est esclave de la *MPC* qui sert à synchroniser les autres machines.

Les rôles des musiciens sont séparés selon les instruments. Lucas Moinet s'occupe principalement des instruments électriques (une basse, une guitare), et des synthétiseurs analogiques et numériques, comme un *kingKORG* et un *Reface CP*. Il les joue en direct pendant la performance et se concentre principalement sur la section harmonique et mélodique. Alex Valard s'occupe de la partie rythmique et du mixage. Il

manipule une *MPC*, l'ordinateur, ainsi que des boîtes à rythmes. Il utilise ainsi une *TR-707*, une *TR-8*, ainsi qu'une *Tanzbaär*.

Leurs performances sont "très écrites", c'est-à-dire qu'ils laissent peu d'évènement au hasard. Elles se composent principalement de lignes écrites à l'avance auxquelles le duo ajoute des solos d'improvisation. Elles se rapprochent ainsi des concerts de musique actuelle traditionnels. Cette logique est motivée par l'envie de garder l'identité de chaque morceau de la manière la plus fidèle entre le studio et la scène.

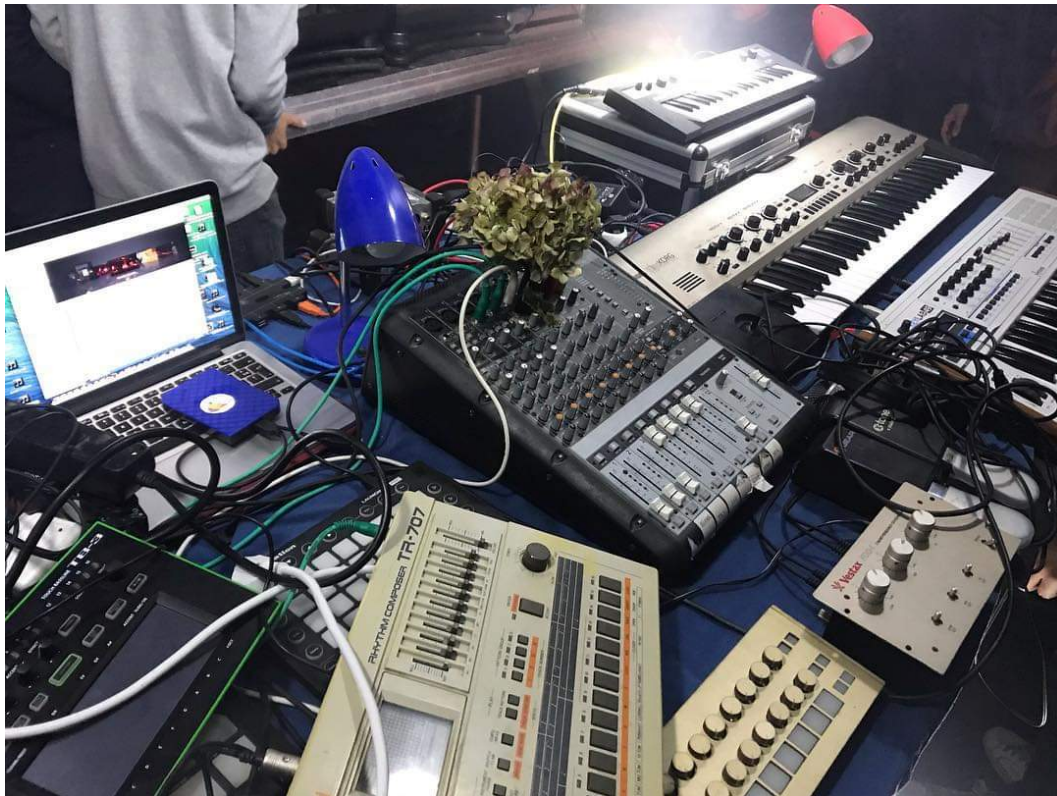


FIGURE 3.5 – Le matériel utilisé par les Groove Boys Project durant une performance.

Nous avons également interviewé Tom Ferreira, connu comme artiste sous le nom de Fasme. Il effectue des performances qu'il décrit comme se rattachant au genre *Electro*⁷. Son approche de la performance se range du côté "contrôle" : Il joue des motifs MIDI réalisés à l'avance et se concentre sur l'application d'effets, l'arrangement et le mixage des différentes machines à sa disposition. Comme les Groove Boys, il est dans une démarche d'émancipation vis-à-vis de l'ordinateur et n'en utilise plus depuis au moins un ans. La machine qu'il utilise comme séquenceur central est *Octatrack* d'Elektron. Elle contient les différents motifs MIDI qu'il utilise durant sa performance, et les envoie vers des synthétiseurs. Un des intérêts de cette machine est qu'elle permet de mettre en place des conditions de déclenchement relativement complexes sur les motifs MIDI (des probabilités et des systèmes de suivis par exemple).



FIGURE 3.6 – Le matériel utilisé par Fasme durant une performance.

7. Le genre *Electro* décrit précisément la musique jouée par Kraftwerk et des groupes similaires dans les années 70, mais il est parfois revendiqué par des musiciens gravitant dans la scène *Techno*.

Il utilise différents synthétiseurs, comme l'*Analog 4* d'Elektron, le *TD3* de Beringer, le *Prophet Rev2* de Sequential ou le *Minilogue XD* de KORG. Son approche de contrôle se retrouve particulièrement dans le choix des synthétiseurs : certains comme le *Minilogue XD* ne comportent pas de clavier maître. Il utilise en outre un *Analog Rythm* d'Elektron comme boîte à rythme.

Il ne laisse aucune place à l'improvisation instrumentale sur scène. En revanche, il se laisse une grande liberté dans l'arrangement des différents motifs qu'il joue simultanément.

Le choix de s'éloigner de l'ordinateur est lié à une envie d'avoir un son "analogique". Il préfère utiliser une sommation analogique sur une table de mixage et utiliser des compresseurs hardware sur la piste master, plutôt qu'utiliser les effets intégrés d'un "DAW" comme à ses débuts.



FIGURE 3.7 – Fasme en pleine performance.

Chapitre 4

La musique algorithmique

L'utilisation d'algorithmes dans un cadre d'improvisation musicale s'inscrit dans la lignée d'une branche de la musique expérimentale que l'on nomme musique algorithmique. De la même manière que le terme "musique classique" est souvent utilisé pour décrire l'ensemble de la musique savante occidentale alors qu'il désigne en réalité une période précise de cette musique savante, nous utilisons ici le terme "musique algorithmique" au sens large, c'est-à-dire de musique composée, en temps réel ou à l'avance, en utilisant un ou des algorithmes, et non pas en utilisant le sens précis du courant de la musique algorithmique initié par Pierre Barbaud¹. L'utilisation d'algorithmes pour composer de la musique est loin d'être récente. Cependant leur utilisation en temps réel et en interaction avec un musicien humain n'a pu se développer que dans les dernières décennies du fait de la puissance de calcul disponible sur les ordinateurs modernes. C'est pour cette raison qu'elle est souvent classée dans la recherche comme faisant partie de la musique informatique ou "computer music". Il est à noter que la notion de musique aléatoire, que nous allons régulièrement évoquer, est très liée à celle de musique algorithmique, ce qui motivera plusieurs parallèles que nous allons effectuer.

1. Voir 4.1.2 : La musique algorithmique assistée par ordinateur.

4.1 Les origines de la musique algorithmique

4.1.1 La musique algorithmique avant l'ordinateur

L'idée d'utiliser des algorithmes pour composer de la musique n'a pas attendu l'arrivée de l'ordinateur pour se développer. Bien avant les débuts de l'informatique, on trouve chez de nombreux compositeurs de musiques savantes et populaires des logiques de compositions utilisant des règles ordonnées et précises.

Selon l'Encyclopædia Universalis l'algorithme est un "ensemble prédéterminé d'instructions pour résoudre un problème donné dans un nombre limité d'étapes" [6]. Il apparaît ainsi que de nombreuses logiques fournissant à un compositeur des instructions à suivre pour faciliter la composition sont de l'ordre de l'algorithme.

La musique est une forme d'art dont l'appréciation se base en partie sur un grand nombre de normes tacites. Il est donc compréhensible qu'on ait souhaité très tôt formaliser des règles pour la composer, et qu'on ait cherché à agencer ces règles dans un ordre précis pour établir un protocole de composition. Le premier exemple connu de cette tentative de formalisation de règles est le *Micrologus de disciplina artis musicæ*, rédigé en 1025 par le moine bénédictin Guido d'Arezzo, où il formule une procédure à suivre pour transformer un texte liturgique en musique [7]. La musique savante occidentale comporte de nombreuses règles d'écriture. Concernant les modes, par exemple, elle en sélectionne deux, un majeur et un mineur, ce dernier étant décliné sous plusieurs versions selon que la gamme est ascendante ou descendante. Des règles sont aussi en usage dans les principes harmoniques, ou encore

dans les formes qu'une œuvre peut prendre, comme la forme sonate. La forme sonate est constituée de trois parties : une exposition, où l'on fait entendre deux thèmes musicaux, un développement, où on les agrmente et on les enrichit, et enfin la réexposition, où l'on expose à nouveau les thèmes [8]. Il s'agit donc de règles narratives². Ces codes peuvent être vus comme un ensemble de règles à suivre pour composer une pièce de forme sonate, et si on les considère comme un processus à suivre, cela s'apparente à un algorithme.

On trouve aussi chez des compositeurs de musique savante occidentale la volonté de varier l'exécution d'une œuvre en se basant sur le hasard en accompagnant la partition d'un processus à suivre. Les *Musikalisches Würfelspiel*, "Jeu de dé musical" en allemand, populaires en Europe au XVIIIème siècle, utilisent cette logique et reposent sur le fait de jeter aux dés différentes caractéristiques d'un morceau. On y rattache souvent un manuscrit de Wolfgang Amadeus Mozart rédigé en 1787 et qui contient 176 fragments de partitions d'une mesure à recombinaison pour créer un morceau [9]. Le titre du manuscrit, *Anleitung zum Componieren von Walzern so viele man will vermittelt zweier Würfel, ohne etwas von der Musik oder Composition zu verstehen*, signifie "Instructions pour la composition d'autant de valses que l'on peut souhaiter avec deux dés, sans ne rien comprendre de la musique ou de la composition" et préfigure ainsi l'idée de musique composée à l'aide d'instructions.

Cette idée de musique composée directement avec un ensemble d'instructions sera ensuite développée dans la musique dodécaphonique et la musique sérielle qui apparaissent au début du XXème siècle. On y trouve les premiers exemples de la prédominance d'une logique algorithmique dans la composition musicale.

Arnold Schoenberg, un compositeur autrichien de la première moitié du XXème siècle

2. Les règles narratives sont les règles qui concernent l'organisation générale et momentanée d'un morceau.

Table des Chiffres pour le Walzer.
Zahlentafel für den Walzer.

Première Partie.
Erster Theil.

	A	B	C	D	E	F	G	H
2	96	22	141	41	105	122	11	30
3	32	6	128	63	146	46	134	81
4	69	95	158	13	153	55	110	24
5	40	17	113	85	161	2	159	100
6	148	74	163	45	80	97	36	107
7	104	157	27	167	154	68	118	91
8	152	60	171	53	99	133	21	127
9	119	84	114	50	140	86	169	94
10	98	142	42	156	75	129	62	123
11	3	87	165	61	135	47	147	33
12	54	130	10	103	28	37	106	5

Seconde Partie.
Zweiter Theil.

	A	B	C	D	E	F	G	H
2	70	121	26	9	112	49	109	14
3	117	39	126	56	174	18	116	83
4	66	139	15	132	73	58	145	79
5	90	176	7	34	67	160	52	170
6	25	143	64	125	76	136	1	93
7	138	71	150	29	101	162	23	151
8	16	155	57	175	43	168	89	172
9	120	88	48	166	51	115	72	111
10	65	77	19	82	137	38	149	8
11	102	4	31	164	144	59	173	78
12	35	20	108	92	12	124	44	131

FIGURE 4.1 – La Table des Chiffres du Jeu de Dés de Wolfgang Amadeus Mozart. Elle permet de choisir les motifs musicaux à jouer, parmi les 176 disponibles, en fonction du résultat des jets de dés.

considéré comme le père du sérialisme, définit dans les années 1920 des règles de composition pour créer un morceau utilisant les douze tons dans le but de s'éloigner le plus possible des codes musicaux classiques [10]. L'idée est de créer une série de notes, en respectant certaines contraintes, pour ensuite composer la pièce en utilisant diverses techniques pour développer et altérer cette série. La musique sérielle comporte en revanche une spécificité importante. Avant elle, les règles énoncées dans la composition musicale permettaient d'obtenir des effets musicaux et narratifs sans être obligatoires. Elle servaient surtout à donner une charnière au morceau et à se concentrer sur des motifs mélodiques et rythmiques dont on savait qu'ils sonnaient de manière cohérente dans l'oreille des auditeurs de l'époque. Dans la musique sérielle, les règles sont, à l'inverse, partie intégrante de la composition, l'algo-

rithme guidant et contraignant le compositeur plutôt que l'inverse [11].

Dans cette lignée, le compositeur français Olivier Messiaen travaille sur des systèmes qu'il appelle des "modes" en utilisant les apports de la musique sérielle pour formaliser de manière mathématique des processus de composition complexes. Dans *Modes de Valeurs et d'intensités*, un des mouvements de l'œuvre *Quatres études de rythme*, il sépare et énumère les éléments musicaux de l'œuvre en leur assignant des nombres, et utilise des règles mathématiques pour structurer la composition [12]. On trouve donc déjà chez Olivier Messiaen la volonté de séparer et d'énumérer de manière précise les possibilités musicales. Cette séparation sera par la suite une étape nécessaire dès lors que l'on utilisera un ordinateur pour exécuter la composition.

Ce morceau utilise un mode de hauteurs (36 sons), de valeurs (24 durées), d'attaques (12 attaques), et d'intensités (7 nuances). Il est entièrement écrit dans le mode.

Attaques: > 1 2 3 4 5 6 7 8 9 10 11 (avec l'attaque normale, sans signe, cela fait 12.)

Intensités: ppp pp p mf f ff fff

Sons: Le mode se partage en 3 Divisions ou ensembles mélodiques de 12 sons, s'étendant chacun sur plusieurs octaves, et croisés entre eux. Tous les sons de même nom sont différents comme hauteur, comme valeur, et comme intensité.

Valeurs:

Division I: durées chromatiques de 1 à 12 (etc.)

Division II: durées chromatiques de 1 à 12 (etc.)

Division III: durées chromatiques de 1 à 12 (etc.)

Au total 24 durées: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24

FIGURE 4.2 – Le préambule de *Mode de valeurs et d'intensités* d'Olivier Messiaen. Y sont exposés les différents éléments musicaux qui composent le "mode" et leurs valeurs [12].

Au XXème siècle, la musique aléatoire³ apporte des procédés qui seront réutilisés

3. La musique aléatoire est une branche de la musique contemporaine et expérimentale.

dans la musique algorithmique. Si l'utilisation d'aléatoire ne suffit pas pour inscrire un morceau dans une logique algorithmique, la démarche de se donner une suite d'instructions pour produire un morceau en choisissant des nombres ou des valeurs au hasard témoigne d'un processus de composition qui s'apparente au fonctionnement interne de certains programmes modernes. En effet, un algorithme de composition doit permettre des variations dans son exécution, sans quoi l'œuvre résultante serait toujours la même. Ces variations peuvent être dues à des choix arbitraires ou esthétiques d'un compositeur humain qui effectue l'algorithme, à des choix effectués par un logiciel en fonction d'une ou de plusieurs variables, ou à des choix laissés au hasard.

Les premiers exemples modernes sont posés par des compositeurs comme Charles Ives [13], un compositeur américain, au début du XX^{ème} siècle, et repris par Henry Cowell dans les années 1930 dans des œuvres comme *String Quarter No.3* (1934), dans laquelle le musicien peut réarranger certaines séquences au hasard.

John Cage, un compositeur américain de musique contemporaine de la seconde moitié du XX^{ème} siècle, crée plusieurs œuvres se basant sur des règles arbitraires et l'introduction de l'aléatoire dans la performance, comme *Music of changes* (1951) ou *Two Pastorales* (1952) qui utilisent une technique de notation musicale inspirée du *I Ching* ou *Yi King*, un livre de divination chinoise qui se base sur des tirages aléatoires à effectuer soi-même.

On trouve d'autres exemples de composition aléatoire chez Earle Brown, un autre compositeur américain, qui en 1953 compose *Octet 1* en utilisant un livre contenant des milliers de nombres aléatoires pour définir les événements sonores de sa composition.

On trouve enfin chez Iannis Xenakis, un compositeur français, dans les années 50,

avec la musique stochastique, plusieurs exemples de règles, qu'elles soient abstraites ou techniques, établies dans le but de construire la charnière de la composition ou de son exécution. Il utilise de nombreuses règles complexes issues de la physique et des mathématiques. Dans l'œuvre *Pithoprakta*, composée en 1956, il calcule la vitesse des glissandi⁴ des cordes à l'aide de la loi de Gauss, une loi de probabilité issue des mathématiques, et la distribution des événements sonores selon la loi de Poisson, une autre loi de probabilité issue des mathématiques [14]. L'introduction de probabilités permet d'avoir plus de contrôle dans les résultats de tirages aléatoires : la courbe de Gauss, comme la loi de Poisson permettent ainsi de pondérer des tirages aléatoires en augmentant la probabilité de tirage autour d'une valeur centrale. Iannis Xenakis crée d'ailleurs un programme informatique pour produire des données pour ses compositions, en utilisant la grande capacité de calcul de l'ordinateur pour effectuer les calculs théoriques de probabilité qu'il intègre dans ses œuvres.

4.1.2 La musique algorithmique assistée par ordinateur

L'arrivée de l'informatique signe le début de l'autonomie des algorithmes, qui peuvent dès lors fonctionner indépendamment du compositeur. Si l'on a pu voir chez Iannis Xenakis des utilisations de l'ordinateur pour effectuer des calculs, il faut attendre la fin des années 50 et les débuts de la musique assistée par ordinateur pour voir les premiers exemples de programme effectuant la composition. Une des premières pièces composée de manière algorithmique à l'aide d'un ordinateur est la Suite Illiac pour quatuor à cordes composée par Lejaren Hiller et Leonard M. Isaacson en 1956. Il s'agit d'une œuvre générée grâce à l'ordinateur Illiac IV de l'univer-

4. Le glissando désigne le fait de passer d'une note à une autre de manière continue.

sité d'Illinois, à l'aide d'une suite d'instructions. Elle se décompose en quatre mouvements, chacun explorant un aspect différent de la génération algorithmique. Ces quatre mouvements correspondent en réalité à quatre expériences menées à la suite. Les deux premières permettent à Lejaren Hiller et Leonard M. Isaacson de travailler sur la mise en forme mathématique de l'écriture musicale et la création d'un ensemble de règles pour composer des mélodies. La troisième repose sur des processus plus aléatoires et une idée plus abstraite de composition (le but est de travailler sur la dynamique), et enfin la quatrième, dans la lignée des travaux de Iannis Xenakis, utilise des lois de probabilités et des chaînes de Markov pour composer de la musique avec un matériau de base non musical [15].

On pourra aussi citer la pièce *Push Button Bertha*, réalisée en 1956 par Martin Klein et Douglas Bolitho, deux mathématiciens américains qui composent une mélodie avec des procédés aléatoires et des contraintes logiques interdisant certains intervalles ou enchaînements d'accords. Les contraintes sont basées sur des observations relatives à la musique populaire américaine de l'époque, et vont de l'observation large sur la structure ("le morceau produit doit avoir une structure de type AABA", suivit de spécifications sur le nombre de notes par partie), à des observations très spécifiques ("si 5 notes consécutives vont dans la même direction, la sixième doit aller dans la direction contraire") [16]. L'œuvre fait parler d'elle jusque dans les colonnes du *New York Times*, et contribue à populariser l'idée de musique algorithmique auprès du public. L'engouement des journaux pour la pièce se base cependant moins sur la qualité musicale de la pièce que sur une question légale nouvelle : celle de la propriété des droits musicaux [17].

Le terme "musique algorithmique" peut être attribué à Pierre Barbaud, un pionnier de l'informatique musicale. Il cofonde en 1958 le Groupe de musique algorithmique

avec Jeannine Charbonnier, Roland Douat et Roger Blanchard. Le groupe s'attachera à définir des méthodes de composition, qui seront utilisées pour composer diverses œuvres, comme *Variations heuristiques* ou *Lumpenmusik* de Barbaud. Pierre Barbaud développe en 1959, avec Roger Blanchard, le programme *ALGOM I-5*, qui se base sur des processus aléatoires pour composer de la musique et fonctionne sur l'ordinateur Gamma 60 du centre de calcul électronique de la compagnie *Bull* à Paris [18]. Il compose, en utilisant le même ordinateur, l'œuvre *BULLA* pour quintette à corde, en 1962, et revient sur les mêmes interrogations liées aux droits d'auteurs que celles précédemment évoquées, en demandant, lors de l'enregistrement de l'œuvre à la *SACEM*, que les tableaux de chiffres obtenus par son algorithme soient enregistrés au même titre que l'œuvre. Pierre Barbaud contribue fortement au développement de la musique algorithmique comme discipline scientifique, publiant en 1968 *La musique, discipline scientifique : introduction élémentaire à l'étude des structures musicales*, dans lequel il détaille de nombreux procédés de compositions algorithmiques et des manières théoriques d'appréhender la musique comme produit d'un algorithme. Comme le note André Riotte dans la préface d'un ouvrage de Pierre Barbaud consacré à Arnold Schoenberg, on assiste ensuite dans les années 60 à une "période explosive des années 1960 où l'usage de l'ordinateur et des technologies de l'information a commencé à bouleverser les modes de pensée et de travail" [19].

L'informatique étant peu accessible dans les années 70 à 80, la musique algorithmique reste circonscrite au cadre des chercheurs et de certains compositeurs ayant les moyens d'accéder à des ordinateurs. Certains compositeurs contemporains contribuent néanmoins à cette époque à étoffer le répertoire de la musique algorithmique en dépassant le cadre de la recherche, comme Brian Eno, qui en 1978 publie l'album *Music for Airports* qui regroupe ses premières compositions algorithmiques.

1		1		6
9	8	8	8	8
6		1	1	
P	6	6	2	2
4	1	3		
		3	3	3
3	1	4		
P	P	V	V	V
4	3	1		
P	5	5	5	5

The image shows a handwritten musical score for a string quartet and woodwinds. The score is written on five staves. The top staff is for Flute 1 (Fl. no 1), the second for Flute 2 (2°), the third for Viola, the fourth for Cello 1 (Cello 1°), and the fifth for Cello 2 (Cello 2°). The tempo is marked as quarter note = 76. Dynamics include forte (f) and mezzo-forte (mf). The notation includes various rhythmic values, accidentals, and phrasing slurs.

FIGURE 4.3 – Les tables obtenues pour la composition de Bulla, et l'équivalent en partition.

La manière dont Brian Eno a eu l'idée de l'œuvre préfigure d'ailleurs de nombreuses problématiques : c'est en patientant dans l'aéroport de Cologne, en attendant d'embarquer, qu'il imagine une musique qui puisse s'adapter au lieu et à ses événements sonores (comme les annonces) et bruits environnants (comme le brouhaha ambiant des discussions des voyageurs) [20].

Cette idée d'adaptation et de musique de longue durée est au cœur de l'intérêt de l'utilisation de la musique algorithmique, qu'il s'agisse de son utilisation à des fins d'écoute, ou d'accompagnement, comme dans le jeu vidéo ou la scénographie.

Le compositeur d'origine turque Kemal Ebcioğlu crée en 1986 *CHORAL*, un système logiciel pour créer des chorales à quatre voix dans le style de Jean Sébastien Bach. Ce système, qui fonctionne avec plus de 270 règles avec des ordres de priorité, montre le potentiel de l'utilisation d'ordinateurs pour implémenter un grand nombre de règles, ce qui était jusqu'alors compliqué à faire pour un humain seul, et pour un ordinateur avec une puissance de calcul limitée [21].

Une année plus tard, en 1987, David Cope, un compositeur américain, initie un tra-

vail nommé *EMI, Experiments in Musical Intelligence*, dont le but est de réaliser des programmes qui analysent des œuvres pour créer de la musique en réutilisant et modifiant les motifs identifiés. Il s'agit d'un des premiers exemples de ce qu'on appelle la musique générative, qui désigne une branche de la musique algorithmique où les programmes fonctionnent par apprentissage [22].

Il est en revanche fastidieux de formaliser des règles durant cette période, car chaque règle doit être codée avec un langage de programmation informatique très "bas-niveau", ce qui rend compliquée et longue l'implantation d'une règle, et réserve l'utilisation et le développement de ces systèmes à des utilisateurs ayant des compétences avancées en programmation.

Le nombre d'œuvres réalisées avec des moyens algorithmiques augmente ensuite fortement à partir de la fin des années 90, avec la démocratisation de l'informatique chez les particuliers, et l'accès plus simple à des méthodes de programmation.

Les environnements PureData et Max permettent par exemple à des artistes et des musiciens de créer des programmes sans avoir besoin de maîtriser une syntaxe et une grammaire de programmation complexe.

Des artistes et chercheurs en informatique développent ainsi des programmes capables de générer de la musique de manière algorithmique à volonté, parfois en temps réel, comme Laurie Spiegel qui réalise en 1997 le programme *Music Mouse*, qu'elle qualifie "d'instrument intelligent", et qui génère seul de la musique à partir d'informations entrées par l'utilisateur.

Certains artistes développent des applications pour téléphone qui produisent des flux ininterrompus de musique générée par algorithme, comme *Bloom*, une application qui génère de la musique *Ambiant* développée par Brian Eno.

Dans les années 2000 émergent des nouvelles pratiques liées à l'informatique musi-

cale, comme le "live-coding" ("programmation en direct"), qui consiste à programmer en temps réel des algorithmes de composition. On peut penser à *Algorave*, un évènement rassemblant des performances au cours desquelles des artistes programment en direct des instructions pour qu'un algorithme génère de la musique *Techno* : Il s'agit d'un renversement intéressant de la logique générale de la performance. Si les règles sont normalement rédigées à l'avance pour être exécutées en direct, dans les performances d'*Algorave* les règles sont rédigées en direct par des artistes programmeurs [23].

On peut aussi citer l'exemple d'*Orca*, un logiciel de "live-coding" développé par Hundred Rabbits, qui permet de créer à la volée des instructions pour effectuer des arpegges en MIDI que l'on peut ensuite envoyer vers des synthétiseurs.

En parallèle du "live-coding", de nombreuses tentatives de créer des programmes de composition algorithmiques se développent, comme dans le jeu vidéo, que nous développerons plus tard, et commencent à intégrer des logiques d'apprentissage de plus en plus complexes, voir de "Deep-Learning"⁵, comme le "doodle" de *Google* que nous évoquerons dans la section 4.2

Concernant les origines de la musique algorithmique, il est nécessaire de distinguer dans les règles que nous avons précédemment évoquées celles qui tiennent des codes esthétiques et des critères musicaux de celles qui tiennent du processus de composition.

Les premières posent en effet des charnières très liées à l'époque et aux habitudes d'écoute du public et peuvent être transgressées pour obtenir divers effets.

Les secondes renvoient au processus de composition lui même et cherchent à le faciliter. Cette distinction permet de rendre compte de l'importance de l'erreur et de la

5. Le "Deep-Learning" ou apprentissage profond, est un modèle d'apprentissage automatique qui se base sur un haut niveau d'abstraction dans le traitement des données.

transgression musicale, qui constitue un des intérêts de l'utilisation de l'algorithme en musique, et plus largement dans l'art.

Certaines disciplines artistiques se basent ainsi entièrement sur les "glitches", des effets issus de l'utilisation détournée d'un algorithme qui ne sont pas contenus dans les intentions de la programmation.

Le principe du *Glitch Art* est ainsi de détourner l'utilisation d'un programme, soit en lui envoyant des ordres ou des valeurs qu'il n'est pas censé manipuler, soit en introduisant des changements dans son code, pour obtenir des résultats contre-intuitifs [24]. Si cette discipline est principalement visuelle, on peut constater qu'il est possible, en musique, de faire une utilisation créative des "glitches".

Si le "glitch" en musique se rapporte plus au fait de faire "glitcher" les programmes d'instruments et de lecteurs musicaux pour créer des sons intéressants, certains musiciens s'amuse à faire "glitcher" des programmes de composition autonome pour obtenir du contenu musical nouveau.

Dans une thèse consacrée à l'improvisation musicale entre humain et ordinateur, Jérôme Nika note ainsi que les musiciens qui ont essayé ses systèmes tendent à dire qu'ils essayent de pousser le système dans ses retranchements et de le mettre en difficulté dans le but d'obtenir des résultats intéressants [25].

4.2 Les deux approches de l'algorithme

Il y a dans la logique algorithmique deux approches générales : l'approche par règle et l'approche par apprentissage.

Dans l'approche par règle, le programme est construit avec un ensemble de règles. Il s'agit de l'approche la plus traditionnelle de l'algorithme. À chaque situation le programme compare les entrées avec son ensemble de règles internes pour produire une sortie. Cette approche demande beaucoup de réflexion en amont de la part du concepteur, et le fonctionnement interne reste identique tant que le concepteur n'en modifie pas le fonctionnement. Cette méthode peut incorporer des événements aléatoires, et un système de probabilités. En outre, on peut noter qu'elle permet au concepteur d'avoir plus de contrôle précis sur l'aspect artistique produit par la machine. Appliquée à la musique, cette approche revient à essayer de formuler un vocabulaire musical, et une grammaire pour agencer ce vocabulaire. Les systèmes qui utilisent cette approche tendent à avoir un nombre très élevé de règles (plusieurs centaines, dans le cas de *CHORAL*, ou de *Voyager*), et présentent comme avantage le fait de pouvoir être rapidement modifiés et d'être intuitifs dans leur conception. Il est relativement simple pour un musicien utilisant un système par règle de poser une règle supplémentaire s'il constate un résultat non désiré, et d'agencer ces règles avec un ordre de priorités pour organiser la génération de musique.

Cette conception est à rapprocher du terme de "musique procédurale", qui revient souvent dans les publications de recherche. Il s'agit de générer de la musique en suivant une procédure indiquée par le compositeur, donc un ensemble de règles ordonnées. Cette musique suit de manière plus ou moins précise une charnière posée par le compositeur dans sa réalisation.

La musique procédurale a pour intérêt de proposer à des utilisateurs de composer des morceaux sans disposer de connaissances avancées en théorie musicale ou en code. On peut ainsi réaliser des programmes qui s'occupent d'harmoniser en fonction d'instructions simples de la part de l'utilisateur, ou proposent des manières alternatives d'envisager la composition. On trouve ainsi divers programmes qui permettent de composer de la musique avec une interface matricielle en deux dimension comme *Automatune*, où l'utilisateur fournit au logiciel des instructions sous forme de flèches et de notes. Certains programmes produisent de la musique en suivant des instructions abstraites et qui s'éloignent de l'idée attendue de composition musicale. On peut citer à cet égard *CAMUS* de Eduardo Miranda, publié en 2001, qui produit de la musique en se basant sur les positions de cellules d'un algorithme de *Game of life*⁶ pour générer des intervalles de triades⁷ [26]. Cette manière d'approcher la composition permet de penser la création de musique d'une manière détournée, et offre une piste pour permettre des changements automatiques basés sur des variations d'un contexte non musical.

Dans l'approche par apprentissage, le logiciel reçoit des données qu'il analyse pour en intégrer le comportement, et le retour du concepteur du logiciel sur les sorties peut réajuster le fonctionnement interne du logiciel. Cette idée n'est pas spécifique à l'informatique musicale, de nombreux logiciels s'en servant pour affiner des manières de traiter des données. Par exemple les programmes de reconnaissance faciales s'appuient sur des millions de photos dans lesquelles les visages sont identifiés pour que le programme apprenne à reconnaître des visages en ajustant son fonctionnement interne.

6. Le game of life est un programme qui simule la vie de cellules en se basant sur des règles simples.

7. Une triade est un accord formé de trois notes.

Cette approche permet au logiciel de s'améliorer et de changer légèrement de fonctionnement à chacune de ses utilisations, mais demande un temps relativement long de "calibrage" pour obtenir une sortie conforme aux attentes du concepteur : il faut déjà fournir au logiciel des exemples de compositions pour qu'il les analyse, et le faire travailler un nombre important de fois pour qu'il arrive à un résultat exploitable.

Certaines implantations de systèmes par apprentissages sont riches de possibilités, comme le *Continuator* de François Pachet, que nous évoquerons plus en détail au chapitre suivant, un programme qui écoute en temps réel un musicien et tente d'en imiter des traits stylistiques, et dont la visée n'est pas tant de copier le musicien mais de lui permettre de jouer avec un système qui essaye de jouer comme lui.

Ainsi, en 2019, à l'occasion de l'anniversaire de Jean Sébastien Bach, le moteur de recherche *Google* intègre un "doodle", une petite animation sur son moteur de recherche, qui invite les utilisateurs à écouter un morceau généré par apprentissage "à la manière de Jean Sébastien Bach", les nombreux morceaux écrit par Jean Sébastien Bach ainsi que les retours des utilisateurs aidant à recalibrer le programme pour améliorer le résultat. Mais certaines critiques opposent que le programme, n'ayant pas une approche de la composition traditionnelle, commet des erreurs "qu'un étudiant en première année de musique, apprenant les règles du contrepoint, n'aurait pas commis" [27].

Si les entorses aux règles sont courantes dans la composition musicale, Jean Sébastien Bach les ayant transgressées à de nombreuses reprises, celles que produisent un algorithme qui ne s'est formé que par apprentissage se ressentent rarement comme étant motivées par un choix esthétique.

On voit ainsi une des limites de l'approche par apprentissage : même si les créateurs d'un programme ont connaissance de certaines règles musicales, ils ne peuvent pas

directement les inclure dans le logiciel et doivent espérer qu'il comprenne ces règles par lui même. On remarque, en revanche, que l'analyse de nombreux échantillons musicaux permet parfois de faire apparaître des règles difficilement repérables par des êtres humains.

Ces deux approches peuvent être combinées, comme dans le cas du logiciel *Band out of a box*, que nous verrons au chapitre 5, et qui fonctionne avec des règles, et y ajoute de l'apprentissage.

Ces deux approches peuvent avoir recours à un troisième aspect de l'algorithme qui est l'utilisation précédemment évoquée de l'aléatoire. Nous avons choisi de la distinguer, étant donné que certains travaux de recherches distinguent les systèmes musicaux par règle, par apprentissages, et stochastiques, sans pourtant en faire une approche à part entière, car son utilisation doit se faire dans un contexte général, et que les deux approches y ont recours.

4.3 Les logiques algorithmiques dans la musique Techno

Si l'on a pu précédemment citer *Algorave* comme exemple de musique algorithmique se rattachant au genre *Techno*, plusieurs modes de composition algorithmique sont déjà utilisés dans les musiques électroniques actuelles.

On trouve ainsi les arpégiateurs, qui réalisent des arpèges à un rythme donné en fonction des notes jouées par le musicien. Plusieurs boîtes à rythme permettent en outre de composer automatiquement des motifs aléatoires, et permettent d'en générer à la volée comme la fonction "Fill-In" des différents modèles de *TR* de Roland. Il reste cependant à l'utilisateur le soin d'agencer ces différents motifs à la suite et de les faire varier en fonction du temps.

On peut citer le duo *Algobabez*, formé en 2016, composé de la "data-musician" Shelly Knott et de l'artiste Joanne Armmitage, qui composent des morceaux en utilisant un programme nommé *SuperCollider*. Elles utilisent ce programme pour coder et contrôler des motifs pour créer une musique se rapprochant de la *Techno*, décrite par les artistes comme "Synth-driven algo-pop". Elles participent régulièrement aux évènements *Algorave*.

Le caractère répétitif de la *Techno*, ainsi que son aspect fortement codifié, lui permettent d'être facilement créée par des algorithmes génératifs. *Eternal Flow*, réalisé par l'artiste russe Petr Serkin, est une station de radio générative qui produit en continu de la musique *Techno*, et dont le programme fonctionne sur un appareil portable ou via une application sur smartphone. L'idée se base sur le fait que Petr Serkin, étant un producteur de *Techno* depuis longtemps, dispose de beaucoup de motifs et d'automatismes à utiliser durant la composition, et peut donc les assembler et les modifier avec différents algorithmes, en introduisant une part d'aléatoire, pour créer un flux infini de musique [28].

Certaines entreprises proposent aussi des flux infinis de musique *Techno* générative, comme *Mubert*, qui propose une application et des solutions d'entreprises pour disposer d'un flux de musique libre de droit et générée en temps réel. On peut en revanche objecter que ces musiques restent dénuées d'humanité (au sens propre, étant générées entièrement par algorithme, comme au sens figuré, puisqu'elle n'ont pas pour vocation de transmettre la moindre émotion). Un autre problème d'ordre éthique est posée par le fait de prendre une musique issue de la contre-culture et dont certains acteurs revendiquent une remise en question de la surconsommation pour en faire une "solution d'entreprise".

Chapitre 5

L'improvisation algorithmique

Si l'on se réfère au sens strict de "l'algorithme" comme une suite de règles et d'instructions, on peut trouver plusieurs logiques algorithmiques dans l'approche de l'improvisation dans des styles de musiques très divers. En revanche, si l'on a pu faire, au chapitre précédent, la distinction entre la logique algorithmique avant l'ordinateur et après l'invention de celui-ci, il ne paraît pas pertinent d'effectuer cette distinction dans le cas du temps réel : les algorithmes étant, en musique, des processus de composition, il est compliqué pour un humain de les utiliser formellement dans le cadre d'une improvisation.

Les musiciens, même s'ils suivent instinctivement des processus ("la grille est en La mineur donc je vais jouer en La mineur"), ne formalisent pas ces pensées sous la forme d'une suite d'instructions. Cela ouvre la voie vers une question majeure dans le cadre de la réalisation d'un système d'improvisation algorithmique : comment peut-on formaliser la pensée d'un musicien qui improvise pour en faire une suite d'instructions claires. Cette question permet d'explorer des manières de faire d'un logiciel d'accompagnement une sorte de "musicien robotique".

Un musicien qui désire improviser peut ainsi être accompagné par des logiciels, et il existe dans ce but plusieurs solutions incorporant des logiques de composition al-

gorithmique en temps réel. Nous distinguons en revanche ce qui relève de l'accompagnement simple, qu'il s'agisse de solution pour simplifier l'improvisation (comme certains "Digital Work Stations" et solutions à visée d'entraînement musical), et ce qui relève de l'interaction musicale à visée créative entre un musicien et un logiciel.

5.1 L'improvisation

S'il est difficile de définir succinctement l'improvisation, on peut en poser une définition simple qui rend compte de la manière dont l'improvisation est perçue dans la performance musicale. Il s'agit de jouer quelque chose qui n'est pas précisément écrit au préalable. Certains musiciens et théoriciens de la musique opposent ainsi création et improvisation : l'improvisation serait une forme réduite de la création où l'on utilise un langage musical préalablement acquis en temps réel [29].

On pense d'emblée à la pratique du "solo" improvisé en musique actuelle, dans laquelle un musicien soliste improvise une partie sur une grille prévue à l'avance jouée par les musiciens qui l'accompagnent. L'improvisation peut aussi être collective, c'est-à-dire dépasser le simple cadre d'un musicien qui improvise en étant accompagné par des musiciens qui suivent une partition, et c'est dans cette idée que s'inscrit notre module. On retrouve fréquemment dans les sessions d'improvisation une logique selon laquelle un soliste "guide" le reste des musiciens, qui improvisent aussi en essayant de suivre les mélodies qu'il développe ou la narration qu'il crée. C'est une des idées du *Free Jazz*, qui s'oppose au *Jazz* classique dans laquelle le musicien improvise sur une trame collective.

De la même manière que la composition par un humain est régie par des règles préalables et une part d'aléatoire, l'improvisation humaine est en grande partie basée sur des règles musicales pré-établies.

L'improvisation instrumentale est souvent enseignée aux musiciens en leur recommandant de se baser sur des gammes précises et en utilisant des modes déterminés, dont la nature est relative à la grille harmonique du morceau sur lequel ils improvisent. Par exemple les guitaristes, dans les genres comme le *blues*, sont encouragés à travailler la gamme pentatonique, afin d'en avoir une certaine maîtrise musculaire instinctive, pour pouvoir jouer rapidement divers motifs musicaux sans qu'ils soient écrits à l'avance. Ils travaillent souvent des petits motifs musicaux, appelés "licks", qu'ils peuvent lier les uns aux autres dans une improvisation, ou bien faire varier. Il existe donc de nombreuses règles tacites d'esthétiques dans l'improvisation [29], qu'elles soient narratives (l'improvisation doit raconter une histoire, donc varier dans le temps), harmoniques (il est fréquent, en *jazz* par exemple, de placer des tensions en utilisant certains modes sur des cadences particulières), ou rythmique (il reste relativement important pour le musicien qui improvise de garder une cohérence rythmique vis-à-vis de l'ensemble qui l'accompagne).

Une autre règle tacite est celle de la répétition. Si la répétition est considérée comme un élément essentiel de toute musique (les morceaux dont aucun motif ne se répète sont très rares), elle est d'autant plus importante dans l'improvisation. On constate fréquemment que les musiciens répètent et développent des motifs improvisés, un aspect qui permet aisément son utilisation dans la musique *Techno* où l'usage intensif de la répétition est largement répandu et constitue presque un code musical.

5.2 Les logiciels d'improvisation algorithmiques

L'accompagnement d'un musicien peut être exécuté par un programme qui va utiliser des algorithmes pour créer ou bien jouer de la matière musicale en temps réel. Cette utilisation des algorithmes s'inscrit dans l'idée des "Interactive Music System", qui permettent de la composition interactive.

Certains logiciels permettent d'accompagner un musicien qui improvise soit en fournissant une base musicale, soit en jouant un instrument précisément, permettant de simuler d'autres musiciens avec lesquels interagir. On appelle ces logiciels des "Automatic Accompaniment Systems" [30]. Ces logiciels sont souvent pensés pour accompagner un musicien lors d'une session individuelle d'improvisation, à des buts d'entraînement ou de création.

On peut ainsi distinguer les "AAS" avec partitions et sans partitions.

Les "AAS" avec partitions se basent sur une partition indiquée au logiciel au préalable. C'est par exemple le cas du logiciel *Band in a Box* [31]. Il s'agit d'un logiciel conçu pour la musique actuelle et le jazz qui génère de la musique jouées par divers instruments en fonction d'une grille d'accords donnée. Il propose ainsi différents arrangements musicaux et des grilles d'accords alternatives. *Band in a Box* n'improvise cependant pas à proprement parler et sa visée reste d'aider le musicien à travailler l'improvisation (et non pas d'être utilisé dans une performance).

Il y a aussi le cas d'*ImPact*, un logiciel qui génère une ligne de basse en suivant une grille d'accord établie à l'avance, à partir d'une base de données importantes de fragments de ligne de basse. Son fonctionnement est relativement plus complexe que celui de *Band in a box*, car chaque fragment de musique dont il dispose dans sa banque de données est associé à des caractéristiques sonores plus ou moins abstraites, comme la densité musicale ou le contexte d'utilisation [30].

On peut aussi penser à D'Accord Guitar, dont le but est d'accompagner un guitariste à l'improvisation en essayant de jouer des notes "plausibles", c'est-à-dire qu'un guitariste aurait réellement pu jouer en fonction des limitations techniques du jeu sur l'instrument. Le logiciel, dans le but de se rapprocher d'un jeu humain, imite aussi la manière de jouer d'un guitariste en choisissant la corde la plus "logique" à jouer pour une note donnée [32]. Ce fonctionnement se rapproche de systèmes cherchant à jouer un instrument précisément en fonction de contraintes, et s'éloigne de la logique d'accompagnement pour aller vers celle d'un instrument autonome.

Les "AAS" sans partitions reposent soit sur l'écoute du musicien, soit sur des choix autonomes. Ils sont plus indépendants et leur fonctionnement dépend moins d'un paramétrage préalable concernant la performance. Le plus ancien de ces systèmes est le *Voyager* du musicien américain Georges Lewis, réalisé dans les années 90. Ce système, que son concepteur décrit comme un "orchestre d'improvisation virtuelle", se base sur un système d'algorithme qui fonctionne avec des règles pour interagir avec un musicien. Sa particularité est d'avoir intégré autant des considérations technologiques dans sa conception que des aspects socioculturels renvoyant à la composition musicale. George Lewis vise ainsi à s'inscrire musicalement dans la continuité de la culture afro-américaine. L'autre caractéristique est que ce système vise précisément à interagir avec George Lewis lui-même. La conception des règles et leur organisation relèvent ainsi d'une démarche de création artistique.

Certains systèmes se basent en outre sur une approche de l'algorithme par apprentissage. Le Continuator, développé par François Pachet chez Sony CSL, est un projet qui cherche à fournir des outils qui permettent à un "AAS" d'écouter un musicien jouer et d'en repérer des traits stylistiques, afin de pouvoir les imiter. Les possibilités sont larges : on peut ainsi se constituer un "AAS" qui joue en développant les codes

que l'on a établi. Pachet mentionne plusieurs modes de fonctionnement issus du projet, qui permettent plusieurs manières d'envisager la collaboration entre le musicien et l'"AAS".

Le système "Band out of a box", nommé ainsi en référence au système "Band in a box", se base sur une interaction musicien-ordinateur dans laquelle le programme répond au musicien après avoir l'avoir écouté. Il s'agit d'ajouter au fonctionnement basé sur une partition de Band in a Box une logique d'interaction par apprentissage. On peut enfin citer le logiciel *OMax*, développé par l'IRCAM et décrit comme un environnement logiciel d'improvisation algorithmique. *OMax* se base sur de l'apprentissage en temps réel, auquel il ajoute une "représentation sémantique" (c'est-à-dire une communication avec le musicien basée sur le langage de la notation musicale) et divers algorithmes de générations de motifs musicaux en temps réel. Des évolutions notables ont été ajoutées au logiciel comme le système *ImproteK* qui permet d'intégrer des scénarios musicaux dans son fonctionnement, ainsi que *SoMax* qui ajoute la réaction à un contexte extérieur et permet l'idée de "co-improvisation" avec le musicien.

Concernant les travaux les plus récents, les recherches de l'IRCAM ont récemment conduit au projet DICY2. Il s'agit d'un projet qui consiste à créer différents outils d'improvisation algorithmique en collaborant avec des musiciens. Ces outils ont pour but de mélanger différentes approches de l'improvisation et la collaboration musicien-machine [33].

On peut aussi mentionner le système PyOracle, réalisé à l'Université de Californie de San Diego, qui utilise des algorithmes pour analyser un flux musical et improviser en collaboration avec un musicien [34].

Deuxième partie

Partie théorique : Apports

Chapitre 6

Motivations

La problématique de ce mémoire est de réaliser un modèle théorique qui puisse permettre de répondre à quatre besoins. Le premier concerne l'automatisation des tâches qui représente un problème récurrent quand on considère que les performances sont le fait d'un seul musicien manipulant plusieurs instruments. Le second est plus spécifique à la *Techno* et concerne la logique de surprise et de répétition formant la base de l'expérience d'écoute de cette musique. Le troisième renvoie aux possibilités créatives induites par le fait de concevoir des systèmes d'instructions pour un système d'accompagnement musical automatique. Le quatrième renvoie enfin à l'aspect modulaire. Le modèle théorique, et le prototype que nous avons réalisé prennent la forme de ce que nous appelons un "Module d'improvisation Algorithmique Interactif".

6.1 L'automatisation des tâches

Certaines actions accomplies par un musicien lors d'une performance de musique *Techno* sont très simples et logiques. Le fait par exemple d'appuyer sur un bouton pour déclencher le "kick"¹ à l'arrivée du "drop"² est une tâche facile et rapide à exécuter. Le fait d'avoir à y penser contribue cependant à ce que l'on pourrait appeler la "charge mentale" du musicien. Or, il existe des manières de simplifier ces gestes ou de les prévoir à l'avance.

Pour reprendre l'exemple du lancement du "kick", on peut simplement utiliser les réglages de "quantization"³ générale du logiciel avec lequel on effectue la performance pour pouvoir envoyer l'instruction quelque temps avant qu'il soit déclenché. On peut aussi mettre en place des moyens logiciels simples pour indiquer que l'on va lancer le "kick" après un certain délai. Mais cela revient seulement à étaler l'exécution des tâches dans le temps.

On peut ainsi se demander s'il est nécessaire que le musicien donne cette instruction et s'il existe des moyens pour un programme de décider lui même de donner cette instruction en fonction d'un contexte musical. Automatiser des tâches peut donc répondre à un besoin d'ergonomie dans une performance, et permettre d'alléger la charge mentale du musicien et de simplifier certaines tâches rébarbatives. Comme le dit Lucas Moinet que nous avons interrogé dans la section 3.4, "Le problème quand on fait un live⁴ tout seul, c'est qu'on n'a pas cinq bras".

Ainsi l'automatisation permet d'envisager une performance d'une manière plus conceptuelle, en se concentrant moins sur l'exécution de petites actions et plus sur de la

1. Un "kick" ou "kickdrum" correspond à un pied de grosse caisse.

2. Le drop est le moment d'un morceau d'*EDM* où l'intensité est maximale.

3. La quantization consiste à déplacer des notes, des marqueurs ou des instructions temporellement pour coller à une grille rythmique.

4. Une performance.

composition en temps réel. On peut même, de manière plus abstraite, automatiser le processus de composition pour que le musicien se concentre plus sur ses intentions musicales. L'instruction "lancer le kick à la fin de la mesure" (dans le cas d'un simple retardement d'instruction) peut devenir "lancer le kick dans un nombre donné de mesures" (dans le cas d'une instruction planifiée de temps plus large) et enfin "à un moment donné, faire un drop" (dans le cas d'un programme plus indépendant). En déléguant les étapes les plus fines de la composition comme l'agencement des notes, on peut ainsi permettre au musicien d'interagir avec un matériau musical plus riche et moins prévisible.

6.2 Surprise, danse et performance

La musique Techno se base, dans son expérience d'écoute, sur une opposition entre surprise et répétition. Étant donnée qu'elle a pour vocation d'être dansée, elle repose sur une logique de "drop". Le jeu entre le musicien et le public s'effectue souvent, d'une manière plus ou moins prévisible, sur le moment où le "drop" est lancé, et le public peut ainsi se mettre à danser.

De très nombreux titres jouent sur cet aspect. On peut citer *Use Me Again* de Tom Trago (2012) dans lequel la voix est allongée et le tempo ralenti juste avant que le "drop" tombe, ce qui donne un effet de retard pour le public qui entend le morceau. On peut aussi citer *Que Belle Époque* de Ricardo Villalobos (2006), dans lequel le placement du clap qui marque le second temps échange avec le premier temps à plusieurs reprises, ce qui crée une sorte de malaise pour l'auditeur qui danse souvent en se basant sur un cycle à quatre temps.

Des motifs sont donc mis en place, parfois pour des périodes très longues (certains styles, comme la *Micro House* disposent dans leur répertoire de morceaux très mi-

nimalistes dans leur composition et d'une durée souvent supérieure à 10 minutes), dans l'idée de poser un contrat avec le public : il faut lui permettre de danser sur quatre temps et d'anticiper les moments où il va intensifier ses mouvements pour suivre la musique. Mais de nombreux musiciens jouent avec ce contrat, en changeant subitement le tempo, ou en rajoutant des temps momentanément, dans le but de briser la monotonie de la danse et de renouer ce contrat. L'imprévisibilité de la musique permet aussi au public d'être dans une forme d'écoute active, car il ne peut pas se contenter de suivre simplement le mouvement pour danser : il faut qu'il s'y adapte et qu'il y réagisse.

C'est dans cette logique qu'il nous paraît pertinent d'introduire des changements plus ou moins aléatoires dans la performance à l'aide d'un module algorithmique. Cela permet de mettre le musicien qui exécute la performance dans une posture similaire à celle du public, puisqu'il ne peut pas être sûr de ce que le module va fournir, et doit donc renouveler le contrat et retrouver ses marques quand le module lui impose un changement bref.

Cela permet aussi de proposer des variations au musicien. Les performances de *Techno* et *House* durent en général entre une et deux heures, et les changements dans ces musiques se faisant souvent sur des cycles de 16 ou 32 mesures, cela équivaut, à 120 BPM environ, à entre 30 et 60 occasions de jouer sur l'aspect répétitions/surprises par heure. Il est donc difficile de proposer systématiquement quelque chose de nouveau.

Le hasard, quand il est utilisé en art, peut, au delà d'un outil pour lever une indécision, permettre de fixer des règles arbitraires qui donnent un support pour la création. Certains mouvements artistiques se basent ainsi sur l'introduction de règles arbitraires comme l'OuLiPo, un mouvement littéraire du XX^{ème} siècle. Au delà de

l'aspect "imprévu", la contrainte arbitraire est riche de possibilités. Il paraît donc intéressant, étant donné le nombre de changements à effectuer, d'utiliser le hasard et l'algorithme pour établir des règles arbitraires et permettre au musicien d'avoir plus rapidement des idées créatives.

6.3 Règles et création

Comme le montrent la musique sérielle ou les différents travaux algorithmiques du XX^{ème} siècle, la création de règles peut être vue comme un processus artistique à part entière. Une des idées de début de ce mémoire était de transposer à la performance musicale le système de "Gambits" issu du jeu *Final Fantasy XII*, publié par *Square Enix* en 2006.



FIGURE 6.1 – L'écran de configuration des "gambits" de *Final Fantasy XII*.

L'action du jeu se déroule en temps réel, et le joueur doit donner en temps réel des instructions aux trois personnages qu'il contrôle. Il peut, à l'aide d'instructions simples ordonnées par ordre de priorité, laisser les personnages agir seul lors des

combats. Ces instructions peuvent être conditionnelles, et si leur utilisation n'est pas obligatoire dans le jeu, elles permettent de simplifier de nombreuses actions et de rendre les personnages plus réactifs. Le caractère ludique de la constitution de ces règles et la personnalité qu'on pouvait percevoir dans les actions d'un personnage pourtant programmé succinctement à l'avance m'ont paru être des pistes intéressantes pour penser une manière, pour le musicien, de se concevoir des assistants. Il n'est pas nécessaire de s'y connaître précisément en programmation pour formaliser des règles et des instructions, et pour expliquer des formes d'intentions.

L'idée de personnalité évoquée précédemment est importante : le rapport entre répétition et surprise peut s'apparenter à une forme d'espièglerie de la part du musicien, et l'ajout d'aléatoire et de décisions "impulsives" ou contre-intuitives fournissent des pistes pour humaniser un assistant avec des règles simples. On peut ainsi penser, dans ce rapport entre personnalité perçue et règles suivies par un algorithme, à l'exemple du jeu *Rimworld*, publié par *Ludeon* en 2018, dans lequel les événements qui rythment le jeu sont gérés par des "raconteurs", donc des algorithmes, qui sont présentés avec des traits humains et des personnalités lors de la création de la partie. Ces "raconteurs" prennent des décisions pour influencer le cours du jeu, comme l'apparition de perturbations climatiques, en fonction de la situation du joueur. L'un d'eux, *Randy Random*, qui correspond simplement à un mode de jeu où tous les événements sont produits de manière aléatoire sans aucune considération pour leur impact sur la difficulté de la partie, est présenté visuellement comme étant fou.



FIGURE 6.2 – Le visuel du "raconteur" Randy Random de *Rimworld*.

On voit ainsi la possibilité, avec des règles plutôt simples, pour un module algorithmique, de dépasser le simple statut d'outil et de s'humaniser, permettant au musicien de voir sa performance comme une collaboration, et au public, si les enjeux sont bien communiqués, de voir la performance comme un jeu entre le musicien et ses modules. A ce sujet, Joel Chadabe, un compositeur américain, écrit à propos de son système algorithmique Coordinated Electronic Music Studio (CEMS) "J'étais, en

pratique, en train de converser avec un instrument qui paraissait avoir une personnalité propre et intéressante" ⁵ [35]. Il semble que l'humanisation de l'intelligence artificielle soit un enjeu récurrent et porteur de nombreuses possibilités créatrices.

6.4 La logique modulaire

Les performances de *Techno* sont souvent construites, comme précédemment expliqué, autour de différents modules. Ces modules sont rarement acquis tous en même temps par le musicien, et sont intégrés au fur et à mesure dans la construction de la performance. Il est courant de voir des musiciens réorganiser leurs performances après l'acquisition d'une nouvelle machine, ou d'un nouvel élément (un instrument, un changement de séquenceur...), tout en gardant une grande partie de la charnière précédemment utilisée, qu'il s'agisse du matériau musical ou de la manière d'exécuter la performance.

Cette logique se retrouve à l'extrême dans ce que l'on appelle les performances modulaires, pour lesquelles les musiciens associent de nombreux modules dont les possibilités sont extrêmement limitées. Ces modules électroniques sont connectés les uns aux autres par des câbles jacks dont le branchement évolue parfois durant la performance. Les dispositifs utilisés sont visuellement impressionnants de complexité, mais permettent paradoxalement très rapidement l'inclusion de modifications.

5. Traduction de l'auteur.

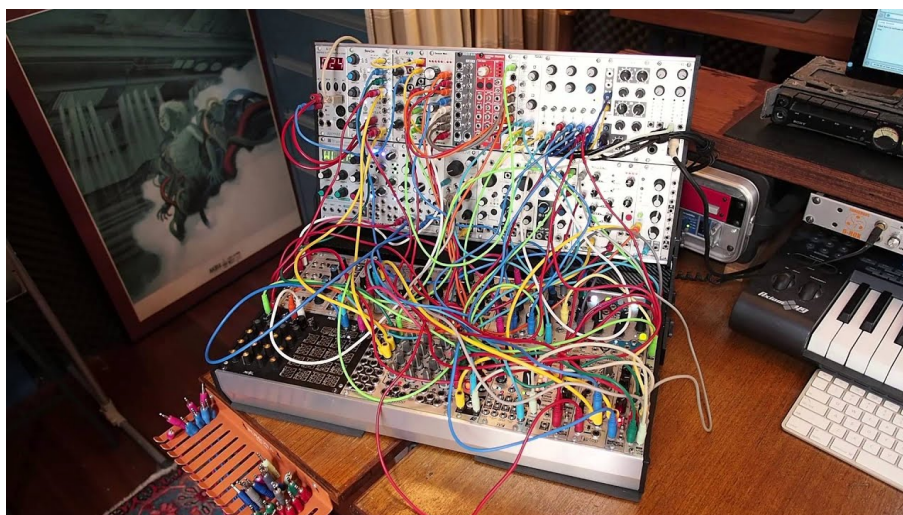


FIGURE 6.3 – Un dispositif de performance modulaire réalisé par Luke Killen.

Il paraît donc intéressant de conceptualiser un outil que l'on puisse facilement intégrer dans une performance, sans qu'il n'y ait besoin de la repenser intégralement. Il s'agit donc à la fois d'alléger le travail en temps réel et de proposer des idées au musicien sans pour autant prendre une part majeure de la performance. L'idée est qu'utiliser un tel module revienne, pour le musicien qui effectue la performance, à demander à un autre musicien de prendre en charge l'une des sections musicale de la performance en lui expliquant les critères et le langage musical spécifique à la performance.

Il faudrait donc que ce module s'inscrive dans la performance sans la changer profondément. Les précédents travaux sur l'algorithme dans l'improvisation font que les systèmes d'accompagnements occupent une part majeure de la performance. Il nous paraît pertinent de créer des petites unités que le musicien peut combiner, et qui pourraient éventuellement communiquer entre elles. Nous avons choisi de réaliser un module car il est aussi plus simple, pour s'inscrire dans une démarche de recherche, de travailler sur un petit composant dont la conceptualisation pourra être reprise pour des travaux ultérieurs. En outre, le fait d'utiliser un petit élément a

moins d'impact potentiel sur les partis pris artistique de la performance.

6.5 Objectifs

Une des difficultés lors ce que l'on traite de la musique algorithmique est la multiplicité des termes utilisés par les chercheurs et les artistes, ainsi que le manque de distinction entre ces termes [36]. Ces sous-familles comportent elles-mêmes diverses approches qui peuvent être spécifiques ou non, et combinées ou non, qu'il s'agisse des approches dans la conception de l'algorithme ou encore des approches de compositions.

L'essentiel du travail de ce mémoire est donc, plus que la mise en œuvre d'un module, de fournir un inventaire théorique des méthodes qui peuvent être utilisées pour réaliser un module. L'intérêt est qu'avec un aperçu général suffisamment étoffé, on puisse facilement choisir comment réaliser le module que l'on souhaite en fonction des contraintes imposées par la nature de la performance ou de critères esthétiques et artistiques. Il y a plusieurs aspects à considérer pour réaliser un tel module. Le matériau musical que le module manipule, l'interaction avec le musicien, et enfin la manière dont le module effectue sa prise de décision.

Nous utiliserons le terme de "matière musicale" pour décrire de manière large tout contenu audio, MIDI, ou de partition, et le terme "motif" pour désigner tout morceau audio, MIDI ou de partition qui constitue ce matériau musical. La réalisation d'un tel module amène à réfléchir autour des règles que l'on choisit en fonction du style de musique et des choix artistiques que l'on veut adopter, et nous avons donc constitué un corpus de règles spécifiques à la *Techno*.

L'idée de faire de ce module un module algorithmique renvoie à son indépendance durant la performance. A la manière d'un musicien, qui connaît les règles et le lan-

gage musical, il doit pouvoir prendre lui-même des décisions en fonction de ses règles internes et générer du contenu musical cohérent. L'algorithme permet aussi, au-delà de le rendre indépendant, de le rendre partiellement imprévisible avec l'inclusion de règles aléatoires. Il doit jouer des choses nouvelles et proposer des idées musicales au musicien. Il se pose alors la question du dosage de cette imprévisibilité. Selon la fréquence des changements aléatoires, et l'ampleur qu'ils peuvent prendre, le module peut assumer différents rôles, allant du simple assistant du musicien au partenaire, jusqu'à en être en quelque sorte l'adversaire durant la performance s'il le met en difficulté. Cet aspect algorithmique prend ici la forme d'un algorithme par règle : il nous semble plus accessible pour des musiciens de pouvoir formuler des règles simples, plutôt que d'interagir avec la conception d'un algorithme par apprentissage.

Ces différentes logiques permettent au module de s'inscrire dans la performance d'une manière qui permette de simplifier et d'aider l'improvisation, ce qui amène à lui proposer un nom : le module d'improvisation algorithmique interactif

Chapitre 7

La matière musicale

Une question importante, étant donné qu'il s'agit d'un module destiné à produire de la musique, est celle de la matière que le module manipule. Cette matière peut prendre deux formes principales : Il peut s'agir d'un signal audio ou d'un contenu plus abstrait permettant de se représenter la musique et qui relève de l'écriture musicale et de l'instruction de jeu. On peut en effet considérer que tout processus de composition fait appel, à un moment donné, à un système de représentation quel que soit son niveau d'abstraction [37]. La création de musique peut donc se faire sur plusieurs plans. On peut manipuler et créer directement du son sous forme de signal audio. On peut aussi rester sur un plan plus abstrait et manipuler une partition ou son équivalent. L'ampleur de cet abstraction est variable : on peut ainsi agencer des notes (ce qui est le cas le moins abstrait) ou manipuler des informations plus larges, comme des idées de mouvements. Le MIDI représente la forme la plus courante de cette abstraction, et il est facilement utilisable pour faire jouer du signal audio par des machines et des logiciels.

On peut générer directement de la matière sonore, générer directement une partition, ou générer une partition et manipuler cette dernière et ensuite manipuler la matière sonore qu'elle a créée. On note cependant, de manière logique, que la géné-

ration abstraite précède toujours la génération de matière sonore.

On peut distinguer deux manières d'envisager la musique algorithmique en ce qui concerne le matériau musical :

- Il est possible de générer, ou simplement d'exécuter de la musique en utilisant des fragments rangés dans des banques de données diverses, qu'il s'agisse d'échantillons sonores, de morceaux de partitions, ou de morceaux de musique entiers. Dans ce cas, une grande part de la réflexion autour de la conception de l'algorithme se fait autour de la sélection de fragments dans la banque de données. On peut ainsi réfléchir à une nomenclature pour ces fragments, et à une manière pour l'algorithme de sélectionner ces fragments.
- Il est aussi possible de générer purement de la musique. La réflexion s'oriente alors autour des manières de structurer la création de ce contenu musical. Elle peut ainsi être générée *ex-nihilo*, avec diverses règles de composition, ou bien être obtenue par transformation ou combinaison.

Cette distinction n'est cependant pas absolue. La différence entre une musique générée à l'avance composée de grand motifs musicaux, et une musique qui peut être générée durant la performance en utilisant des fragments de motifs tient finalement dans la taille de ces fragments. Un motif peut toujours être vu comme un fragment d'un motif plus long. Ces deux manières reflètent les deux théories principales sur le fonctionnement cognitif de l'improvisation musicale chez les musiciens. L'une stipule en effet que les musiciens font appel à un corpus de motifs musicaux qu'ils arrangent et combinent [38], tandis que l'autre affirme que les musiciens se basent avant tout sur le contexte tonal, rythmique et mélodique pour créer des motifs [39].

7.1 La musique préalablement créée

Dans le cas où le module choisi, il faut au préalable lui fournir une "banque" de motifs musicaux. Cette banque peut être intégralement réalisée par l'artiste, ou générée via un algorithme. Il est possible d'utiliser les différentes techniques de génération que nous allons évoquer dans la génération durant la performance pour créer un certain nombre de variations autour des motifs créés au préalable.

L'algorithme qui accède à des motifs musicaux doit le faire en tenant compte d'un contexte. Ainsi, le cas de l'utilisation par un algorithme d'une banque de motifs musicaux amène à réfléchir autour de la nomenclature utilisée pour que le module puisse distinguer les différents motifs auxquels il a accès.

Certains critères apparaissent naturellement : la nature rythmique, la tonalité et la grille d'accords. Dans le cas d'un signal audio, la nature de l'instrument qui a été utilisé pour jouer le motif s'impose comme un critère de sélection utile.

D'autres critères permettent au musicien de constituer une banque de donnée en fonction de la fonction narrative des motifs musicaux. On peut mettre en place un système de notation qui rende compte du mouvement effectué par un motif (augmentation d'une tension, montée vers l'aigu, etc..), et d'un contexte d'utilisation (introduction, développement, outro, etc..).

Enfin, certains critères d'un ordre plus abstrait, peuvent être utiles pour guider le module et l'aider à déterminer des intentions pour interagir avec le musicien. Ces critères peuvent se baser sur une description musicale abstraite du contenu (le motif est minimaliste, ou au contraire saturé d'éléments). Ils peuvent aussi renvoyer à l'esthétique perçue du motif : par exemple, un musicien, dans le rangement de motifs concernant une ligne de basse, pourrait noter que certains sont dans une esthétique *Disco*, tandis que d'autres sont plus agressifs et enfin que certains sont dans une lo-

gique calme et mélodique renvoyant à une esthétique de *Deep-House*.

Dans le cas de matière préalablement générée se pose la question de la constitution d'un "cartouchier" (pour reprendre le terme utilisé pour désigner les logiciels qui permettent de lancer des sons en radio). Si la matrice de la vue session d'*Ableton* représente une forme de cartouchier, elle est relativement limitée en possibilités. D'abord, car elle est liée au fonctionnement du transport général d'*Ableton*. Lancer une "scène" (une ligne entière) lance tous les clips disposés sur les pistes, et coupe les clips qui jouaient disposés sur une autre ligne. Ensuite, car chaque piste ne peut ordonner ses clips que dans une seule dimension. On pourrait imaginer un système dans lequel chaque module, dispose, en interne, de sa propre matrice, et où l'on disposerait horizontalement les variations d'un motif dans un cas donné. La position verticale donnerait le contexte d'utilisation de chaque lignes de motifs. Si cette vision n'est pas nécessairement utile dans la conception de l'algorithme lui même, elle permettrait au musicien d'agencer facilement et instinctivement les différents motifs qu'il choisit d'utiliser. Un système matriciel permet, en outre, en terme d'interface, d'incorporer de nombreuses manières pour le musicien de mettre en place des règles. Ces règles peuvent prendre la forme d'attributs individuels pour chaque motif (représenté, par exemple, par des couleurs), ou bien de "fils" et de "flèches" qui pourraient imposer des règles directionnelles et former une sorte d'arbre décisionnel (si l'on joue tel motif, jouer les deux motifs ensuite juste en dessous à droite, par exemple). Cette mise en place visuelle des règles paraît être une piste ludique et intuitive.

La sélection comporte un autre aspect que celui du choix des motifs. Si plusieurs motifs sont joués, on peut se poser la question de la combinaisons de ces motifs et

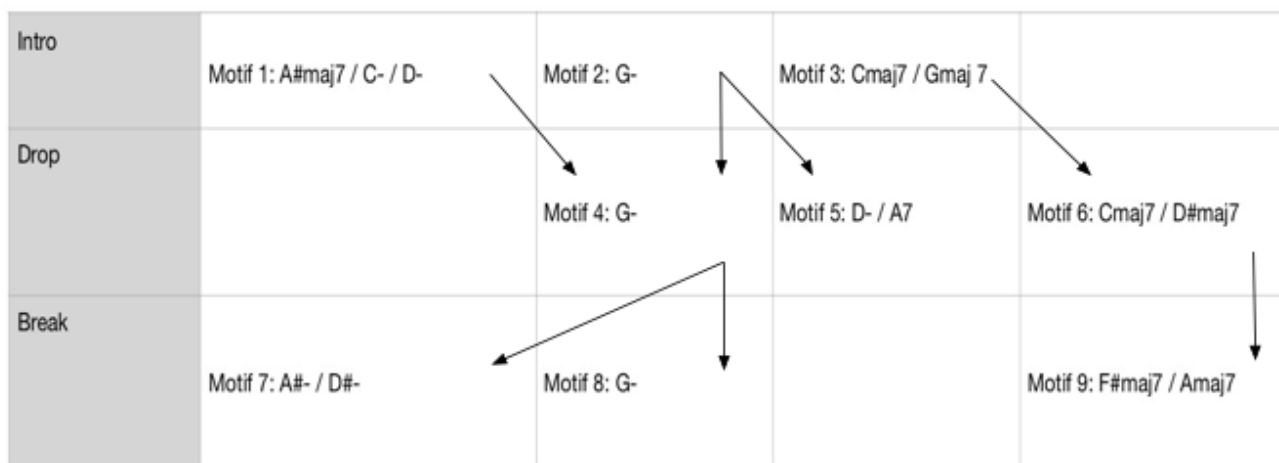


FIGURE 7.1 – Schéma de l’interface d’un cartouchier matriciel. Les flèches représentent des règles d’organisation (après tel motif, jouer tel motif). Les variations sont ici classées selon leur grille d’accords.

des règles qui en découlent. Cette idée de combinaison est aussi présente dans le cas où un seul motif est joué, car un motif long peut être créé par combinaison à la suite de motifs courts. C’est d’ailleurs l’idée majeure à l’origine des jeux de dés musicaux préalablement évoqués. La combinaison de différents motifs jouant simultanément peut aussi se faire en changeant les instruments qui jouent un motif. L’appareil musical portable *MadPlayer*, développé par Doctor Mad depuis 1998, utilise entre autre cette technique [40]. Elle a pour intérêt d’être relativement simple à mettre en place, puisqu’il suffit de rendre possible le changement d’instrument qui joue un motif MIDI. Les règles de combinaisons sont généralement basées sur un système de critères évoqué précédemment, et leur complexité dépendra du niveau de détail de la nomenclature de la banque de matériau musical.

7.2 La musique générée durant la performance

Un des concepts intéressants de cette approche de la composition musicale par l'algorithme est qu'il est souvent fait une distinction entre les algorithmes de génération, de transformation, et combinatoires, auxquels on ajoute parfois les algorithmes stochastiques [41]. L'approche transformative se concentre surtout sur la modification de matériaux pré-existants, en utilisant diverses méthodes de transformation, un procédé analogue à celui que l'on peut retrouver dans les règles de composition de la musique sérielle. L'approche générative, elle, crée directement de la musique, que ce soit de manière algorithmique ou aléatoire. Ces deux approches peuvent nécessiter la compréhension du contexte harmonique et tonal de la partie de la performance en cours. Enfin, l'approche combinatoire se base sur la combinaison temporelle de morceaux de motifs, ou bien le ré-arrangement d'un motif. Les algorithmes que nous décrivons ici renvoient principalement à des méthodes de composition musicale. Ils ne sont pas nécessairement liés à l'informatique.

7.2.1 La musique générée en temps réel : l'approche transformative

L'approche transformative consiste à utiliser des motifs préalablement enregistrés ou écrits, et modifier ces motifs de différentes manières pour obtenir des nouveaux motifs. Ces motifs peuvent aussi être issus de la performance et enregistré par le module. Certaines méthodes permettent de transformer le motif instantanément sans qu'il soit nécessaire de l'avoir en mémoire, tandis que d'autres nécessitent de travailler sur le motif entier. Nous présentons ici des méthodes de manière abstraite en nous concentrant sur la partition tant les façons d'implémenter ces méthodes dans un programme informatique sont variées. Nous avons choisi de représenter

des exemples simples, mais les cheminements des algorithmes peuvent être bien plus complexes et combiner diverses méthodes.

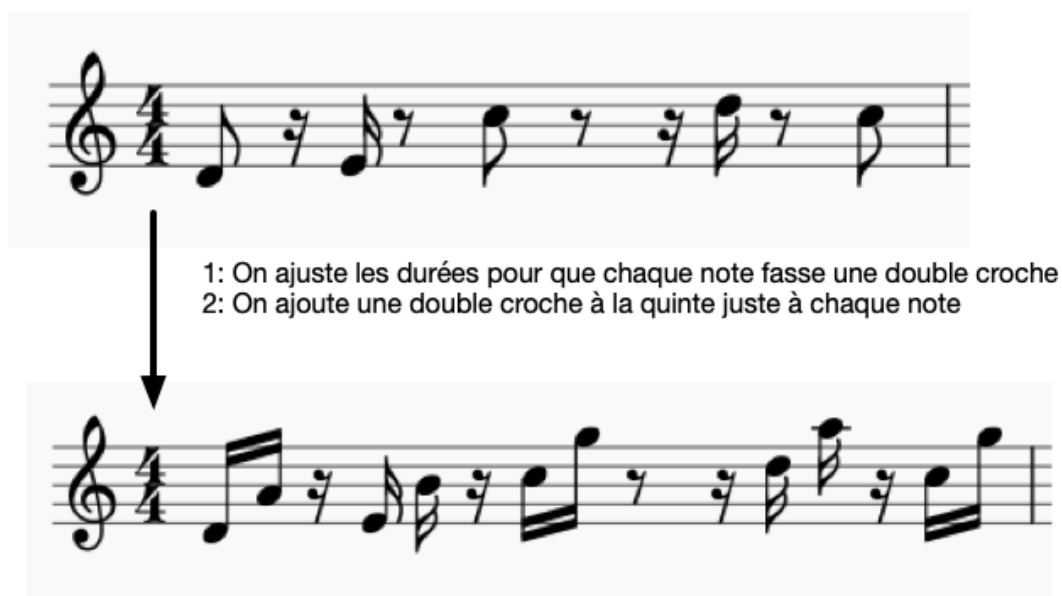
- On peut repositionner les notes en inversant leur position. Ce repositionnement peut se faire en inversant la position horizontale des notes (il s'agit alors d'une inversion rythmique). Il peut aussi se faire en inversant la position verticale des notes, en prenant en compte la hauteur des notes ou leurs intervalles respectifs (il s'agit alors d'une inversion mélodique). L'inversion peut s'accompagner de corrections afin que le motif soit cohérent dans un contexte rythmique ou mélodique. Cette méthode a pour intérêt d'être facile à implémenter et d'être facilement combinée avec le motif originel.

The diagram illustrates the process of creating a musical motif through melodic inversion. It consists of two musical staves in 4/4 time, connected by a downward-pointing arrow. The top staff shows a sequence of seven notes with intervals between them: +3, +1, +1, +2, +1, -2, and +1. A text box in the middle states: "On passe la première note à l'octave au dessus puis on inverse les intervalles entre chaque notes successives". The bottom staff shows the resulting inverted motif, where the first note is moved up an octave and the intervals are inverted: -3, -1, -1, -2, -1, +2, and -1.

FIGURE 7.2 – un exemple de création d'un motif musical avec un algorithme simple d'inversion mélodique.

- On peut ajouter ou retirer des notes à un motif existant. Cela peut se faire en doublant simplement les notes. On peut aussi créer un motif que l'on répète sur chaque note, par exemple, comme dans la figure 7.3 en ajoutant à chaque note sa quinte juste. Il est possible de modifier la durée des notes du motif originel pour garder une cohérence. Dans le cas où l'on retire des notes, on

peut décider que l'on retire des notes selon une période donnée. Cette méthode peut être mise en place en temps réel sans que le module ait besoin d'avoir en mémoire un motif. On peut parfois y retrouver une logique similaire à celle de l'arpéggiation que nous détaillerons ci-après. Il existe en outre des techniques qui permettent de rajouter des ornements sur un motif mélodique. Leur fonctionnement étant très simple (il s'agit parfois simplement de jouer une note au dessus de celle que l'on joue avant de revenir, ou encore de jouer tout les demi-tons contenus entre deux notes successives pour passer de l'une à l'autre) et ponctuel, nous ne présenterons pas un schéma de fonctionnement détaillé. Elles peuvent néanmoins être utilisées pour enrichir les motifs joués par le module.



The figure illustrates a transformation of a musical motif. It consists of two musical staves in 4/4 time, connected by a downward-pointing arrow. The top staff shows a sequence of six notes: a quarter note, a quarter note, a quarter note, a quarter note, a quarter note, and a quarter note. The bottom staff shows the same sequence of notes, but each note is now a half note, and a half note is added to the fifth of each note, creating a richer, more complex sound. The notes in the bottom staff are: a half note, a half note, a half note, a half note, a half note, and a half note. The notes added to the fifth are: a half note, a half note, a half note, a half note, a half note, and a half note.

1: On ajuste les durées pour que chaque note fasse une double croche
2: On ajoute une double croche à la quinte juste à chaque note

FIGURE 7.3 – Un exemple de transformation d'un motif musical avec un algorithme simple d'addition.

— Cette méthode peut être combinée à une "extension" ou une "réduction" temporelle du motif. Il s'agit de ralentir, ou d'accélérer le motif en multipliant ou divisant sa longueur par deux. Il faut ensuite en général ajouter ou retirer des notes pour éviter que le motif obtenu paraisse trop rapide ou trop lent. L'ex-

tension et la réduction ont pour intérêt de permettre d'allonger ou de réduire la structure d'accords si besoin, mais nécessitent de disposer de l'intégralité du motif dès le début. Ces deux méthodes peuvent en outre faire appel à des algorithmes de génération, dès lors qu'il s'agit d'ajouter des notes ou d'effectuer un motif.

1: On multiplie la durée du motif par deux, en doublant les silences et la durée des notes

2: On réduit chaque note pour qu'elle dure une croche pointée
3: On détermine une gamme cohérente pour l'ajout de notes: ici, la gamme de Do majeur
4: On génère arbitrairement des notes sur la gamme de Do majeur
5: On ajoute ces notes d'une durée d'une double croche à chaque note du motif originel

FIGURE 7.4 – Un exemple de transformation d'un motif musical avec un algorithme simple d'extension.

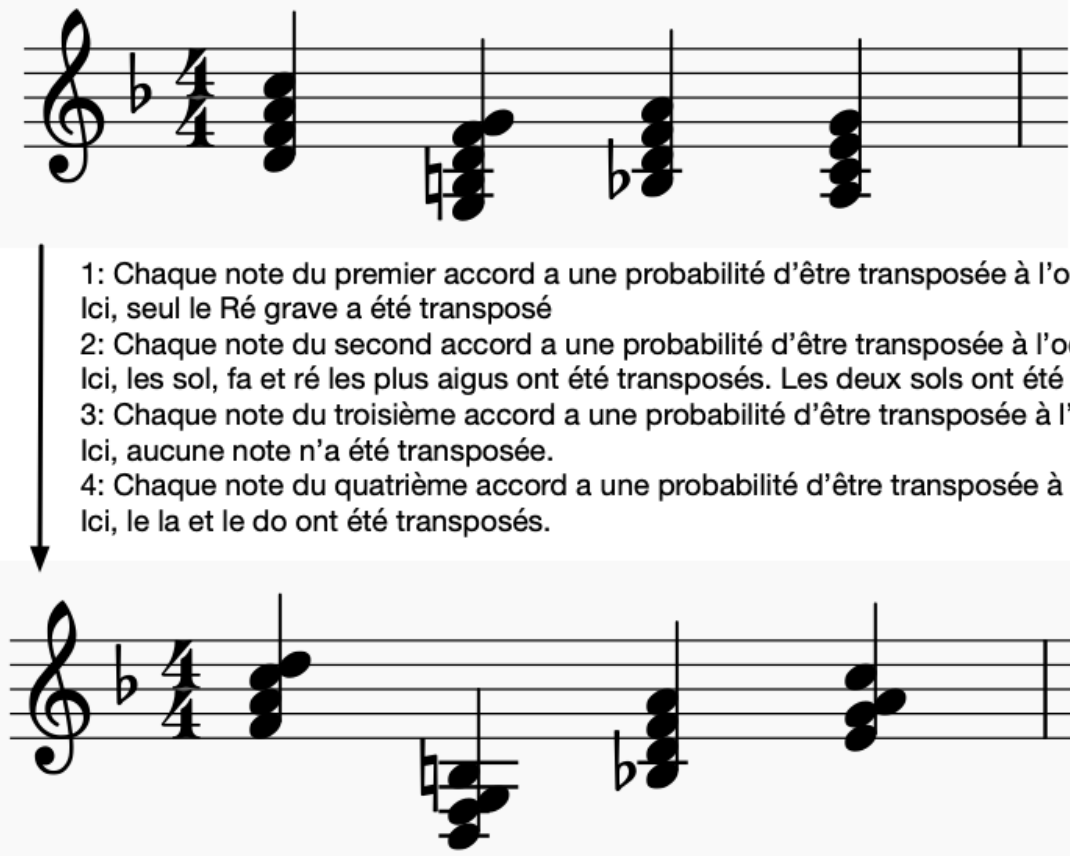
- Une méthode relativement simple consiste à changer le mode d'une mélodie en modifiant certaines notes qui la composent. On peut obtenir des motifs qui ouvrent vers d'autres tonalités. Elle peut facilement être effectuée en temps réel sur du MIDI en interdisant certaines notes et en les remplaçant en temps réel. Elle peut aussi s'effectuer sans connaître la tonalité du motif, en posant arbitrairement une gamme d'arrivée sans prendre en compte la gamme du motif originel.

1: On identifie le mode de la mélodie
On a ici La éolien.
2: On augmente la tierce d'un demi ton
La mélodie est alors dans un mode mixolydien

The image displays two musical staves in 4/4 time, illustrating a modal transformation. The top staff shows a melody in the Lydian mode (La éolien), characterized by a major third interval between the first and second notes. The bottom staff shows the same melody transformed into the Mixolydian mode, where the third note is lowered by a half tone, creating a major second interval between the first and second notes. An arrow points from the first staff to the second, indicating the transformation process.

FIGURE 7.5 – Un exemple de transformation d'un motif musical avec un algorithme simple de changement modal.

— Dans le cas où des accords sont présents, on peut changer la manière dont ils sont joués sans pour autant changer la structure harmonique du motif. Cette technique est fréquemment utilisée par les musiciens eux mêmes. On peut jouer le même accord à une octave différente, ou encore jouer un renversement¹ de l'accord. On peut aussi utiliser des accords de substitution, dont la nature est différente mais dont la proximité harmonique avec l'accord originel permet de garder une cohérence.



1: Chaque note du premier accord a une probabilité d'être transposée à l'octave supérieure
Ici, seul le Ré grave a été transposé

2: Chaque note du second accord a une probabilité d'être transposée à l'octave inférieure
Ici, les sol, fa et ré les plus aigus ont été transposés. Les deux sols ont été confondus

3: Chaque note du troisième accord a une probabilité d'être transposée à l'octave supérieure
Ici, aucune note n'a été transposée.

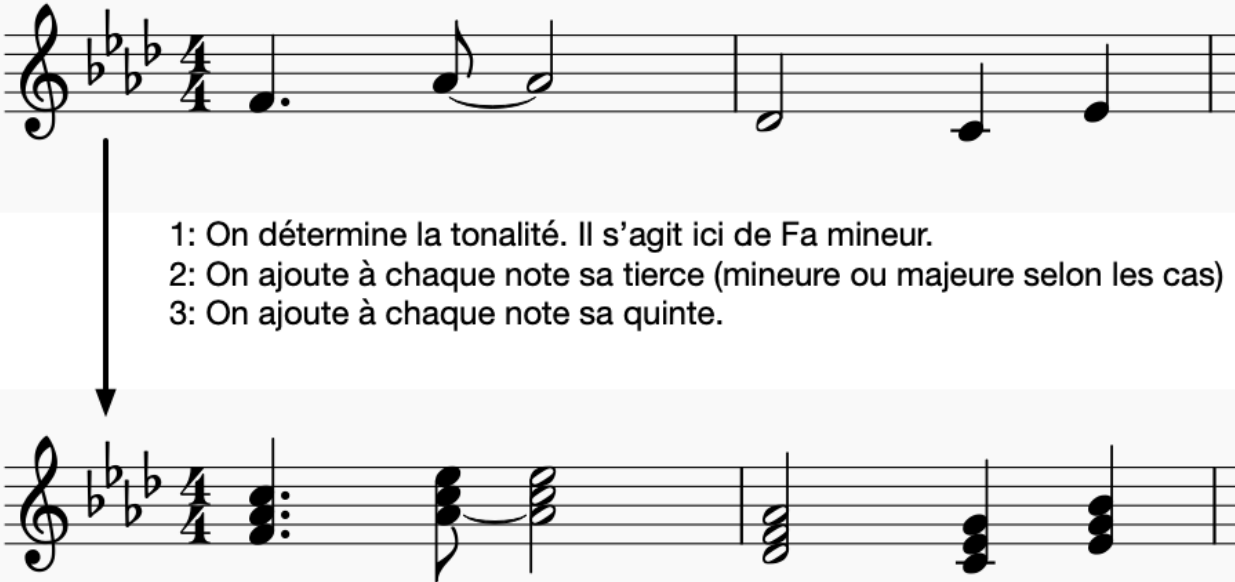
4: Chaque note du quatrième accord a une probabilité d'être transposée à l'octave supérieure
Ici, le la et le do ont été transposés.

FIGURE 7.6 – Un exemple de transformation d'un motif musical avec un algorithme simple effectuant des renversements.

— Dans le cas où des notes simples sont jouées, on peut mettre en place un algorithme qui harmonise chaque note, c'est-à-dire qui rajoute des notes simul-

1. Un renversement d'accord consiste à changer l'ordre des notes constitutives de l'accord en transposant d'une octave vers le haut ou vers le bas certaines des notes qui le composent.

tanées pour lui faire jouer un accord. Cela peut se faire de manière simple en rajoutant des intervalles fixés à l'avance, ou bien en considérant la tonalité et en changeant la nature des accords selon la fonction de chaque note dans la gamme.



The image displays two musical staves in 4/4 time, F minor. The top staff shows a melody: a dotted quarter note on F3, followed by a half note on G3, a quarter note on A3, a quarter note on Bb3, and a quarter note on C4. An arrow points from this staff to the bottom staff, which shows the harmonic accompaniment. The bottom staff consists of vertical stems with chords: a dotted quarter note on F3 with a colon, a half note on G3 with a colon, a quarter note on A3 with a colon, a quarter note on Bb3 with a colon, and a quarter note on C4 with a colon. The text between the staves explains the algorithm used for this transformation.

- 1: On détermine la tonalité. Il s'agit ici de Fa mineur.
- 2: On ajoute à chaque note sa tierce (mineure ou majeure selon les cas)
- 3: On ajoute à chaque note sa quinte.

FIGURE 7.7 – Un exemple de transformation d'un motif musical avec un algorithme simple effectuant une harmonisation.

— On peut enfin, dans l'autre sens, partir d'un motif avec des notes simples ou bien des accords et effectuer une arpégiation. Les arpégiateurs sont des dispositifs répandus sur les différents synthétiseurs. Un arpégiateur peut ainsi partir d'une note et effectuer un arpège montant, descendant, montant, ou encore d'ordre aléatoire. Cet arpège peut se faire avec une certaine hauteur maximum en octaves et différentes périodes. Dans le cas où plusieurs notes sont jouées simultanément, l'arpégiateur va d'une note à l'autre en suivant le mouvement prédéterminé et continue au-delà en ajoutant les octaves. On peut aussi concevoir un arpégiateur qui vise à redescendre précisément sur une note, ou qui fonctionne avec des intervalles précis. Il peut enfin effectuer des motifs plus complexes comme un motif qui monte de deux notes puis descend d'une.

The image displays two musical staves in 4/4 time. The top staff shows a motif consisting of a quarter note G4 followed by a half note chord of G4, B4, and D5. The bottom staff shows the result of arpeggiating this motif: a sequence of eighth notes starting with G4, followed by a descending sequence of eighth notes (F#4, E4, D4), then an ascending sequence of eighth notes (C4, B3, A3), and finally a descending sequence of eighth notes (G3, F3, E3). Two arrows point from the top staff to the bottom staff: one from the quarter note to the first eighth note, and another from the chord to the descending eighth-note sequence.

1: On effectue l'arpège montant à la double croche sur cinq notes
2: On effectue l'arpège descendant à la double croche sur trois notes

FIGURE 7.8 – Un exemple de transformation d'un motif musical avec un algorithme simple effectuant un arpège.

7.2.2 La musique générée en temps réel : L'approche combinatoire

Les techniques de combinaisons sont celles qui se situent le plus à la frontière entre la musique préalablement créée et la musique générée en temps réel, puisque les deux axes sur lesquels la combinaison s'effectue sont aussi deux axes couverts par les techniques de combinaisons de motifs créés au préalable. Elle consiste à découper les motifs et les réarranger temporellement. Cette technique est assez riche en terme de possibilités :

- On peut découper un motif et réarranger les morceaux dans un ordre différent
- On peut découper plusieurs motifs et mélanger leur morceaux entre eux.
- On peut répéter une portion de motif un certain nombre de fois. Le "beat repeat", un effet audio inclut dans *Ableton Live*, fonctionne sur ce principe et permet de répéter une portion de motif, avec divers paramètres, comme la variation de la hauteur du motif pendant la répétition, le "Decay"², ou encore la probabilité de répétition. Il est possible de créer une grande richesse rythmique en effectuant un découpage qui ne soit pas également distribué, par exemple dans le cas de portions d'un nombre impair de temps dans le cas d'une musique de 4 temps par mesure.

Elle a aussi pour intérêt de fonctionner avec du signal audio. Il est ainsi possible d'enregistrer directement des motifs sous forme de signal audio issus de la performance. Cette possibilité est intéressante dans la mesure où certains musiciens basent leurs performances sur l'enregistrement de boucles en temps réel, et donc qu'il est possible de demander à un module de faire varier ces boucles.

2. Le Decay correspond à la décroissance du volume.

1: Les motifs 1 et 3 sont inversés.
 2: Le motif 2 est répété après un quart de soupir
 3: La répétition du motif 2 est tronquée pour éviter le chevauchement avec le motif suivant

FIGURE 7.9 – Un exemple de recombinaison d'un motif musical.

7.2.3 La musique générée en temps réel : L'approche générative

L'approche générative consiste à générer directement une mélodie, un rythme ou une grille d'accord, sans disposer de banque de données préalable. Elle se base souvent sur des tirages aléatoires, mais il est possible d'utiliser des variables externes pour choisir arbitrairement des notes, selon la logique qui sera évoquée dans la section 9.3 : "Variables externes et mise en scène". Le fonctionnement d'un algorithme de génération par règle dépend en grande partie du corpus de règles musicales que le musicien décide d'utiliser. On pourra, selon le style musical, utiliser des règles dont on sait qu'elles sont cohérentes pour écrire des mélodies et des motifs rythmiques. Nous ferons un exemple d'une constitution d'un corpus de règles musicales au cours du chapitre 11 : "Constitution d'un corpus de règles". Il existe plusieurs manières

d'envisager l'utilisation de tirages aléatoires ou externes pour composer une série de notes. On peut générer une séquence entière de notes et filtrer ou non selon des règles harmoniques et rythmiques celles que l'on désire garder. On peut aussi générer des écarts rythmiques et des intervalles successifs. La principale différence se fait dans l'ordre d'exécution que l'on choisit entre les tirages aléatoires et l'application des règles.

The diagram illustrates the generation of a musical motif in 4/4 time. It starts with a starting note (Do) and uses three stages of random intervals: +1, +4, +1, +3, -3, -1, -4. The resulting notes are: C4, D4, G4, A4, F4, E4, D4, C4.

- Etape 1: On choisit arbitrairement une note de départ (ici Do)
- Etape 2: On effectue 4 tirages entre 1 et 7, et on utilise les valeurs obtenues comme intervalles ascendants successifs (ici 1,4, 1 et 3)
- Etape 3: On prends les valeurs dans l'autre sens pour redescendre vers la note de départ

FIGURE 7.10 – un exemple de création d'un motif musical avec un algorithme simple de génération aléatoire.

Il est également possible de générer du signal audio d'une manière plus abstraite que celle que nous détaillons ici : on peut se baser sur la génération d'un signal dont les caractéristiques sonores sont variables (comme du bruit que l'on filtre, ou un synthétiseur dont on fait varier les paramètres). Il ne nous paraît cependant pas pertinent de la traiter ici, car il s'agit plus d'un travail de création sonore à part entière que de la mise en place d'un module servant à assister l'improvisation.

Chapitre 8

Interaction, adaptativité et communication

Nous séparons au cours de ce mémoire les concepts d'interaction et d'adaptativité. L'interaction est la manière dont le module réagit directement aux actions du musicien. L'adaptativité est la manière dont le module réagit au contexte, que ce contexte soit issu des actions du musicien (c'est donc une réaction indirecte) ou des actions du module. Elle n'est ainsi pas forcément liée aux actions du musicien (elle peut être due à un contexte que le module a créé lui-même). Cette distinction, qui vient de la recherche sur la musique adaptative dans le jeu vidéo [42], permet de rendre compte de la différence d'approche entre des réactions à des ordres directs et des réactions à des contextes. Il nous paraît cependant plus pertinent d'utiliser un seul terme facilement compréhensible pour rendre compte de ce que fait ce module : il interagit avec le musicien, il est donc interactif, même si son approche se base sur une combinaison d'adaptativité et d'interactivité.

Il ne s'agit pas de créer un module qui joue dans son coin selon ses propres règles, mais qui soit capable de réagir aux actions musicales du musicien et aux impératifs qu'il a posés en direct. Le module doit en pratique pouvoir réagir à deux types d'évènements.

- Il doit pouvoir recevoir des ordres directs du musiciens en temps réel. Il paraît important d'avoir un contrôle direct si le musicien a des intentions différentes de celles décidées par le module, et ce contrôle doit pouvoir servir de filet de sécurité durant la performance. C'est donc l'aspect interactif, qui se range dans l'idée de communication avec le musicien.
- Il doit pouvoir réagir à un contexte général établi par le musicien. Une partie de ce contexte est déterminé à l'avance (comme la signature rythmique) et le reste découle des choix du musicien au cours de la performance. Il doit enfin réagir au contexte qu'il crée lui même. S'il décide, par exemple, d'effectuer un "drop", il doit comprendre que le contexte musical a changé.

L'enjeu de la réaction du module au contexte musical, et donc de sa compréhension autonome du contexte, renvoie à ce que George Lewis appelle la "transduction émotionnelle" [43]. Il s'agit de la communication inhérente à l'interaction musicale qui se fait via le matériau musical lui même et non pas via des informations extérieures. S'il s'agit de rendre le module plus autonome, il s'agit aussi d'humaniser son fonctionnement et sa relation avec le musicien qui effectue la performance. L'intérêt est que cette interaction musicale s'effectue dans les deux sens, c'est-à-dire que le module comprenne les intentions du musicien intuitivement et que le musicien tire de ce qui est joué par le module des idées et des intentions. La musique exécutée par le module nourrit ainsi artistiquement la performance.

8.1 Communication avec le musicien

La communication avec le musicien s'effectue dans les deux sens. Le module doit recevoir des informations du musicien, mais aussi envoyer des informations au musicien (pour lui indiquer, par exemple, qu'il va effectuer un "drop").

8.1.1 Les informations entrantes

Les informations reçues du musicien peuvent être séparées en deux catégories. Il y a en premier lieu les indications préalables qui concernent le cadre général de la performance. Ces indications peuvent être de l'ordre du cadre musical, comme le tempo ou la signature rythmique, ou bien concerner simplement les réglages préalables du module. Si certaines de ces informations peuvent être amenées à changer au cours de la performance, le module nécessite d'avoir une valeur initiale pour commencer à fonctionner.

Il y a ensuite les ordres directs que le musicien donne au module. Ces ordres peuvent être reçus de manière indirecte (si le musicien change le tempo de la performance, il est facilement possible pour le module de récupérer seul cette information), ou envoyés par le musicien à l'aide d'un système de commande. Plus ces ordres sont abstraits, plus le module est autonome : il existe en effet tout un spectre d'abstraction entre des ordres basiques ("jouer telle note tout de suite"), et des ordres plus complexes ("introduire un changement"). L'enjeu est que le module comprenne le langage musical qu'un musicien pourrait utiliser pour communiquer avec un autre musicien. La mise en place d'un système de commande est très variable selon le musicien, étant donné qu'il suffit de mettre en place un système basé sur le MIDI pour que le musicien puisse choisir lui-même quel type de matériel il utilise pour envoyer ces commandes. Il peut s'agir de boutons sur des claviers maîtres et contrô-

leurs, d'interrupteur sur un pédalier MIDI, ou même de dispositifs plus spécifiques et originaux. Il existe en effet sur le marché une très grande variété de contrôleurs MIDI.

8.1.2 Les informations sortantes

Le module doit aussi faire part de ses intentions au musicien, et indiquer le contexte qu'il comprend afin de permettre au musicien de vérifier si ce contexte est correct. Il nous paraît en outre important que le module puisse faire des propositions au musicien en lui soumettant des changements que le musicien peut rapidement refuser si jamais ils rentrent en conflit avec ses intentions.

Ces informations peuvent être communiquées de manière visuelle, sur une fenêtre dédiée. Il se pose ainsi certaines questions :

- Comment communiquer le temps qui précède un changement, ou encore le temps que va durer un certain événement. La manière la plus simple serait de simplement énoncer le temps, mais il paraît pertinent d'afficher un décompte visuel. Les compteurs numériques sont très peu intuitifs, et une piste intéressante est celle d'un compteur sous la forme d'une horloge analogique, c'est-à-dire d'un mouvement circulaire, qui permet d'apprécier rapidement la temporalité du changement. Cette représentation sous forme d'horloge est présente dans le logiciel *Ableton Live* : un indicateur circulaire affiche sous chaque piste, dans la vue "session", la position temporelle dans un "clip" qui joue en boucle.



FIGURE 8.1 – Les deux systèmes d’affichage du décompte d’*Ableton Live*. Les deux clips sont de la même longueur mais le second joue en boucle et utilise donc un affichage avec horloge.

- Comment communiquer les informations rythmiques, dans le cas où le module produit un motif rythmique. Il semble que l’affichage d’une partition soit un choix peu pertinent, car les musiciens de *Techno* sont plus habitués à une représentation des motifs rythmiques sous forme de séquenceurs. Dans une partition classique, la position de chaque note dans une mesure est déterminée par les éléments précédents sur la partition (c’est-à-dire les silences et les autres notes). L’affichage des séquenceurs, lui, présente une grille, dans laquelle chaque colonne représente un temps donné, et les notes sont ainsi placées sur cette grille sous forme de trait plus ou moins long selon leur durée (et non pas de figures de notes)
- Comment communiquer les informations harmoniques. Il semble pertinent d’utiliser la notation américaine¹ des notes et des accords, car elle est la plus répandue dans la plupart des logiciels musicaux. Pour ce qui est des mélodies, il paraît complexe d’afficher visuellement la mélodie que le module produit. Si cela s’avère nécessaire, il semble qu’utiliser une notation de type "piano roll", c’est-à-dire d’une portée sur laquelle chaque demi-ton correspond à une ligne,

1. La notation française utilise le nom de la note, par exemple "Do, La, Ré" tandis que la notation américaine utilise des lettres par ordre alphabétique en partant du La, par exemple "C, A, D".

serait plus pertinent qu'une partition traditionnelle, étant donné que cette notation est plus répandue dans les logiciels musicaux utilisés en musique électronique.

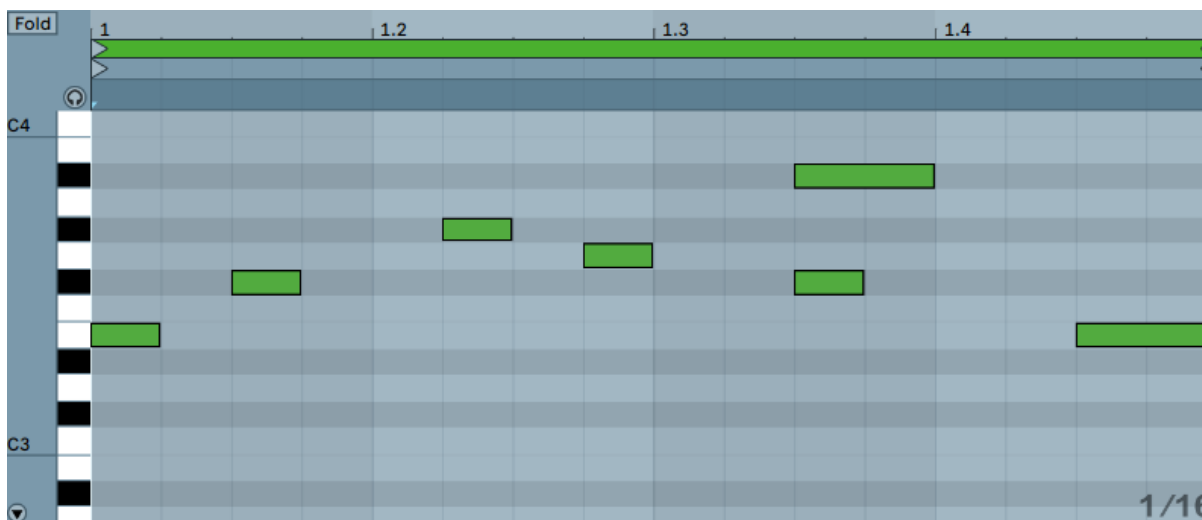


FIGURE 8.2 – Un affichage de partition midi de type "piano roll" issu d'*Ableton Live*, qui présente aussi une représentation rythmique typique des séquenceurs.

Un aspect intéressant de cette communication est qu'il est possible de la diffuser au public et de la mettre en scène. On peut ainsi imaginer, pour que le public puisse facilement comprendre, une communication abstraite et esthétisée des changements à l'oeuvre, qui puisse être diffusée visuellement pendant la performance. Les événements de musique *Techno* incorporent souvent des mises en scènes visuelles avec des projections d'éléments vidéos appelées "VJing", pour "Visual Jokeying", en référence au "DJing". On peut ainsi imaginer qu'un artiste *VJ* diffuse la communication entre le module et le musicien une fois mise en scène. Cette idée permettrait de palier à un problème souvent rencontré par les musiciens qui effectuent des performances : la difficulté de la compréhension des actions du musicien par le public. S'il est facile de comprendre instinctivement ce que font des musiciens qui jouent d'instruments électriques ou acoustiques, il est bien plus difficile de comprendre, pour un néophyte, les actions effectuées par des musiciens sur des ma-

chines et leur rapport avec ce qui est entendu. Cette difficulté se retrouve d'ailleurs dans des anecdotes de musiciens : Fasma, que nous avons interrogé, s'est vu, durant une performance, demandé de jouer un morceau en particulier². Ce genre de demandes rend compte du manque de distinction pour le public entre le travail du *DJ* et celui du musicien qui effectue une performance.

8.2 Compréhension du contexte

Le module doit pouvoir comprendre le contexte musical de la performance. Ce contexte musical peut être issu d'une analyse sonore ou musicale. Il existe ainsi de nombreuses études concernant les différents paramètres à analyser et les manières de les analyser [44]. Il y a en premier lieu le contexte lié à la narration, et donc à la structure générale de la performance, ainsi qu'aux mouvements qui la composent.

En ce qui concerne la structure générale, la compréhension paraît plutôt simple : dès le début de la performance, le module pose par défaut qu'il est dans une phase d'introduction, et s'il décide d'effectuer un changement (ou si le musicien le décide), son état change. Il paraît en outre prudent de réserver au musicien le soin de dire au module de rentrer dans une phase d'outro pour terminer la performance.

La narration momentanée représente un problème plus complexe. Même en se limitant aux trois parties que nous décrivons au chapitre "Constitution d'un corpus de règle", il ne paraît pas évident de trouver un moyen pour le module de comprendre seul dans quel partie il se situe sans disposer d'un contexte préalable, et surtout de comprendre dans combien de temps le musicien compte changer de partie. Cette

2. L'anecdote est d'autant plus étonnante qu'il jouait au niveau du sol, et que le public pouvait voir les différentes machines qu'il utilisait. L'installation du musicien au niveau du public est en outre souvent vue comme plus pertinente par les musiciens qui effectuent des performances : le fait d'être placé sur une estrade empêche le public de voir les machines qui sont le plus souvent posées sur une table.

problématique représente un enjeu majeur d'un système indépendant : il faudrait que le module puisse deviner les intentions d'un musicien comme un humain peut le faire intuitivement. Certains éléments que les musiciens utilisent entre eux pour communiquer sur la structure d'une performance tout en jouant sont difficilement utilisables pour communiquer avec le module :

- Les musiciens utilisent souvent des signes visuels pour communiquer entre eux. Mais les systèmes de reconnaissance visuelle supportent mal les changements fréquents d'arrière plan et nécessitent souvent une caméra de bonne qualité. L'introduction d'un tel système dans une performance devant un public ne serait pas assez fiable, complexe à mettre en place et fastidieux à calibrer.
- Les musiciens communiquent aussi parfois vocalement. Le contexte de la performance, avec son niveau sonore souvent élevé, et un besoin important d'alléger la puissance de calcul des ordinateurs utilisés rend peu envisageable l'intégration efficace d'un système de reconnaissance vocale.

Il faudrait donc que le module comprenne les intentions en fonction des éléments musicaux de la performance sans que le musicien lui indique. On peut ainsi imaginer plusieurs indices :

- Les variations de niveau sonore.
- Les variations dans l'équilibre fréquentiel : en effet, les musiciens de *Techno* utilisent souvent des filtres passe-haut juste avant d'effectuer un "drop", pour marquer une rupture dans l'équilibre fréquentiel de la performance. On peut donc imaginer que le module reconnaisse la baisse rapide des fréquences graves et médium et comprenne qu'il est censé effectuer un "drop". On peut aussi de manière plus simple mettre un déclencheur sur le filtre utilisé par le musicien.

— D'autres variations peuvent être imaginées par le musicien et servir d'indice, comme le fait d'envoyer de plus en plus la piste master dans un délai ou une réverbération, ou un motif musical particulier.

Il y a l'enjeu de la compréhension par le module des motifs rythmiques. Cette compréhension peut être utile dans le cas où le module doit placer temporellement des notes, ou choisir un motif musical. On peut imaginer plusieurs manières d'extraire ce contexte :

- Si les motifs qui posent le rythme sont joués en MIDI, le module peut simplement extraire le MIDI et regarder où sont placées les notes les plus accentuées.
- S'il s'agit d'audio, on peut assez facilement imaginer un dispositif de reconnaissance qui se base sur le niveau sonore ou la hauteur des notes pour comprendre où les accents rythmiques sont posés.

Une fois ce contexte extrait, le module peut décomposer les informations rythmiques selon deux axes : la position des temps forts et des temps faibles ainsi que la position temporelle des éléments rythmiques.

Il y a l'enjeu de la compréhension par le module de l'harmonie et de la tonalité. Cette compréhension peut être utile dans le cas où le module doit générer des motifs mélodiques, des accords, ou bien choisir des motifs musicaux avec des hauteurs précises et des mélodies. Cette compréhension se décompose en plusieurs parties. Il y a d'abord la reconnaissance des notes. C'est l'étape la plus simple si le module extrait les informations de signaux MIDI ou à partir de métadonnées³. Si le module extrait ces informations de signaux audio, il existe différents systèmes pour extraire la hauteur d'une note seule, ou bien celle de plusieurs notes jouées simultanément.

3. Nous utilisons ici le terme de métadonnées pour désigner les données entourant un clip audio ou MIDI, comme la tonalité ou le rythme. Il s'agit donc de parties issues d'une nomenclature, que nous avons évoqué dans la section 7.1.

L'extraction en temps réel des notes d'un accord sous forme de signal audio est en revanche peu fiable. Une fois ces notes obtenues, il faut que le module comprenne la tonalité du morceau en cours, ou bien les fonctions respectives des accords et des notes jouées. Cette étape nécessite de faire des choix concernant l'approche musicale de la performance (tonale ou modale). Il existe ensuite plusieurs moyens d'extraire le contexte musical :

- On peut extraire le contexte musical en comparant les notes jouées à des gammes connues. C'est le procédé le plus simple et le plus répandu [45]. Ce procédé permet parfois d'obtenir plusieurs gammes possibles. Le module peut ainsi se servir de cette information pour proposer une tonalité.
- On peut tenter de reconnaître la grille d'accord, et ensuite de reconnaître une cadence. Le fait de se baser ensuite sur la grille d'accord extraite du contexte musical pour créer de la matière musicale représente l'approche majoritaire dans les systèmes d'accompagnements à l'improvisation orientés vers le Jazz [46].

Chapitre 9

Les choix du module

Le module doit effectuer un certain nombre de choix durant sa performance. Ces choix peuvent concerner des aspects généraux, comme des changements de partie. Ils peuvent aussi concerner des décisions plus restreintes, comme le fait de proposer une grille d'accord, ou d'ajouter des ornements à une section rythmique. Ils peuvent enfin être très spécifiques et concerner par exemple le choix de la hauteur d'une note dans une mélodie.

9.1 Les systèmes de choix

Il existe plusieurs systèmes pour permettre au module d'effectuer des choix. Si leur fonctionnement revient la plupart du temps au même (le module considère des règles et des conditions), ils peuvent se concevoir de plusieurs manières en ce qui concerne la façon de mettre en place ces règles et ces conditions. On peut aussi ajouter à ce système de règles et de conditions des tirages aléatoires pour prendre des décisions arbitraires.

9.1.1 Le hasard

Si la logique de remettre le choix de certaines règles entre les mains du hasard n'est pas à proprement parler algorithmique, son utilisation est largement répandue dans la plupart des champs d'utilisation de l'algorithme en musique que nous avons vus précédemment. De nombreux choix peuvent être déterminés de manière aléatoire en effectuant un tirage et en interprétant le résultat.

L'idée qui revient souvent, quand le hasard est utilisé, est d'utiliser des probabilités ou des pondérations pour avoir un certain contrôle sur le hasard. On peut ainsi imaginer des événements avec une probabilité d'apparition réglable. Un outil statistique utile pour pondérer des tirages aléatoires est la loi de Gauss. Cette loi modélise la distributions de différents tirages sur une courbe. La courbe est centrée autour d'une valeur, et la probabilité d'obtenir une valeur décroît au fur et à mesure que cette valeur s'éloigne de la valeur centrale. Le fait d'utiliser des lois de densités statistiques permet de laisser la possibilité de valeurs et de choix musicaux peu conventionnels, tout en minimisant leur probabilité d'apparition.

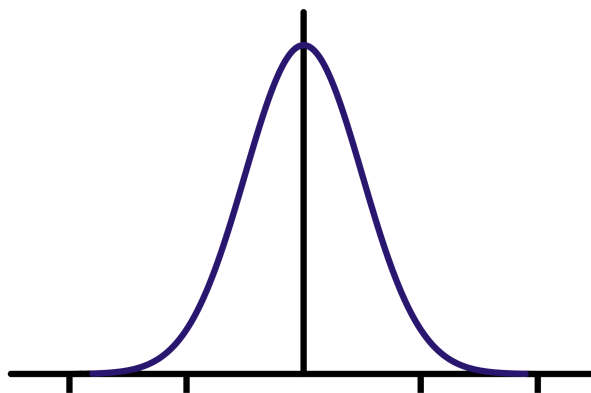


FIGURE 9.1 – Schéma d'une courbe de distribution statistique suivant la loi de Gauss.

Une autre manière d'envisager les tirages successifs (par exemple dans le cas d'une suite de notes) et leur modélisation est l'utilisation de chaînes de Markov. Il s'agit d'un modèle dans lequel les probabilités d'un tirage dépendent des résultats d'un tirage précédent. Par exemple, on peut établir que si l'on tire deux notes à la suite, la probabilité que la seconde note soit un La est de 1/100 si la note précédente était un La, et de 1/12 sinon. Leur utilisation permet donc d'éviter certaines répétitions.

Il se pose aussi la question du moment où l'on effectue les tirages aléatoires. On peut ainsi décider que le module effectue au démarrage un tirage général qui contiendra toutes les valeurs qu'il va utiliser, ou bien effectuer un tirage à chaque fois qu'il a besoin d'une valeur. L'intérêt d'un tirage général est de permettre d'en garder les valeurs pour les réutiliser durant une autre performance : il est possible qu'un tirage particulier produise des résultats intéressants dans le comportement du module. Il paraît aussi important de permettre au musicien de relancer les tirages si besoin, et donc d'afficher les conséquences de ces tirages à l'avance : si on tire au hasard le temps que va attendre le module avant de rajouter un élément musical au début de la performance, il faut que le musicien puisse modifier rapidement ce temps s'il lui paraît trop court.

9.1.2 Le système par liste

La manière la plus simple d'ordonner la prise de décision d'un algorithme est le système par liste, dans lequel les règles sont lues l'une après l'autre, comme dans le cas du système *Voyager*. Ce système est simple à mettre en place et à présenter à l'utilisateur, et présente pour avantage d'être très intuitif d'utilisation, puisque le déroulement de la lecture des règles est linéaire, et s'effectue par ordre de priorité. On peut de plus obtenir facilement des changements de fonctionnement avec l'ajout

d'une règle ou l'inversion de positions de plusieurs règles. Cette liste peut être vue comme une chaîne de fonctions "if", très utilisées en informatique. Le fonctionnement est simple : si une condition est remplie, alors le module effectue une action ou note une information. Il présente néanmoins des limites, en terme de présentation et d'organisation, dès lors que certaines règles sont spécifiques à des contextes particuliers. Si une fonction "if" appelle elle-même d'autres fonctions "if", par exemple, la représentation linéaire de la chaîne devient rapidement confuse.

9.1.3 Les arbres décisionnels

Le cas précédemment évoqué de fonctions "if" appelant elles-mêmes d'autres fonctions "if" peut se présenter sous la forme d'un arbre décisionnel. Un arbre décisionnel consiste en une série de choix sous formes de branches. On fait un choix, et l'on fait ensuite un autre choix, ainsi de suite. Par exemple : décider que, si le morceau est dans une phase de montée, on pourrait intensifier la mélodie pour aider à la montée, peut être vu comme un embranchement dans un arbre décisionnel. Ces embranchements peuvent être des fonctions "if", mais aussi des choix à effectuer de manière aléatoire¹. Les arbres décisionnels peuvent comporter des "retours", c'est à dire des embranchements qui conduisent à remonter au besoin à un embranchement situé plus haut dans l'arbre.

1. Un choix à effectuer de manière aléatoire revient d'ailleurs à utiliser une fonction "if" selon le résultat d'un tirage.

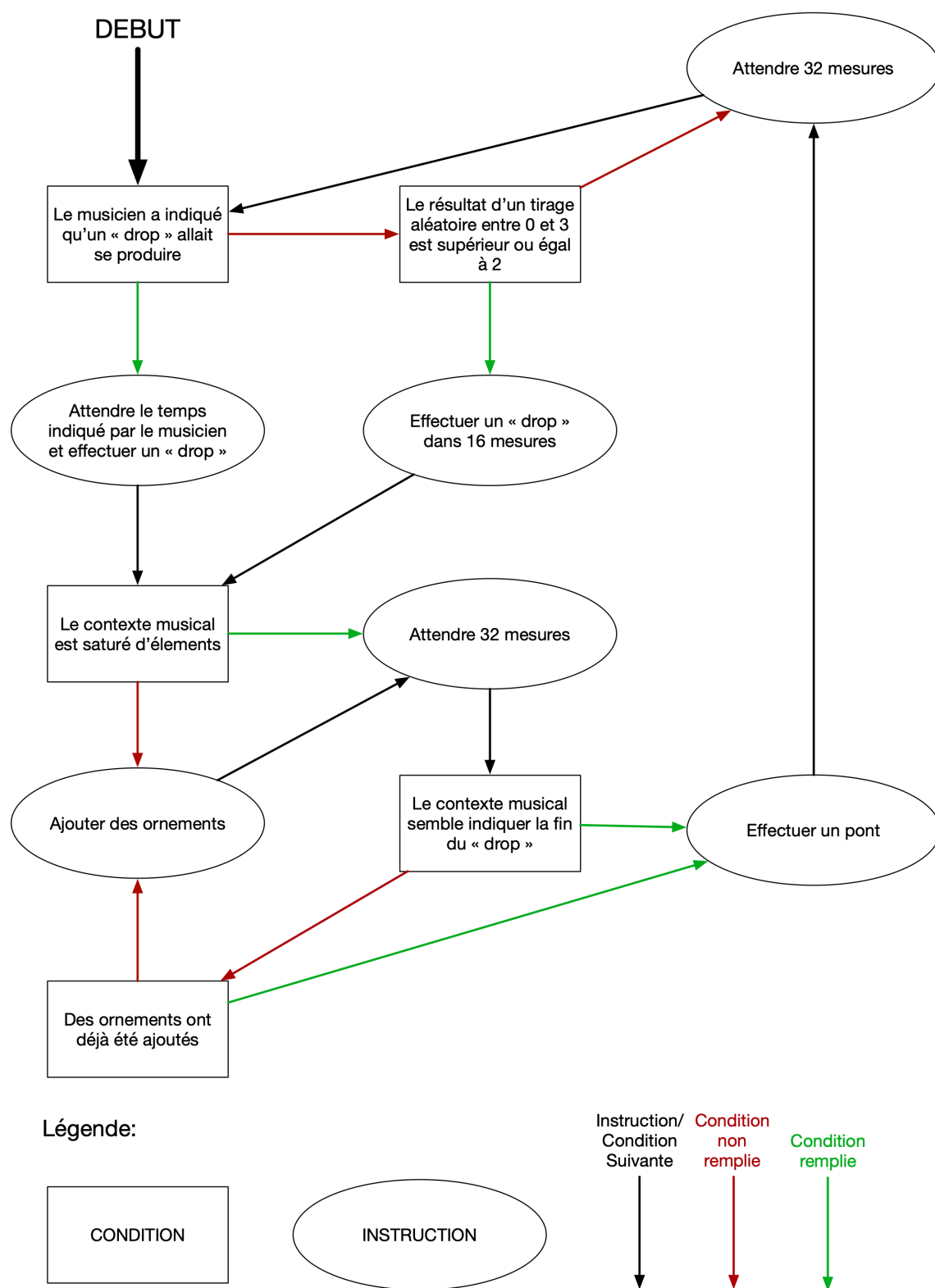


FIGURE 9.2 – Schéma d'un arbre décisionnel : exemple d'un module qui jouerait un "kick" durant un "drop" dans une performance de *Techno*.

9.2 Intentions et Contexte

Le contexte est une information que l'on obtient directement ou indirectement à partir des autres informations dont on dispose sur la performance. De ce contexte dérivent des intentions, c'est-à-dire des choix qui sont cohérents dans le contexte et témoignent d'un changement ou d'une continuité dans ce contexte. L'intention peut ainsi être obtenue par la compréhension du contexte : par exemple, si l'on constate que le niveau sonore diminue (contexte) on peut en déduire que le musicien cherche à faire monter une tension (intention), ou à calmer le morceau (intention). L'intention est au cœur de la notion d'autonomie, car elle revient à prendre des décisions plus ou moins arbitraires. L'impact des intentions sur la performance est aussi plus large et moins prévisible. Un module qui a des intentions se démarque d'un simple module qui exécute des ordres contextuels. Sa position dans la performance est moins dans un rapport d'"esclave" vis-à-vis du musicien, et plus dans un rapport d'égalité. On peut ainsi utiliser le terme de "co-construction" de la performance, car le module nourrit les actions du musicien autant que les actions du musicien nourrissent celle du module.

9.2.1 Systèmes rule-based et case-based

Il existe une distinction entre les différents systèmes d'accompagnement à l'improvisation qui permet d'établir cette idée d'intention plus en détails. Dans un travail de recherche consistant à s'inspirer d'*ImPact* pour réaliser un système qui pourrait improviser en temps réel des grilles de guitares sur de la Bossa Nova nommé Cyber-João, Marcio Dahia et al. notent qu'on peut distinguer *Impact* et *Band in a box* en introduisant la notion de système "case-based" et "rule-based" dans la manière dont le système pioche dans sa banque de données pour choisir ce qu'il faut jouer [47]. La

distinction se fait ainsi :

- Un système "rule-based" est un système dans lequel des règles interdisent l'utilisation de certains motifs selon le contexte. *Band in a box* est ainsi un système exclusivement "rule-based". La grille d'accord, ainsi que certaines règles narratives, servent de base pour exclure des motifs selon l'accord en cours, et le système pioche dans les motifs disponibles au hasard. Ce hasard peut en outre être pondéré.
- Un système "case-based" est un système dans lequel le contexte permet de choisir une intention de jeu, ce qui va donner des indications pour choisir dans une banque de données un motif à jouer en fonction d'attributs. La nuance est ainsi que le choix ne se base pas sur des pondérations et des interdictions (comme dans le cas d'un système "rule-based") mais sur des intentions. *Impact* est ainsi un système "case-based" et "rule-based" : le contexte impose des restrictions (jouer dans telle tonalité), et permet de décider d'une intention (nous sortons de l'introduction du morceau, qui était très dense musicalement, donc nous allons choisir un motif avec une densité musicale faible) qu'il va traduire en effectuant un choix dont le caractère aléatoire est pondéré.

Cette distinction montre deux manières différentes de choisir un motif musical dans une banque de données, dont la seconde utilise le concept d'intention. Marcio D'Almeida et al. montrent en outre qu'un système qui combine "case-based" et "rule-based" tend à produire des résultats plus convaincants musicalement comparé à un système purement aléatoire, et à un système simplement "rule-based". Ces notions se superposent donc aux différents systèmes de choix évoqués précédemment. L'action inscrite dans un contexte, tout comme la décision émanant d'une intention, peuvent se faire en utilisant des listes ou des arbres de décision tout en gardant une part variable

de hasard.

9.3 Variables externes et mise en scène

Une possibilité intéressante que nous n'allons néanmoins pas traiter dans le cadre de la réalisation technique de ce mémoire est celle de la remise de la prise de décision au public. Une idée qui nous est venue est de reprendre un fonctionnement analogue à celui de certains jeux vidéos vis-à-vis du système de diffusion en direct *Twitch*. Le principe de *Twitch* est de permettre à des joueurs de diffuser en direct leur session de jeu, et à un public d'assister à la performance. Le public peut interagir avec le "streamer"² via un système de commentaires et de dons. Certains jeux ont développé une manière supplémentaire pour le public d'interagir : il lui est possible de voter pour que des événements se produisent dans le jeu. Le jeu *Dead Cells*, développé par le studio français Motion Twin et publié en 2018, permet par exemple au public de voter pour choisir le niveau dans lequel le joueur va entrer. On peut ainsi imaginer des systèmes qui permettent au public d'influencer les choix du module, par un système de vote par exemple. Ce système de vote pourrait se présenter sous la forme d'un site internet accessible sur un téléphone mobile ou d'une application. Les variables utilisées par le module pour prendre des décisions pourraient aussi venir de paramètres externes au dispositif de la performance, comme la température, l'heure de la performance, la quantité de public de la salle... Si ces idées s'éloignent du cahier des charges que nous avons posé pour le module, elles nous semblent intéressantes à évoquer car leur intégration peut se faire simplement en remplaçant le processus de décision autonome du module par un autre processus faisant appel à des variables extérieures. Elles accomplissent d'ailleurs en partie le même objectif que le module :

2. Le joueur qui diffuse sa session de jeu.

elles permettent d'ajouter de l'imprévisibilité et de réduire la quantité de choix et d'actions que le musicien doit effectuer.

9.4 L'équilibre des choix

Il faut que le module joue "avec" le musicien, et non pas "à côté". C'est pour cela qu'il est important de trouver un équilibre entre des choix arbitraires et aléatoires, des choix contextuels, et des choix qui résultent de la concertation avec le musicien. Il apparaît que certaines décisions peuvent être abordées de plusieurs manières par le module et qu'un enjeu important est de permettre à ces manières de coexister. Si l'on prend par exemple la décision, pour le module, du temps à attendre avant d'effectuer un drop, le module peut tirer ce temps au hasard, décider en fonction de ce que joue le musicien, ou simplement attendre les ordres du musicien. Il faut donc établir un ordre de priorités entre ces manières. Les ordres du musicien doivent, sauf cas contraire résultant d'une volonté artistique, toujours l'emporter sur les choix du module. On part en effet du principe que le musicien aura toujours une meilleure compréhension du contexte et des enjeux de la performance. Le cas de la priorité entre des choix contextuels et pris au hasard est plus compliqué. Une des idées qui nous semble la plus simple à mettre en œuvre est de rajouter des choix aléatoires dans les embranchements des arbres décisionnels du module. On peut aussi introduire ces choix aléatoires à plus grande échelle et en faire des événements qui conduisent le module à aller directement chercher une des voies dans son arbre décisionnel.

Chapitre 10

Modèle général d'un module d'improvisation algorithmique interactif

Le fonctionnement d'un algorithme musical en temps réel peut être décomposé en quatre étages, qui sont, dans l'ordre du fonctionnement, l'écoute, l'intention, la traduction, et la génération. L'intérêt de découper ce fonctionnement est qu'on peut ensuite concevoir des "sous-modules", chacun réalisant l'une des 4 étapes, et les réutiliser dans d'autres modules. A partir de l'étape d'intention, chacune des étapes est un goulot d'étranglement en terme de possibilités. La marge de manœuvre est ainsi directement limitée par la complexité de l'étage qui suit.

10.1 Ecoute

L'étage d'entrée correspond à l'étage qui reçoit, analyse et stocke les informations entrantes que nous avons décrit au chapitre 8. Il peut accepter, de manière générale, plusieurs entrées :

- Des informations issues d'un séquenceur.
- Des informations musicales pures (la tonalité, la signature du morceau en cours)
- Du signal audio ou MIDI.

Les instructions spécifiques venant du musicien ne sont pas prises en compte dans l'étage d'écoute car il s'agit d'ordres directs pour les étages suivants. C'est l'étage dans lequel se situe le stockage interne, qui communique le contexte à tout les étages ultérieurs. Cet étage se décompose en 4 parties.

- Une partie "informations préalables" qui regroupe les informations obtenues en amont de la performance.
- Une partie "base de donnée", qui regroupe les différentes bases de données auquel le module fait appel pour effectuer ses choix ou son analyse, ainsi que le répertoire de motifs que le module peut jouer.
- Une partie "contexte précédent", qui constitue la mémoire du module et contient toutes les informations qui ont été changées. Cela permet au module de déduire des informations des changements, et cela lui permet de savoir ce qui a déjà été fait. Il paraît important, par exemple, si plusieurs parties de la performances ont été jouées dans une tonalité précise, que le module essaye d'éviter cette tonalité, ou au contraire s'en serve comme pour effectuer un rappel.
- Une partie "contexte actuel" qui regroupe les informations sur le contexte actuel issues des actions du module et du musicien.

C'est aussi l'étage dans lequel s'effectue l'analyse des informations sur le contexte.

Plus le module est capable d'extraire seul des informations, plus il sera autonome, et donc la complexité de cet étage dépendra du niveau d'indépendance visé. Il doit y avoir un système de feedback : le stade de l'écoute permet au module de se construire un ensemble d'informations sur le contexte, et il est logique que si le module lui-même modifie ce contexte, il en ait connaissance. La complexité de ce module comporte deux enjeux. D'une part, plus il peut comprendre des choses de manière autonome, moins le musicien a besoin de lui en communiquer pour qu'il fonctionne. D'autre part, contrairement aux étages suivants qui sont limités par les possibilités de l'étage auquel ils communiquent des ordres, tous les étages sont plus ou moins limités par le niveau de compréhension du contexte dont le module dispose.

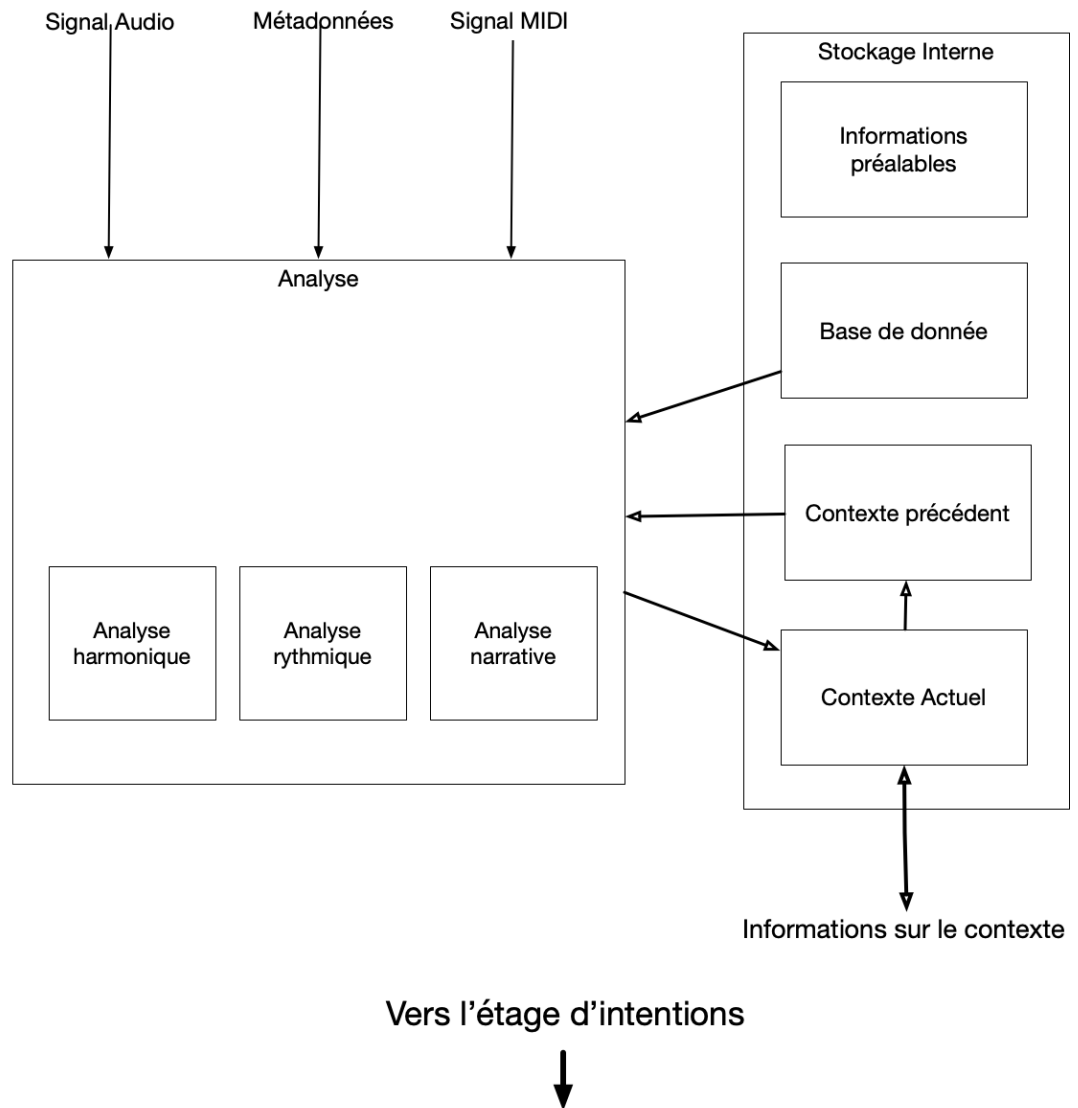


FIGURE 10.1 – Schéma de l'étage d'écoute.

10.2 Intention

C'est l'étage, en quelque sorte, de la prise de décision. Cet étage peut être transparent (on arrive sur un drop -> on traduit le drop en langage musical puis on génère le drop) mais peut aussi être un étage où des décisions sont prises, de manière aléatoire ou non (on arrive sur un drop -> je choisis de jouer de la basse dans une tonalité différente -> je traduis cette intention en partition de basse -> je génère cette basse) Cet étage comporte deux sous-systèmes :

- Un système de tirage aléatoire. Si les tirages se font en temps réel, ils sont sollicités par le système de décision en temps réel et communiquent les valeurs des tirage en temps réel. Ces valeurs peuvent aussi être communiquées à l'étage de traduction si nécessaire. Dans le cas où toutes les valeurs sont tirées au préalable, il est utile de faire remonter l'ensemble des valeurs dans le stockage interne du module, afin de centraliser les informations générales dont le module dispose. En pratique, dans le cas où les tirages sont effectués à chaque fois que nécessaire, il n'est pas obligatoire de mettre le système de tirage à cet endroit. On peut avoir plusieurs systèmes de tirages disposés à chaque endroit où l'on en a besoin.
- Un système de décision qui va donc décider d'intentions de jeu à appliquer pour les envoyer vers l'étage de traduction afin qu'elles soient mises en place, et les faire remonter vers l'étage d'écoute afin qu'elles soient prises en compte dans le contexte et qu'elles permettent la compréhension du contexte. Ce système se divise en deux parties : une partie autonome, qui décide d'intentions sans considérer l'avis du musicien, et une partie interactive, qui propose des événements et des intentions au musicien et doit donc recevoir ses réponses si besoin.

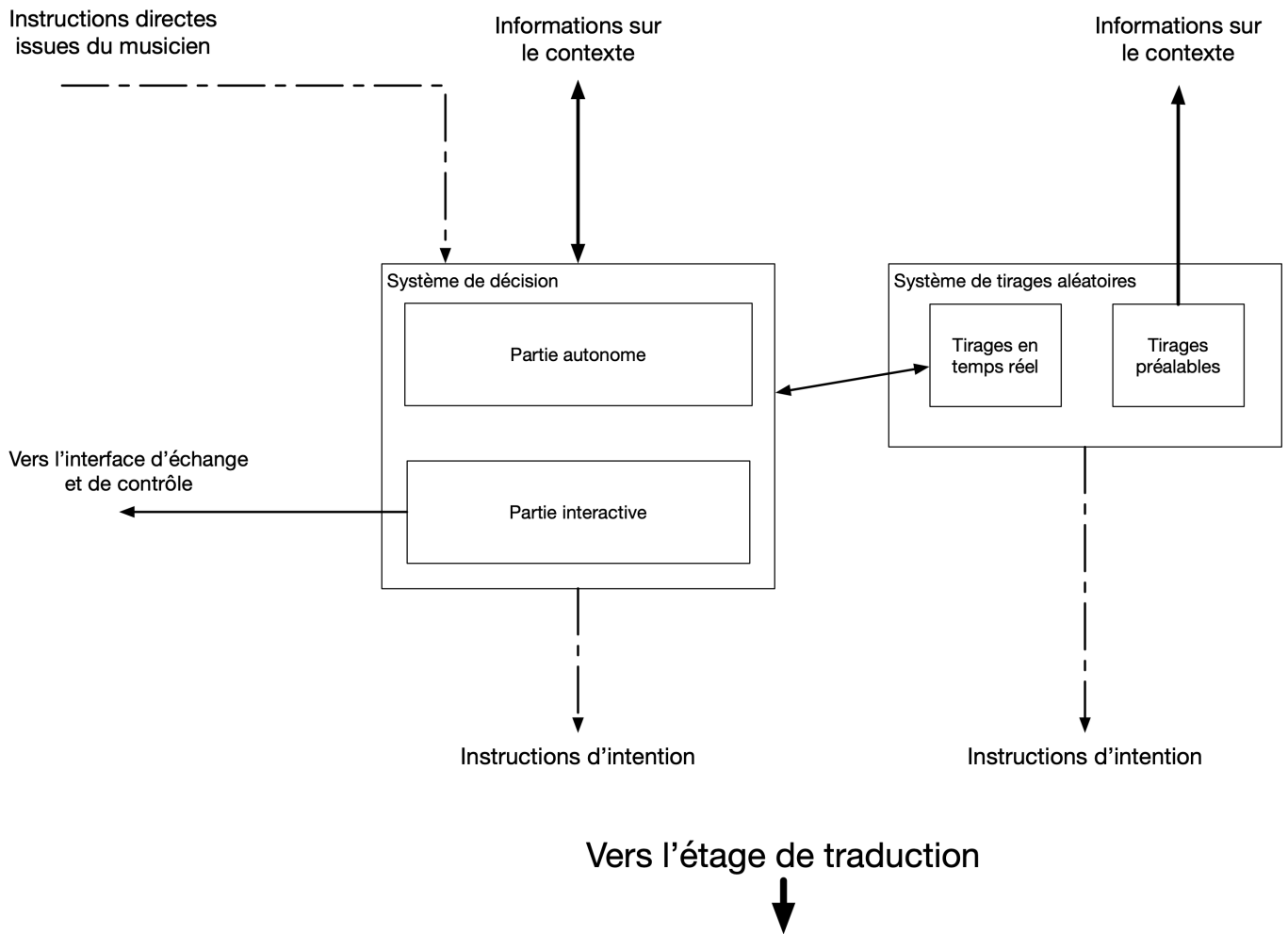


FIGURE 10.2 – Schéma de l'étage d'intention.

10.3 Traduction

L'étage de traduction correspond à la mise en forme des instructions découlant des intentions. Par exemple, si l'intention est de changer de tonalité, l'étage de traduction s'occupe de choisir la tonalité d'arrivée, et de mettre en place les modalités de ce changement. Cette étape reçoit de l'étage d'intention plusieurs types d'instructions.

- Des instructions temporelles complexes ("Faire un break à la fin du cycle") qu'il

traduit en instructions temporelles simples pour les envoyer à l'étage de génération. Le système qui permet cette traduction comporte un système de compteur esclave du tempo de la performance afin d'être synchronisé à cette dernière.

- Des instructions musicales pour le système de génération. Ce système génère des instructions de jeu précises pour qu'un signal audio soit généré. Il peut aussi recevoir des motifs musicaux issus de la performance si le module permet leur utilisation pour générer des motifs nouveaux. C'est ici que rentrent en compte les méthodes que nous avons détaillées au chapitre 7.
- Des instructions vis-à-vis du système d'accès au matériau musical, s'il en possède un. Ce système fait appel à la banque de données que nous avons rangée dans l'étage d'écoute. Il transmet les instructions vis-à-vis du jeu des motifs à l'étage de génération. Il transmet aussi, si besoin, des motifs au système de génération (dans le cas où celui-ci utilise des méthodes de transformations ou de combinaisons de motifs pré-existants). C'est ici que rentre en jeu la nomenclature dont nous avons parlé dans le chapitre 7.

La complexité de cet étage influe directement sur le niveau d'abstraction des ordres que le module peut recevoir et se donner. L'étage de traduction comporte aussi le système qui permet de prévenir le musicien des actions en cours, étant donné que c'est dans cet étage que les détails sont fixés.

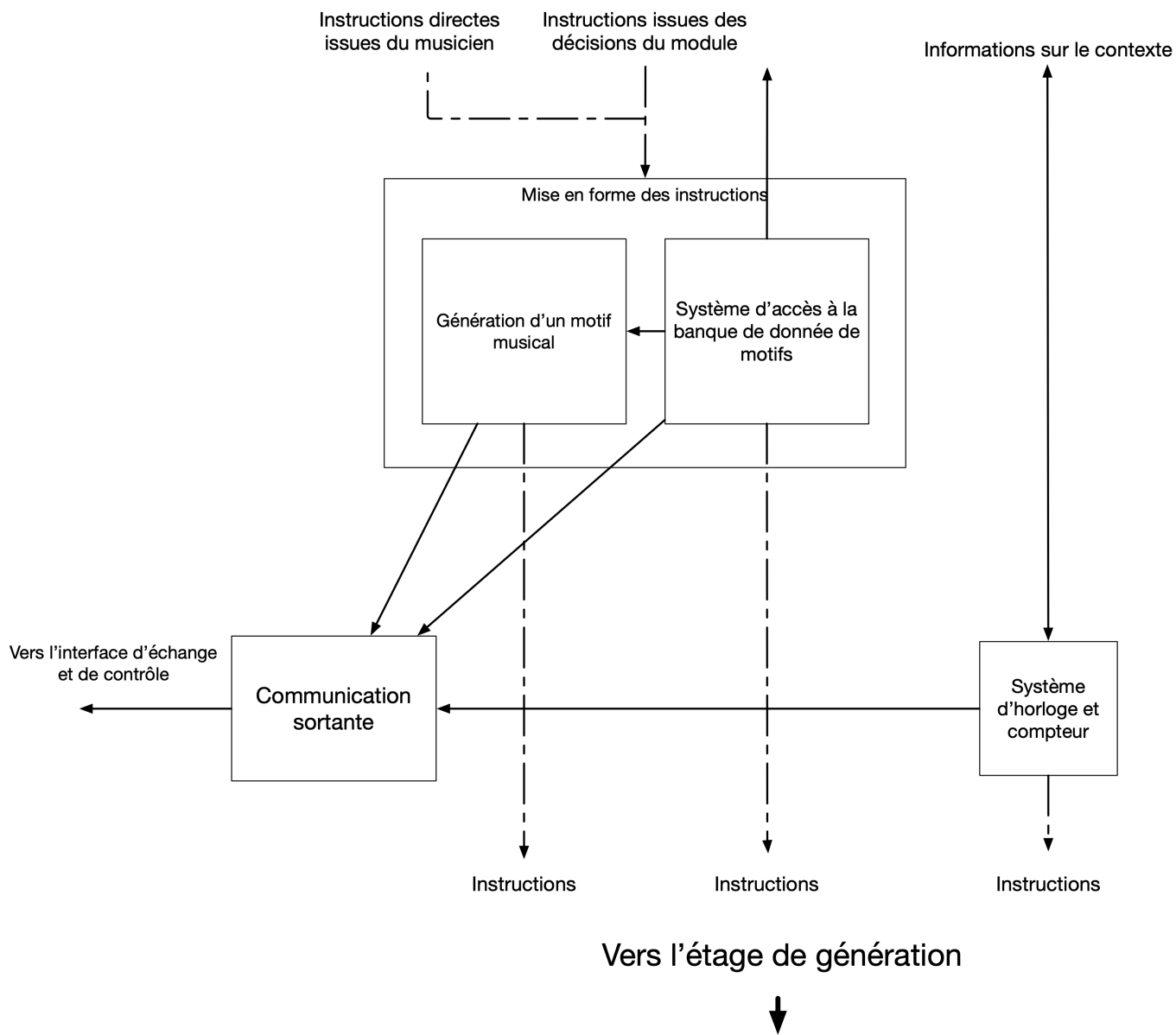


FIGURE 10.3 – Schéma de l'étage de traduction.

10.4 Génération

Cet étage concerne la génération pure du matériau musical, qu'il s'agisse de la génération d'un signal audio avec des synthétiseurs, de la génération d'un signal midi, ou de la générations d'instructions de contrôle d'un autre type (comme l'OSC¹, par exemple). Il s'agit en quelque sorte de l'étage "ouvrier" qui se contente de mettre en œuvre les ordres issus de l'étage de traduction. Le fonctionnement de cet étage diffère fortement selon la méthode de génération du signal audio. Il se rapproche dans certains cas de celui d'un instrument logiciel ou d'un séquenceur. De manière analogue aux étages précédents, sa complexité et les possibilités qu'il offre conditionnent la marge de manoeuvre du module.

1. L'OSC, ou Open Sound Control, est un protocole de communication entre dispositifs multimédias.

Chapitre 11

Constitution d'un corpus de règles

11.1 Définition des différentes règles

Nous classerons les règles évoquées dans ce mémoire en quatre catégories : les règles narratives, les règles rythmiques, les règles tonales, harmoniques et mélodiques, et enfin, les règles sonores. Si nous séparons la constitution d'un corpus de règles de l'étape de théorisation du module, c'est parce que les règles sont dépendantes du style et de la visée de la performance. Les règles que nous allons édicter ci après renvoient aux usages de la musique *Techno*, et fonctionnent plus largement dans le cadre de l'*EDM*.

Les règles narratives renvoient à l'organisation générale d'un morceau et aux mouvements qui le composent. Par exemple, si les annotations d'intensité de jeu sur les partitions constituent des règles sonores, elles font partie d'un ensemble qui tend à exprimer des mouvements, comme une montée ou une baisse en intensité sonore. L'organisation de la plupart des morceaux de musique actuelle suit des règles narratives comme le "couplet refrain", très présent en variété, et le "montée drop" que l'on retrouve dans la plupart des genres de l'*EDM*. Ces règles peuvent conditionner d'autres règles : ainsi comme vu précédemment, dans la sonate, une règle tonale est

liée à une règle narrative.

Les règles rythmiques désignent les règles relatives à l'organisation temporelle des sons. Cela recouvre les règles générales du morceau (le tempo ou la signature rythmique), les motifs rythmiques utilisés dans le morceau (que l'on appelle ici "groove"), ou enfin le placement des sons en fonction de leur fonction rythmique, que l'on définira en relation au style de musique.

Il est logique de classer ensemble les règles renvoyant à la tonalité, à l'harmonie et la mélodie, car les trois renvoient finalement aux notes jouées. On distinguera cependant les règles tonales, qui concernent la tonalité de la musique jouée, les règles harmoniques, qui concernent les accords et leurs relations, et enfin les règles mélodiques qui concernent les gammes et les modes utilisées par les mélodies, ainsi que les mouvements verticaux qu'elle effectuent.

Les règles sonores renvoient à la nature des sons joués, donc à leur intensité, leur timbre et leur enveloppe. Ces règles permettent aussi de classer certains sons et de leur attribuer une nature (un son qui aura des caractéristiques sonores se rapprochant du "hi-hat" pourra être utilisé comme un "hi-hat" dans la nomenclature rythmique).

11.2 Les corpus de règles

Si aucun des règles et des usages que nous allons décrire ne sont absolus, ils permettent de réaliser un module qui puissent facilement interagir avec un musicien en tenant compte des conventions du genre *Techno*. Il convient donc, avant de poser des méthodes de compositions et un éventail de règles arbitraires, de revenir sur ces règles musicales.

La démarche de constitution d'un corpus est elle-même source d'intérêt artistique. Elle permet, d'une part, de réfléchir autour des codes d'usage d'un genre musical. Cet aspect ethno-musicologique est riche car il permet souvent de confronter les usages et leur origines au contexte moderne de la performance. Les intros des morceaux de musique *Techno* sont souvent longues pour faciliter leur usage par un *DJ*, et cette convention se transpose souvent aux performances sans que le contexte ne l'impose réellement. Tout genre musical comporte aussi des conventions qui découlent de considérations organologiques. La *Techno* n'en est pas exempte et son esthétique résulte en partie de son contexte technologique et matériel de création.

La constitution d'un corpus amène aussi le musicien à réfléchir sur ses propres codes et son langage musical personnel. De nombreuses méthodes de composition ne sont jamais formalisées par les musiciens qui les utilisent, et la démarche d'avoir à les expliquer en détail pour en faire des algorithmes permet parfois de réfléchir sur leur utilisation et de les remettre en question.

11.2.1 Le corpus de règles rythmiques

Un critère important, de par la visée de la musique à être dansée, est celui de la signature rythmique "4/4", c'est-à-dire de mesures de 4 temps à la noire. Si certaines exceptions existent, cette signature est quasi tout le temps présente pour permettre

la danse. Il apparaît aussi rapidement compliqué pour un *DJ* de faire une transition cohérente entre deux morceaux de signatures rythmiques différentes.

On note aussi que le tempo est souvent compris entre 110 et 150 battements par minute (ou BPM), ce qui varie selon les différents sous-genres. Il est courant de distinguer les genres selon leurs tempo : la *Techno* "traditionnelle" sera située entre 120 et 135, la *Hard Techno* entre 135 et 150 BPM par exemple. Si le tempo n'est pas une référence absolue pour classer les différents genres, il représente un critère important pour organiser les ordres de passages des différents artistes en soirée *Techno*. Les artistes qui joueront d'un genre avec un tempo rapide, comme la *Techno* ou la *Hard Techno*, seront souvent placés après ceux qui jouent d'un genre avec un tempo plus lent, comme la *Deep House*, la *House* ou la *Tech-House*.

Ce tempo varie souvent pendant les *DJ set* et les performances, souvent d'un tempo modéré vers un tempo rapide, mais pas toujours dans le cas du *DJ set* où l'on peut avoir des mouvements légers entre les morceaux qui cherchent moins à modifier la vitesse générale du "set" qu'à permettre une transition entre deux morceaux de tempo différent¹.

La *Techno*, comme un grand nombre de musiques actuelles, utilise un système rythmique basé sur des temps forts et des temps faibles. Les temps forts correspondent au premier et au troisième temps, tandis que les temps faibles correspondent au second et au quatrième temps. Les termes "fort" et "faible" renvoient à l'origine à l'intensité du jeu des notes placées sur ces temps. Les temps forts et faibles sont traditionnellement, dans un grand nombre de genres de musique actuelle, associés respectivement au pied de grosse caisse et à la caisse claire. La spécificité de la *Techno*

1. Les *DJs* peuvent ajuster le tempo sur leurs platines, mais des modifications trop élevées du tempo ont des répercussions importantes sur les caractéristiques sonores du morceau joué. Ils cherchent ainsi un équilibre entre les tempos de deux morceaux successifs.

est qu'elle utilise un style "four on the floor", c'est-à-dire que chaque temps est marqué d'une grosse caisse et donc que chaque temps peut être ressenti comme un temps fort. Les seconds et quatrièmes temps sont des temps à la fois "faibles" et "forts" étant donné qu'ils sont aussi marqués, le plus souvent, par un son faisant office de caisse claire.

En *Techno* et en *House*, on distingue 4 fonctions rythmiques principales.

- Le "kick", qui correspond à un pied de grosse caisse. Il est souvent généré par un synthétiseur. Au-delà du lien avec la grosse caisse des musiques actuelles, peut être vu comme un "kick" tout son avec une attaque forte et énormément de fréquences graves qui permet de marquer des temps forts.
- La "snare", qui sert à marquer les temps faibles. Une "snare" est à l'origine un coup de caisse claire, mais renvoie plus largement à un son avec une attaque très forte et chargé de fréquences médiums. Cette "snare" peut être échantillonnée ou bien générée par un synthétiseur. On peut aussi la remplacer par un "clap" (un empilement de claquements de mains), un "snap" (un empilement de claquements de doigts) ou une "rimshot" (un son obtenu en frappant le cadre d'une caisse claire).
- Les "hi-hats", qui servent à marquer les contretemps. Il s'agit de sons aigus avec une attaque plus ou moins forte, et peuvent correspondre en musique actuelle à des coups de charleston. On distingue les "hi-hats" ouverts, plus longs, qui servent souvent à marquer les contretemps, et fermés, qui jouent autour du contretemps en double croche.
- Les ornements qui peuvent être de nature diverse, soit d'inspiration instrumentale (des bongos, des congas, des toms...), soit plus abstraits et qui créent des motifs rythmiques plus libres dans leur composition et sont superposés au

reste.

Nous pouvons établir un code concernant le "kick", pour le module qui sera en charge de la partie rythmique : il faut que chaque temps soit marqué d'un "kick", avec éventuellement des ornements, c'est-à-dire des "kick" moins marqués.

Il existe cependant d'autres manières de disposer les "kick" et de créer la base rythmique, comme le *Two-Steps*, si présent dans le *UK Garage* qu'il en constitue presque un impératif musical.

Il faudra qu'un critère important de la construction de la partie rythmique soit l'introduction d'un Hi-Hat sur les croches afin de marquer les contretemps. On notera qu'il est courant de passer d'un "Hi-Hat" fermé à un "Hi-Hat" ouvert sur le contretemps pour intensifier la partie rythmique.

Le reste de la section rythmique peut être étoffé avec des ornements de deux manières :

- soit purement aléatoire, en générant un motif au hasard, une technique déjà largement incluse dans la plupart des boîtes à rythmes,
- soit en utilisant des techniques pour les placer. Une technique ainsi souvent utilisée est de placer un ornement sur des groupes de temps impairs, ce qui produit un décalage rythmique par rapport à la section qui joue en 4/4 et une sensation de polyrythmie. Cela revient par exemple à jouer des croches pointées à la suite sur une grille en 4/4.

On notera enfin une différence importante qui permet souvent de distinguer facilement la *House* et la *Techno* :

- La *Techno* est souvent binaire, c'est-à-dire que les croches sont strictement pla-

cées dans des intervalles de signature rythmique 4/4.

- La *House*, en revanche, est souvent appelée de manière impropre "ternaire". S'il ne s'agit pas à proprement parler d'une signature ternaire, on divise chaque intervalle en croches ternaires et on décale les doubles croches binaires vers la troisième double croche ternaire. La logique reste ainsi binaire (car le temps est subdivisé en multiples de 4), mais le ressenti diffère.

Ces deux aspects se réfèrent à une notion nommée "groove", en référence au potentiomètre qui permettait de déplacer les doubles croches vers l'avant ou l'arrière sur les boîtes à rythme. Il semble nécessaire que le module puisse distribuer le reste des éléments rythmiques en suivant le "groove" déjà en œuvre dans le morceau, ou si ce groove n'est pas encore apparent, qu'il puisse faire un choix. Il semble important que ce choix soit réduit au "groove" binaire ou "groove" ternaire pour faciliter l'interaction avec le musicien.

11.2.2 Le corpus de règles tonales, harmoniques et mélodiques

La *Techno* comme la *House* sont relativement libres de critères tonaux. Il paraît cependant important d'essayer de se fixer certains critères simples : si le module doit jouer une mélodie, il se doit d'essayer de respecter certaines règles tonales pour garder une certaine harmonie musicale. Cette harmonie peut se concevoir de manière "modale", ou bien "tonale".

La manière la plus simple de fixer cette harmonie est de fixer une tonalité, qui interdit certaines notes au module. On peut adjoindre à cette tonalité différentes gammes pour changer la couleur harmonique de la mélodie que produit le module. Dans les

cas où la tonalité n'est pas encore fixée par un contexte musical, certaines pistes intéressantes apparaissent. Si l'on a pour l'instant seulement un instrument qui impose une note précise de basse, le module peut partir de là pour proposer une tonalité avec un motif mélodique, voire même proposer une tonalité en contradiction avec le contexte musical.

Il peut aussi proposer des ponts en se basant sur des règles tonales : certaines gammes comportent des notes communes, et il peut aussi, par exemple, si la tonalité est en La mineur, se mettre à jouer des notes appartenant à la gamme pentatonique de La pour permettre un pont vers une gamme de Mi mineur.

Il existe pour cela différents outils théoriques permettant au musicien de mettre en place des règles de changement de tonalité, comme le cercle des quintes.

Ce diagramme permet, à partir de la tonalité en cours, d'obtenir les tonalités voisines, c'est-à-dire les tonalités ayant le plus de notes en commun. Il est donc utile pour effectuer des changements de tonalités sans marquer une rupture trop importante entre deux passages de la performance, et en permettant à des mélodies issues de deux tonalités différentes de se chevaucher.

Une manière un peu plus complexe d'envisager l'harmonie est de la concevoir de manière "modale" et de constituer une grille d'accords en considérant chaque accord comme une situation unique en relation avec les accords qui le précèdent et qui le suivent. Cette approche, qui vient de la théorie musicale du *Jazz*, permet d'obtenir des motifs musicaux plus complexes, et d'introduire des notions de tensions/relâchement dans la mélodie.

Une des règles que l'on peut facilement mettre en place est relative à la redondance des grilles d'accords utilisées et des tonalités. Il paraît pertinent d'interdire au module de trop se répéter et de choisir plusieurs fois à la suite les mêmes accords ou les

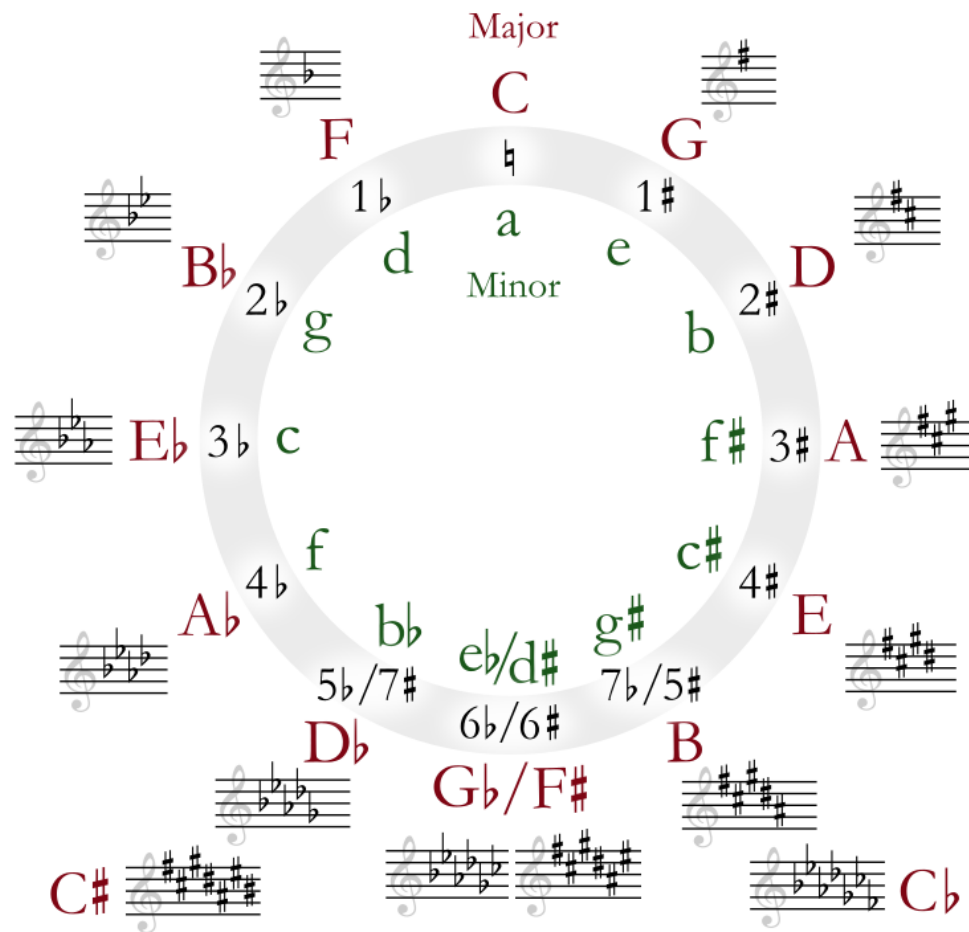


FIGURE 11.1 – Le cercle des quintes.

mêmes tonalités pour ne pas rendre la performance trop monotone.

11.2.3 Le corpus de règles sonores

Nous avons pour l'instant évoqué les règles sonores de manières à effectuer un classement des fonctions rythmiques des différents éléments qui composent habituellement les morceaux de *Techno*. Nous pouvons ensuite remarquer que les instruments utilisés en *Techno* ont des attributs sonores qu'il est possible de faire varier, comme le timbre ou l'enveloppe. En effet, de nombreux paramètres de synthétiseurs et d'effets permettent de modifier la nature sonore d'un motif musical, comme les filtres, et donc de créer des variations d'un motif donné, ou bien d'effectuer un mouvement.

On peut ainsi indiquer au module d'effectuer divers mouvements de caractéristiques sonores en fonction de la partie du morceau, comme le fait de faire monter la fréquence de coupure d'un filtre lors d'une montée. On peut aussi évoquer l'idée de règles quand à la répartition spectrale du morceau joué : il paraît pertinent d'éviter de jouer des notes graves lorsque le bas du spectre est déjà rempli, ou des notes aiguës lorsque le haut du spectre est déjà rempli.

11.2.4 Le corpus de règles narratives

On peut noter une première convention très répandue dans la musique *Techno* : celle de répartir le temps en cycles de 16 ou 32 mesures. On remarque ainsi que la plupart des changements sonores majeurs (ajout d'un instrument, intensification soudaine) sont positionnés toutes les 16 ou 32 mesures dans les morceaux de *Techno*. Cette convention est liée à la pratique du *DJing*. Les "DJs" effectuent sans cesse des transitions entre deux (ou plus dans de rares cas) morceaux, et jouent un nombre élevé de titres lors d'un "DJ set". Il est donc difficile de connaître le détail de l'organisation de chaque morceau joué² et le fait d'avoir des cycles longs de longueur souvent égale permet au DJ de garder en tête un comptage des mesures simple et d'être sûr que la survenue d'évènements sur deux morceaux jouant simultanément se fera de manière synchronisée.

Cette convention s'est propagée dans les performances de *Techno*, les musiciens comme le public étant ainsi, à force, éduqués à compter facilement ces cycles longs. Cette convention est en revanche très souvent brisée pour casser la monotonie et introduire de la surprise, mais lorsqu'elle l'est, c'est précisément pour briser cette convention et non pas pour l'ignorer. A ces longs cycles, nous ajoutons un cycle court "de sécurité", que l'on appelle cycle harmonique, dans le cas où nous aurions un motif musical de 4 ou 8 mesures se répétant, pour éviter un drop soudain en plein milieu d'un cycle. On distinguera, en termes de narration, 3 parties qui peuvent s'enchaîner.

- La "montée", où l'on cherche à construire une tension. On peut ainsi utiliser de nombreux codes musicaux usuels dans ces styles. Il y a déjà les codes qui permettent d'introduire du mouvement, comme le fait d'augmenter les envois des

2. cette difficulté était d'autant plus présente avant l'arrivée des platines CD. Il est impossible de voir à l'avance l'organisation d'un morceau sur un vinyle, contrairement à une platine CD où l'on peut voir la forme d'onde du morceau et deviner quelle partie arrive à quel moment.

divers instruments vers des reverb ou des delays, ou de saturer la section rythmique. Il y a ensuite les codes qui vont permettre le contraste avec le "drop", comme le fait d'essayer d'alléger au maximum le bas du spectre.

- Le "drop", où la tension est relâchée. Cette partie comporte un code impératif : il s'agit du moment où le kick commence à jouer son motif rythmique principal. On constate que cette partie peut s'envisager de deux manières : elle peut être très épurée, pour marquer un contraste fort avec une montée riche en éléments musicaux, tout comme elle peut être au contraire saturée d'éléments, pour représenter un "climax".
- Le "pont" pour sortir du drop et permettre une transition soit vers une autre montée, soit vers la fin de la performance.

Ces parties ne constituent pas une méthode générale pour construire un morceau. Le nombre de drops, leur longueur et leur place dans la structure varient grandement selon les morceaux. On peut en revanche diviser la structure de tous les morceaux de *Techno* et de *House* en une successions de ces différentes parties. On notera aussi l'utilisation d'évènements narratifs : les "breaks", qui servent à briser la monotonie de n'importe quelle partie et à introduire une variation locale, et dans le cas du drop, à relancer la tension en créant une variation brusque.

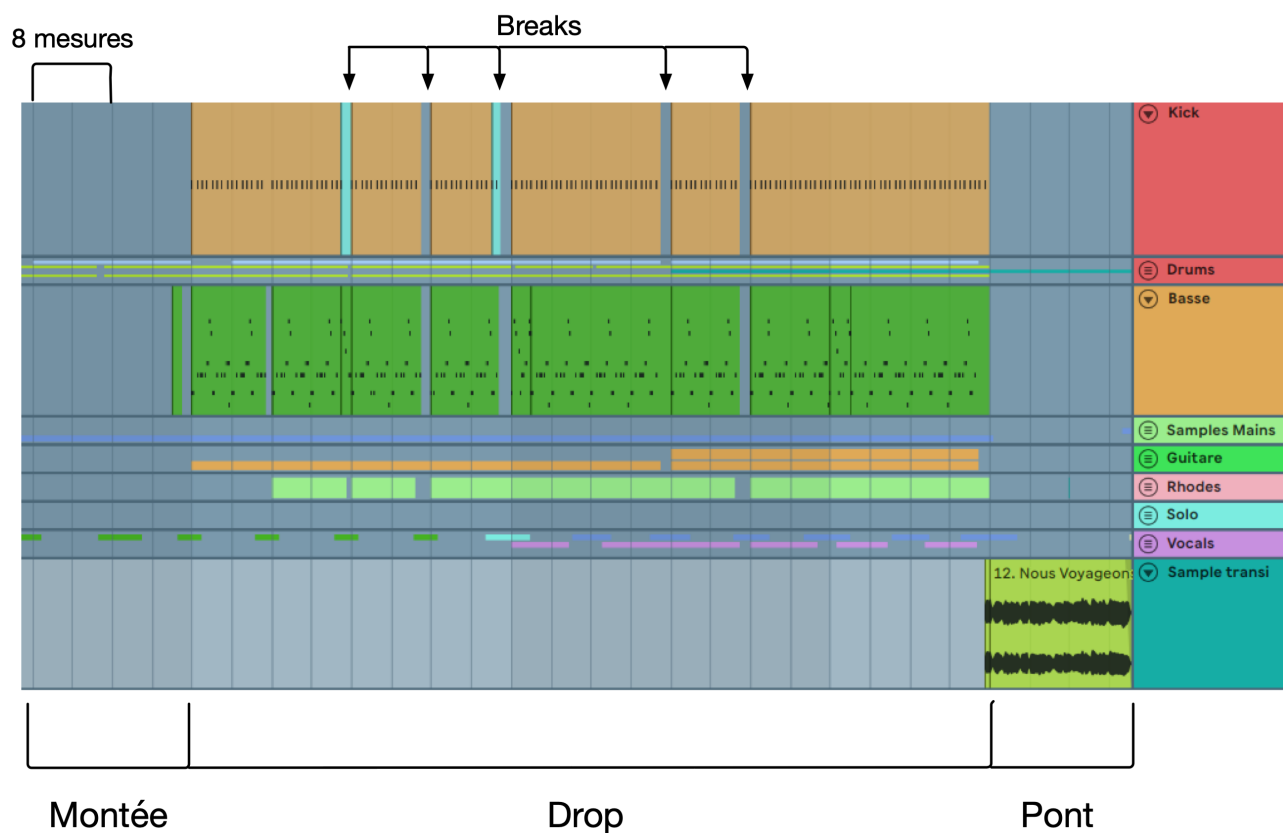


FIGURE 11.2 – Schéma de l'organisation d'un morceau de *House* à partir de la vue arrangement de son set *Ableton Live*

Au delà de ces parties qui permettent de rendre compte du mouvement et de la narration à un moment donné, il paraît important de s'interroger sur la manière d'envisager l'intro de la performance, ainsi que son outro.

Certaines performances commencent ainsi directement par des "drops" quand d'autres commencent par des montées. Il en va de même pour les outros : certaines performances se finissent brusquement tandis que d'autres utilisent des "ponts" pour terminer progressivement.

L'introduction représente un moment crucial de flottement comparé au reste de la performance. Elle pose une question importante : combien de temps attendre avant

d'effectuer le premier drop? Les possibilités sont nombreuses et peuvent dépendre de la vitesse à laquelle le musicien met en place des éléments musicaux, ou encore du contexte de l'évènement (il paraît logique, si le musicien joue en plein milieu de la nuit après des *DJs* ayant fait des sets assez rapides et agressifs, qu'il ne prenne pas trop de temps pour mettre en place le premier drop afin de ne pas perdre le public. Une fois que la mécanique générale de cycles longs avec des drops et des breaks est en place, il paraît moins complexe pour le module, ou pour le musicien, d'envisager le choix des changements de parties.

Chapitre 12

Conclusion

Créer un corpus de règles à formaliser sous forme d'algorithme amène, comme nous l'avons vu, à repenser les habitudes de composition d'un musicien. Certaines de ces règles ne sont pas suivies de manière consciente bien qu'elles structurent de manière implicite la performance. Il apparaît ainsi qu'un module tel que nous l'avons théorisé pourrait prendre des décisions qu'un humain ne prendait pas naturellement. Cela peut être dangereux dans le cadre d'une performance en amenant des choix qui dénotent musicalement avec ce que le musicien joue. Mais cela peut aussi amener à des choix contre-intuitifs et ouvrir des nouvelles voies de création. Le fait de travailler sur des intentions de jeu et de composition permet de dépasser le simple travail d'agencement des notes et de s'orienter vers un concept que l'on appelle la "méta-composition". La méta-composition renvoie au niveau d'abstraction et de complexité du matériau qu'un musicien manipule durant une performance ou un processus de composition. Le *DJing* que nous avons évoqué à de nombreuses reprises représente une forme de méta-composition, étant donné que le *DJ* manipule la forme la plus complexe de contenu musical possible (le morceau lui même) et agence ce contenu en fonction d'intentions qui dépassent les considérations musicales de base. Dans une époque où l'automatisation des tâches se retrouve dans

de nombreuses formes d'art et d'artisanat, la méta-composition permet de faire de cette automatisation une liberté créative et d'approcher la création musicale d'une manière nouvelle.

Certains musiciens avec lesquels nous avons discuté durant l'élaboration de ce mémoire nous ont fait part de leur désir d'écrire leur performance à l'avance et de ne rien laisser au hasard. Pour eux, l'idée de déléguer des actions à un module partiellement imprévisible représente un danger et augmente le risque de "rater" la performance (ou du moins de présenter une performance qui paraît de qualité moindre que la version studio de leur travail).

Mais le concept du module que nous avons développé les intéresse néanmoins dans ce qu'il peut apporter dans la création préalable dans un studio, qui laisse beaucoup plus de place aux expérimentations. Même si l'on ne garde pas les idées proposées par le module, on peut considérer que les idées musicales qu'on a trouvées en réagissant au module auraient difficilement pu être trouvées dans d'autres contextes. Leurs commentaires sur l'idée d'un module indépendant dans une performance renvoyaient aussi à une logique liée aux attentes du public. L'un d'eux s'inquiétait ainsi du temps de réaction nécessaire à un comportement inattendu de la part d'un module, de peur de devoir faire attendre le public pendant qu'il réfléchit à une manière de réagir. Il paraît important de penser à l'avance à une manière pour le musicien ou pour le module de palier à ces temps d'attente durant lesquels le musicien réfléchit, ou effectue des tâches, sans qu'on puisse directement entendre un changement. Certains musiciens que nous avons interrogés nous ont en outre fait part d'une demande liée à leur envie de s'éloigner de l'utilisation d'un ordinateur dans une performance : il serait intéressant de concevoir un module comme un objet hardware que l'on pourrait intégrer physiquement dans la performance, et non pas comme un

élément logiciel à intégrer dans un "DAW".

L'idée nous est venue, lors de la réalisation du module, d'aller jusqu'au bout du concept de personnalité perçue d'un algorithme. On pourrait ainsi imaginer des performances dans lesquelles le module prend une indépendance telle qu'il n'est plus présenté comme le partenaire du musicien mais comme son adversaire. Ce module chercherait ainsi à pousser le musicien dans ses retranchements, à le forcer à abandonner certaines conventions musicales, et à le priver de certains repères dans le but de faire apparaître des nouvelles manières d'improviser et de réagir. Si cette idée nous paraît intéressante, c'est en partie parce qu'elle joue sur des thèmes communs en science-fiction dont l'esthétique est si chère à la *Techno*. Le thème de la machine esclave qui s'émancipe, se rebelle et développe des intentions propres renvoie ainsi à deux aspects de la société moderne que le mouvement *Techno* cherche à remettre en question : la surconsommation et la surproduction qu'elle entraîne, ainsi que la déshumanisation liée à l'utilisation intensive de machines et d'algorithmes au service de l'efficacité. On notera cependant qu'il n'est pas forcément utile d'associer systématiquement l'utilisation d'algorithme en musique à une esthétique futuriste, ni même à un propos sur le rapport entre l'homme et la machine.

On voit que le module ne se contente pas de simplifier le travail du musicien ou de lui proposer des idées créatives. On peut s'intéresser à la manière de l'intégrer dans une performance et à ce que sa mise en scène peut raconter. Il permet alors de développer des thèmes divers et variés.

L'aspect ironique de l'automatisation des tâches mise au service d'un propos sur l'automatisation des tâches paraît résonner avec les différentes questions qui apparaissent dans l'industrie musicale à l'heure où chacun peut avoir accès à des moyens de production musicale avancés. On voit souvent revenir des questions légales qui se transforment en questions philosophiques. Certaines affaires judiciaires liées au plagiat ont ainsi poussé certains à s'interroger sur l'aspect fini de la composition musicale : deux musiciens américains ont récemment réalisé un programme capable de générer jusqu'à 300 000 mélodies par seconde, dans le but de réaliser un immense catalogue qu'ils pourraient déposer pour en détenir les droits [48]. Le but de la démarche était de questionner le public sur l'idée de plagiat en considérant que le nombre de mélodies que l'humanité peut réaliser est nécessairement fini.

C'est là que l'automatisation et la méta-composition apparaissent être des manières d'aborder le sujet dans l'art. Voir l'étape d'agencement des notes comme triviale et moins importante que le travail d'intention permet de se rattacher à cette idée de "fin de la créativité". Le recours des compositeurs de *Techno* et de *House* au sampling est d'ailleurs parfois vu comme une prise de position sur cette idée. Si tout a déjà été composé, il paraît plus intéressant d'assumer le rôle du compositeur comme celui de quelqu'un qui ré-agence et qui re-contextualise.

Nous avons peu traité de l'aspect sonore de la composition, préférant nous concentrer sur l'aspect musical. Il semble pourtant qu'il s'agisse d'un champ de recherche riche et large. Le travail de méta-composition et d'abstraction nous paraîtrait tout

aussi pertinent dans le cadre de musiques s'intéressant plus à la texture sonore qu'au contenu musical, comme l'*Ambiant*.

Une autre idée qui nous est venue serait celle d'un environnement de programmation dédié à la réalisations de modules, permettant d'intégrer soi-même facilement des règles dans un certain ordre de priorité ou bien de constituer des arbres décisionnels, séparant les 4 étapes de fonctionnement. L'intérêt serait de pouvoir réaliser soi-même plusieurs modules, voire éventuellement de les faire communiquer entre eux. En effet, les idées que nous avons évoquées au cours de ce mémoire sont limitées par les compétences techniques du musicien qui souhaiterait les mettre en oeuvre. Un environnement de programmation simplifié permettrait alors aux musiciens néophytes en informatiques de s'emparer de ces outils.

Chapitre 13

Bibliographie

- [1] Kembrew McLeod. Genres, subgenres, sub-subgenres and more : Musical and social differentiation within electronic/dance music communities. *Journal of Popular Music Studies*, 13(1) :59–75, 2001.
- [2] La Fougère. Quelques idées reçues sur l’histoire de la house. *Audimat*, 2(2) :63–83, 2014.
- [3] Guillaume Kosminci. *La musique Techno : approches anthropologiques*. CIREN, Université Paris VIII, 2001.
- [4] Jean-Yves Leloup. *Les Basiques : La musique Electronique*. Leonardo/Olats, 2011.
- [5] Renaud Epstein, Jean Gaudillère, Irène Jami, Patricia Osganian, and François Ribac. Techno, une histoire de corps et de machines. *CAIRN* <https://www.cairn.info/revue-mouvements-2005-5-page-5.htm>, 2005. [En ligne ; lu le 17 mai 2020].
- [6] Alain Feron. Musique algorithmique. *Encyclopædia Universalis*. <https://www.universalis.fr/encyclopedie/musique-algorithmique>, 2014. [En ligne ; lu le 4 mars 2020].
- [7] Richard H. Hoppin. *Medieval Music*. Norton, 1978. ISBN : 978-0393090901.
- [8] André Hodeir. *Les formes de la musique*. Que sais-je? n°478. Presses Universitaires de France, 1951. ISBN : 978-2130594796.
- [9] Gerhard Nierhaus. *Algorithmic Composition : Paradigms of Automated Music Generation*. Springer, 2009. ISBN : 978-3211755396.
- [10] Julien Falk. *Technique de la Musique Atonale*. Editions Musicales Alphonse Leduc, 1959. ISBN : 979-0046222290.

- [11] John A. Maurer. A brief history of algorithmic composition. *Center for Computer Research in Music and Acoustics*. <https://ccrma.stanford.edu/~blackrse/algorithm.html>, 1999. En ligne; lu le 9 juin 2020.
- [12] Olivier Messiaen. *Quatres études de Rythme*. Durand, 2008. ISMN : 979-0044081097.
- [13] James Peter Burkholder. *All Made Of Tunes : Charles Ives and the Uses of Musical Borrowing*. Yale University Press, 1995. ISBN : 978-0300102123.
- [14] Jean-Yves Bosseur. A propos de pithoprakta (1955-56). *Les Amis de Xenakis*. <https://www.iannis-xenakis.org/fxe/actus/bosseur.html>, 2013. [En ligne; lu le 25 juin 2020].
- [15] Lejaren Hiller and Leonard Isaacson. *Experimental Music : Composition with an algorithmic Computer*. 1959.
- [16] Moreno Andreatta. *Musique Algorithmique*. IRCAM, 2009.
- [17] Anonyme. Brain computes new tune for tv. *New York Times*, page 51, 1956. 3 juillet 1956.
- [18] Nicolas Viel. Pierre barbaud et la naissance de la musique par ordinateur en france : De la cybernétique à l'algorithmique. Université Paris 4 Sorbonne, 2007.
- [19] Pierre Barbaud. *Schoenberg*. Editions de la Main d'Oeuvre, 1997. ISBN : 978-2911973017.
- [20] Brian Eno. *Journal, une année aux appendices gonflés*. Le Serpent À Plumes, 1998. ISBN : 978-2842610920.
- [21] Kemal Ebcioglu. *An Expert System for Choral Harmonization*. AAAI, 1986.
- [22] David Cope. Experiments in musical intelligence. *Site internet de David Cope* <http://artsites.ucsc.edu/faculty/Cope/experiments.htm>, 2005. [En ligne; lu le 17 mai 2020].
- [23] Steph Kretowicz. Algorave : the live coding movement that makes next level electronic music. *MixMag* <https://mixmag.net/feature/algorave///>, 2017. [En ligne; lu le 25 juin 2020].
- [24] Jonas Downey. *Glitch Art*. University of Illinois, 2002.
- [25] Jérôme Nika. *Guiding human-computer music improvisation : introducing authoring and control with temporal scenarios*. PhD thesis, Université Pierre et Marie Curie, 2016.
- [26] Russ Hepworth-Sawyer, Jay Hodgson, Justin Paterson, and Rob Toulson. *Innovation in Music : Performance, Production, Technology and Business*. Routledge, 2019. ISBN : 978-1351016698.
- [27] Alissa Barna. Bach-lash! *Slate* <https://slate.com/technology/2019/03/google-doodle-bach-ai-music-generator.html>, note = " [En ligne; lu le 23 juin 2020]", 2019.

- [28] Peter Kirn. Automated techno : Eternal flow generates dance music for you. *cdm* <https://cdm.link/2019/04/automated-techno-eternal-flow/>, 2020. [En ligne; lu le 9 juillet 2020].
- [29] Alain Feron, André-Pierre Boeswillwald, and Pierre-Paul Lacas. Improvisation musicale. *Encyclopaedia Universalis*, 2014. [En ligne; lu le 20 juin 2020].
- [30] Geber L. Ramalho, Pierre-Yves Rolland, and Jean-Gabriel Ganascia. An artificially intelligent jazz performer. *Journal of New Music Research*, 28(2) :105–129, 1999.
- [31] Maximos Kaliakatsos-Papakostas, Andreas Floros, and Michael Vrahatis. Intelligent real-time music accompaniment for constraint-free improvisation. *Proceedings - International Conference on Tools with Artificial Intelligence, ICTAI*, 1 :444–451, 11 2012.
- [32] Giordano Cabral, Izabel Zanforlin, Rodrigo Lima, Hugo Santana, and Geber Ramalho. Playing along with d'accord guitar. 2020.
- [33] Jérôme Nika, Ken Déguernel, Axel Chemla-Romeu-Santos, Emmanuel Vincent, and Gérard Assayag. Dyci2 agents : merging the "free", "reactive", and "scenario-based" music generation paradigms. *International Computer Music Conference*, 2017.
- [34] Greg Surges and Shlomo Dubnov. Feature selection and composition using pyoracle. AAI Workshop, 2013.
- [35] Joel Chadabe. *Electric Sound : The Past and Promise of Electronic Music*. Pearson, 1997. ISBN : 978-0133032314.
- [36] Rene Wooller, Andrew R. Brown, Eduardo Miranda, Rodney Berry, and Joachim Diederich. A framework for comparison of process in algorithmic music systems. *Generative Arts Practice*, 2005.
- [37] Horacio Vaggione. Some ontological remarks about music composition processes. *Computer Music Journal*, pages 54–61, 2001.
- [38] Jeff Pressing. Improvisation : Method and model. *Generative process in music*, 1988.
- [39] P. N. Johnson-Laird. How jazz musician improvise. *Music Perception*, 2002.
- [40] Anonyme. Madplayer - interactive digital music player. *PC Hardware* <https://www.pchardware.co.uk/madplayer.php>, 2005. En ligne; lu le 9 juin 2020.
- [41] Andrew R Brown and Thoring Kerr. Adaptive music techniques. *Improvise : The Australian Computer Music Conference*, pages 26–31, 2009.

- [42] Karen Collins. An introduction to procedural music in video games. *Contemporary Music Review*, 2009.
- [43] Georges Lewis. Too many notes : Computers, complexity and culture in voyager. *Leonardo Music Journal*, pages 33–39, 2000.
- [44] Toby Gifford, Shelly Knotts, Jon McCormack, Stefano Kalonaris, Matthew Yee-King, and Mark d’Inverno. Computational systems for music improvisation. *Digital Creativity*, 29 :19–36, 01 2018.
- [45] Matthew John Yee-King. An automated music improviser using a genetic algorithm driven synthesis engine. *University of Sussex*, 2007.
- [46] Martin Norgaard, Mariana Montiel, and Jonathan Spencer. Chords not required : incorporating horizontal and vertical aspects independently in a computer improvisation system. *Proceedings of the International Symposium on Performance Science*, 2013.
- [47] Marcio Dahia, Hugo Santana, Ernesto Trajano, Geber Ramalho, Carlos Sandroni, and Giordano Cabral. Using patterns to generate rhythmic accompaniment for guitar. *Proceedings of Sound and Music Computing*, 2004.
- [48] Anthony Cuthbertson. Musician uses algorithm to generate every possible melody to prevent copyright lawsuits. *The Independent* <https://www.independent.co.uk/music-copyright-algorithm-lawsuit-damien-riehl-a9364536.html>, 2020. [En ligne; lu le 9 août 2020].