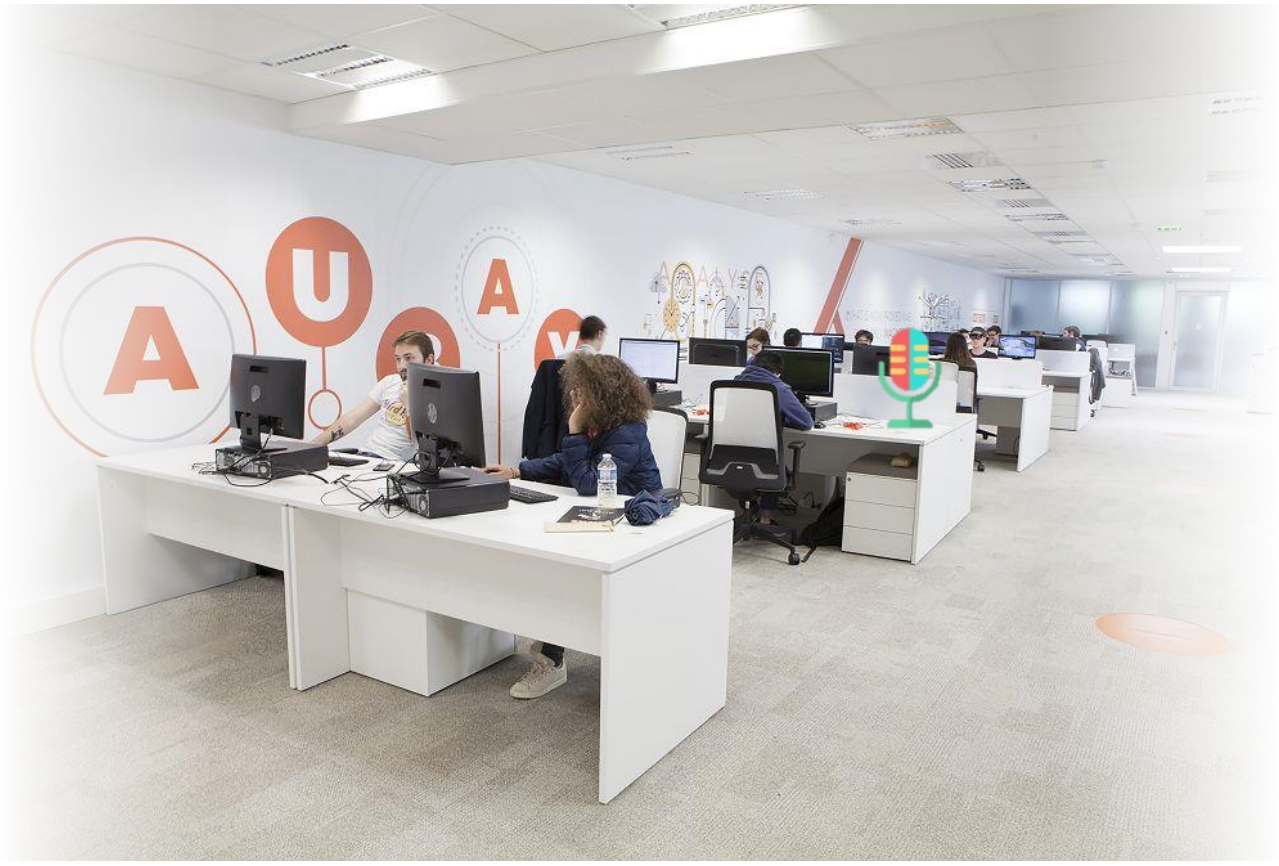

Rapport de PFE chez Aubay

Génération de musique

Encadrant
entreprise :
Eric Remilleret

Alexandre Argento
alexandre.argento@ensta-bretagne.org

ENSTA Bretagne
2 F. Verny street
29806 BREST Cedex 9
France

Année 2019-2020
Semestre 2
Profil SPID
Spécialité Robotique

Remerciements

Je tiens à remercier tout d'abord mes encadrants de projet Didier Giot, Eric Remilleret et Arjen Kraneveld pour leur soutien tout au long du projet. J'ai beaucoup apprécié les nombreux échanges que j'ai eu avec eux, tant lorsqu'il s'agissait de discuter d'aspects complexes du projet que pour l'ambiance conviviale qu'ils ont entretenue avec nous.

Je remercie également les référents Céline Gouzin, Marianne Fernandez, Morgan Buisson et Sandrine Gouirand qui ont contribué de près ou de loin à l'avancement du projet dans de bonnes conditions.

Enfin, mes remerciements vont aux stagiaires avec qui j'ai pu construire un projet riche techniquement, en échangeant sur de nombreuses idées et des points de vue complémentaires : Louis Couturier, Arthur Feyt, Nathan Fradet, Mathieu Diep et Clarisse Charrier.

Abstract

Deep Learning music generation demands a lot of resource and training time in order to achieve a good result in generating music. The shortcomings of this first solution lie in the lack of consonance and apparent musical structure. To address this issue, it is necessary to use learning methods focused on specific musical aspects, linked together by rules of harmony. By decomposing a piece of music into two voices, a melody, and an accompaniment, we can reach for a fairly rich music, whose musical rules can be defined to ensure its consonance. These rules lie in the relationship between the notion of chord and the judicious choice of an associated range of notes.

Résumé

La génération de musique par Deep Learning est exigeante en termes de ressources et de temps d'entraînement nécessaires pour arriver à un résultat convenable en génération de musique. Les lacunes de cette première solution résident dans le manque consonnance et de structure musicale apparente. Pour remédier à cela, il est nécessaire de revenir à un apprentissage orienté sur des aspects musicaux spécifiques reliés entre eux par des règles d'harmonie. La décomposition d'une musique en deux voix, une mélodie et un accompagnement, permet d'avoir une musique assez riche, dont des règles musicales peuvent être définis pour assurer la consonnance de la génération. Ces règles résident dans la relation entre la notion d'accord et le choix judicieux d'une gamme de notes associée.

Table des matières

I. Introduction 5
II. Présentation d'Aubay 6
III. Etat de l'art 7
III.1. Musicologie 7
III.1.a) Physique 7
III.1.b) Aspect psychophysiologique 7
III.1.c) Outils de représentation musicale 10
III.2. Techniques de génération de musique 10
III.2.a) Génération par apprentissage 10
III.2.b) Utilisation de datasets préparés 11
IV. Organisation 13
IV.1. Diagrammes de Gantt 13
IV.1.a) Diagramme de Gantt prévisionnel 13
IV.1.b) Diagramme de Gantt réel 13
IV.2. Ressources humaines 14
IV.3. Communication, stockage et partage de données 15
IV.4. Développement en équipe 15
IV.4.a) Gestion des tâches 15
IV.4.b) Développement collaboratif 16
V. Conception et Analyses système 18
V.1. Approches en composition musicale 18
V.2. Légitimité d'une génération créative 19
V.3. Créativité et interface utilisateur 21
VI. Preuves de concept 24
VI.1. Modélisations successives avec Sonic Pi 24
VI.1.a) Développement 24
VI.1.b) Analyse et conclusions 25
VI.2. Génération de mélodies par matrice de Markov 26
VI.2.a) Approche préliminaire 26
VI.2.b) Principe 26
VI.2.c) Normalisation tonale 28
VI.2.d) Dataset 29
VI.2.e) Résultats 30
VI.3. Génération d'accords par générateurs de Markov 32
VI.4. Génération de mélodies par réseaux de neurones (inachevé) 33
VI.5. Harmonie mathématique 34
VII. Application mobile 37
VII.1. Cahier des charges 37
VII.1.a) Fonctionnement et organisation générale 37
VII.1.b) Fonctionnalités 37

VII.2. Serveur web 38
VII.3. Frontend 39
VIII. Conclusion 42
IX. Bibliographie 43
X. Table des illustrations 45
XI. Table des annexes 46

Glossaire

Terme	Définition
Accord	Jeu de plusieurs notes simultanément.
Backend	Partie de code dédiée au serveur web.
Dataset	Ensemble de données cohérentes et documentées utilisées pour faire de l'apprentissage.
Frontend	Partie de code dédiée à l'application mobile.
Gamme	Ensemble de notes situées sur une octave, à partir d'une note de référence.
Mesure	Temps élémentaire d'une musique basé sur des répétitions d'accords.
Mode	Répartition des notes d'une gamme parmi les 12 notes de l'octave.
Octave	Intervalle de rapport ½ entre deux fréquences.
Outlier	Valeur considérée comme extrême par rapport à un ensemble de données
Pitch	Note repérée par un numéro dans le format MIDI.
Tonalité	Information contenant le mode et la note de référence d'une musique.

I. Introduction

En tant qu'étudiant à l'ENSTA Bretagne dans la formation d'ingénieur généraliste, j'ai eu l'occasion, pendant un stage d'une période minimale de quatre mois, de travailler dans de l'innovation en l'entreprise. Ce rapport a pour but de présenter le travail que j'ai réalisé durant ce stage, qui s'est déroulé au sein de la cellule Innovation d'Aubay (INNOV), située à Boulogne-Billancourt, du 16 mars au 21 août 2020. Ce travail s'est porté sur la génération de musique.

J'ai choisi de me diriger vers Aubay afin de découvrir comment fonctionne une entreprise de services numérique et de participer à leur projet d'innovation dans le domaine de la musique. Le sujet du stage, Aubay Musical Playlist (AMP), était un moyen de concilier ma passion pour la composition et ma formation d'ingénieur. C'était aussi une occasion de construire un projet technique sur une problématique d'intelligence artificielle, où des notions humaines telles que la créativité ou la beauté sont questionnées.

Le sujet proposé a été affecté à une équipe de 9 stagiaires et demandait de créer une IA capable de produire une composition musicale qui plaise à tout le monde. Il a été traité en trois approches :

- La recommandation de musique, grâce à de la classification,
- La génération « libre » de musique, grâce au Deep Learning,
- La génération « guidée » de musique, grâce à des connaissances musicales.

Mon travail s'est axé sur la troisième approche, envisagée afin de combler certaines lacunes de la deuxième, déjà explorée par d'autres stagiaires à mon arrivée.

Dans une démarche de recherche et d'expérimentation donnant lieu à un « démonstrateur », présentant les résultats des travaux, le cadre du projet se traduit par une période de preuves de concepts (POC, pour Proof Of Concept), puis une phase de projet dédiée au développement d'une application mobile pour la présentation des travaux. Cette deuxième phase reflète l'activité d'Aubay par rapport à son service client.

Je vais donc présenter l'entreprise Aubay, avant de développer en cinq parties mon travail pendant ce stage :
Tout d'abord, je ferai l'état de l'art du projet pour mettre en lumière les points de départ envisagés.
Ensuite, je parlerai des aspects organisationnels, au niveau des personnes et des supports utilisés.
Je passerai en revue les différentes questions posées lors de phases de conception afin d'exposer les enjeux autour du sujet.
Enfin, je détaillerai mes travaux de POC, puis de développement de l'application de démonstration.

II. Présentation d'Aubay

Aubay est une Entreprise de Services Numériques créée en 1998, qui agit dans des secteurs tels que la finance, l'assurance, l'industrie et l'énergie.

Figure 1 – Répartition du chiffre d'affaires d'Aubay par secteur d'activité (en 2018) (Aubay, s.d.)

Le groupe Aubay s'est étendu en Europe par l'acquisition de nombreuses sociétés basées en France, puis en Espagne, en Belgique, au Luxembourg, en Italie et au Royaume Uni. Deux sites sont basés en France, un à Nantes et un à Boulogne-Billancourt.
A la fin de l'année 2019, le groupe compte plus de 6500 collaborateurs en Europe dont plus de 2500 en France.

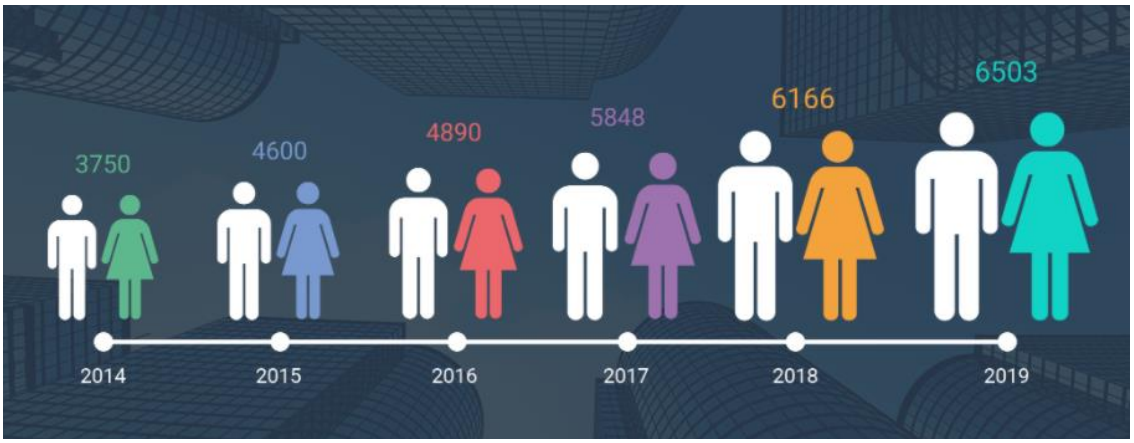

Figure 2 - Evolution des effectifs d'Aubay depuis 2014 (Aubay, s.d.)

Aubay a réalisé un chiffre d'affaire de 417,8 M€ en 2019, soit une croissance de 4.9% par rapport à 2018. Ses principaux clients sont BNB Paribas, Société Générale, Crédit Agricole, Allianz et Axa.

Depuis 2013, Aubay développe ses activités en R&D afin de tester de nouvelles technologies, d'améliorer son champ d'action et de promouvoir son image d'entreprise innovante. Ses domaines de recherche traitent des interfaces homme-machine, des techniques de traitement de données et de services intelligents tels que la détection d'émotions par image ou la navigation dans un réseau en réalité virtuelle. Le sujet AMP a été créé pour une première fois cette année dans le livret de stages Aubay.

III. Etat de l'art

III.1. Musicologie

III.1.a) Physique

La musique existe grâce à la propagation de sons. Et d'un point de vue physique, « Le son est une sensation auditive provoquée par une vibration » (Agence culturelle Grand Est, 2009).
Une musique est donc dépendante de trois facteurs :

- la source qui émet le signal sonore,
- le milieu de propagation du signal,
- le récepteur de l'information.

On peut différencier la musique d'un son quelconque par la présence d'une volonté, du côté de la source, d'émettre un signal sonore. Si cette volonté est la nôtre, alors le milieu de propagation et le récepteur sont des influences accidentelles auxquelles que nous devons prendre en compte.

Le milieu de propagation, s'il s'agit de l'air, aura une influence principalement du fait de la distance parcourue par le signal sonore et de la présence éventuelle d'autres signaux dans le milieu pouvant causer des interférences. Il est à noter cependant que le son se propage à la vitesse de 344 mètres par seconde. Ainsi, si nous voulons limiter le temps de réception du signal audio en-dessous d'un dixième de seconde, il faut alors se trouver dans un rayon de 34,4m de l'émetteur, ce qui est tout à fait acceptable dans des conditions d'écoute au sein d'un petit groupe de personnes. Cela peut représenter une différence plus notable pour un concert avec plus de 10 000 personnes mais ces dimensions sortent du cadre du projet.
Si la source est en mouvement, alors un décalage de fréquence se produit selon si l'on se rapproche de la source ou si l'on s'en éloigne. Il s'agit de l'effet Doppler.

Le récepteur, s'il s'agit de l'être humain, présente, lorsque nous prenons en compte le cerveau, des, spécificités peu évidentes, voire inconnues. Cependant, nous pouvons affirmer que nos deux oreilles nous permettent d'estimer la direction de la source sonore, d'une part de gauche à droite avec les intensités, d'autre part sur toutes les directions avec la façon dont le signal est distordu lorsqu'il entre dans nos oreilles (Lorenzi, 2019).
Nous noterons enfin que nous ne pouvons entendre des sons dont la fréquence n'est pas comprise entre 20 Hz et 20 000 Hz.

Nous ne considèrerons ici pas d'autres types de récepteurs et nous ne tiendrons par compte d'éventuelles interférences dans le milieu de propagation. Ceci étant dit, les phénomènes que nous venons de citer sont parfois reproduits par des producteurs de musique afin de donner des effets particuliers liés à l'espace sur l'auditeur.

III.1.b) Aspect psychophysiologique

Si l'on se place d'un point de vue commercial, ce qui stimule principalement le besoin de musique chez le client réside dans les effets qu'elles ont sur notre cerveau.
En scannant la présence de dopamine dans le cerveau grâce à l'injection d'un produit chimique, une étude a permis de montrer que l'écoute d'une musique pouvait provoquer la libération de dopamine dans le cerveau (Nature Neuroscience, 2011), la dopamine jouant un rôle dans la motivation et le bien-être d'une personne (Wikipédia, 2020). L'écoute de musique peut aussi être responsable de frissons, d'une augmentation du rythme cardiaque, de la respiration et de la transpiration (Nature Neuroscience, 2011).
Le potentiel de la musique s'étend même dans un domaine médical : la musicothérapie. En effet, par la recherche d'un cadre d'écoute favorable et de sonorités, la musique peut contribuer à « améliorer l'humeur », à « réduire l'anxiété », au « soulagement de la douleur », à « améliorer le sommeil » ou à aider des personnes atteintes d'autisme ou de schizophrénie (Medoucine, 2018).

La musique peut donc être utilisée à des fins médicales, mais aussi de marketing, en « influençant l'humeur des consommateurs » (Remisse, 2020). Cette stratégie est utilisée de façon identifiable par Apple dans ses publicités, ou alors la SNCF, par un jingle récurrent que l'on retient facilement. L'usage de la musique permet de créer à une entreprise une identité sonore qui va contribuer à rendre la marque familière et partie de la vie du consommateur. On appelle cela le « Design sonore ».

Pour caractériser un peu plus les mécanismes autour de la perception de la musique, un concept plus élémentaire serait celui de la consonance. Il intervient notamment dans le fait de construire une musique non seulement avec des sons différents, mais de les jouer en même temps et lui donner ainsi une dimension harmonique.

Le système de notes MIDI actuel, l'accordage d'un piano d'une guitare et la position de ses frettes, ou, plus généralement, l'établissement des notes Do, Do dièse, Ré, Ré dièse, Mi, Fa, Fa dièse, Sol, Sol dièse, La, La dièse et Si, ont tous pour origine d'anciens travaux effectués à l'aide des mathématiques.
Ces travaux, dont le document de Pascal et Tomas (Pascal & Tomas, 2000) détaille les étapes, remontent au Ve siècle av. J.C. grâce à notamment Pythagore, puis Zarlino pendant la Renaissance ou Euler au XVIIIe siècle.
Ces travaux partent initialement des observations suivantes :

- Loi de Weber-Fechner (Wikipédia, 2018) : L'oreille est sensible non pas aux différences, mais aux rapports de fréquences des sons.
- Deux sons sont consonants lorsque l'oreille n'éprouve pas la nécessité de les dissocier.

A partir de cela, des propriétés de consonnances de sons ont pu être mises en évidence par la capacité ou non d'un auditeur de pouvoir dissocier deux sons. En particulier, des sons de rapport 1 (unisson), ½ (octave), 2/3 (quinte), ¾ (quarte), 4/5 (tierce), 8/9 (seconde) ont une certaine chance d'être confondus et sont perçu comme plus consonants que d'autres, comme l'on peut voir d'après la . Ce tracé a été obtenu d'après le concept d'entropie harmonique de Paul Erlich (Wikipédia, 2020). Ce concept passe par la modélisation mathématique d'effets liés au fonctionnement de notre oreille.

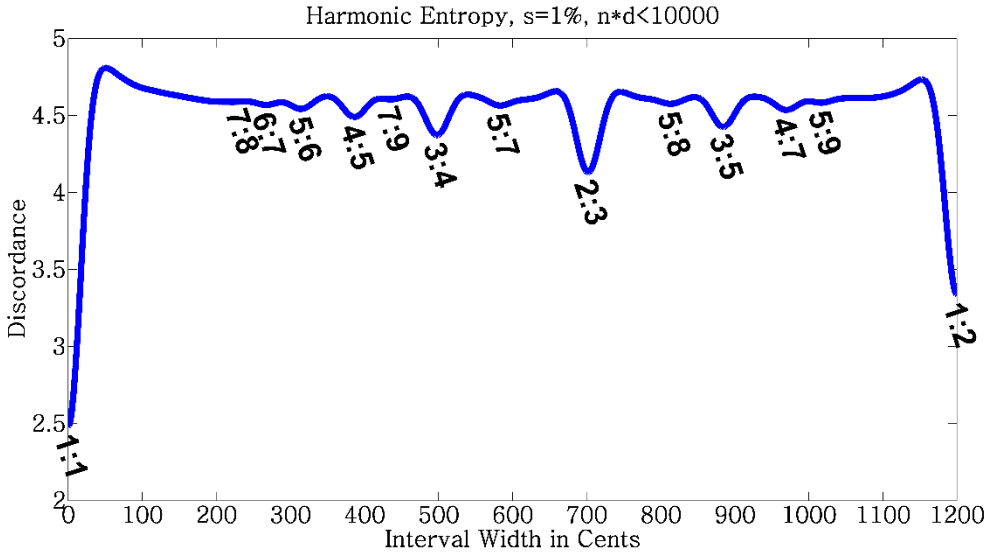

Figure 3 - Modèle de perception des dissonances entre deux sons (Wikipédia)

Une étude similaire de Carl Stumpf, d'après l'étude de Pascal et Tomas (Pascal & Tomas, 2000), montre que d'un échantillon d'auditeurs :

- 75% d'entre eux confondent les octaves,
- 50%, les quintes,
- 33%, les quartes,
- 25%, les tierces,

- 10%, les secondes.

Dans la recherche de notes (sons d'une certaine fréquence) comprises dans l'intervalle [f, 2f[qu'on appellera gamme, Pythagore se base sur un cycle de quintes qu'il ramène dans l'octave. Il arrive à construire plusieurs gammes en se basant sur le fait d'arrêter le cycle de quintes une fois que l'on est assez proche d'un bouclage en f ou 2f. Parmi ces gammes, celle de la Figure 4 se rapproche de la gamme actuellement utilisée, avec un découpage de l'octave en 12 notes.

n	0	7	2	9	4	11	6	1	8	3	10	5
f_n	f	$\frac{2^8}{3^5}f$	$\frac{3^2}{2^3}f$	$\frac{2^5}{3^3}f$	$\frac{3^4}{2^6}f$	$\frac{2^2}{3}f$	$\frac{3^6}{2^9}f$	$\frac{3}{2}f$	$\frac{2^7}{3^4}f$	$\frac{3^3}{2^4}f$	$\frac{2^4}{3^2}f$	$\frac{3^5}{2^7}f$
	DO	RE^b	RE	MI^b	MI	FA	$FA^\#$	SOL	LA^b	LA	SI^b	SI

Figure 4 - Notes de la gamme chromatique de Pythagore (Pascal & Tomas, 2000)

Cette gamme présente le désavantage de ne pas avoir de rapports de fréquences constants entre les notes (Figure 5), ce qui rend une musique impossible à jouer sur différentes hauteurs, à moins de réaccorder son instrument.

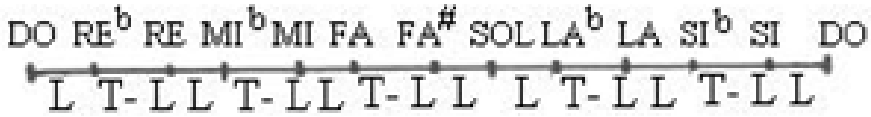

Figure 5 - Représentation linéaire de la gamme chromatique de Pythagore (Pascal & Tomas, 2000)

La gamme actuelle résout ce problème en fonctionnant avec un rapport constant de $2^{1/12}$. Cette gamme est appelée gamme au tempérament égal, ou gamme tempérée (Figure 6).

	1	2	3	4	5	6	7	8	9	10	11	12
Fréquence	f	$2^{1/12}f$	$2^{1/6}f$	$2^{1/4}f$	$2^{1/3}f$	$2^{5/12}f$	$2^{1/2}f$	$2^{7/12}f$	$2^{2/3}f$	$2^{3/4}f$	$2^{5/6}f$	$2^{11/12}f$
Nom de la note	DO	$DO^\#$ RE^b	RE	$RE^\#$ MI^b	MI	FA	$FA^\#$ SOL^b	SOL	$SOL^\#$ LA^b	LA	$LA^\#$ SI^b	SI

Figure 6 - Notes de la gamme tempérée (Pascal & Tomas, 2000)

Cette gamme possède donc le désavantage de ne pas être parfaitement réglée par rapport à des rapports de quintes parfaites ou d'octaves, les décalages des fréquences de ses notes par rapport à celles de la gamme de Pythagore restent très légers et peu perceptibles à l'oreille.

En plus des nombreuses gammes que propose la suite de Pythagore, d'autres gammes a 12 notes ont été créées à partir de celle de Pythagore pour l'améliorer qualitativement. Par exemple, la gamme Zarlinienne part de la quinte et de la tierce pour définir l'accord parfait majeur et travailler avec ses quintes ascendantes et descendantes. Le résultat est représenté dans la Figure 7.

n	0	1	2	3	4	5	6	7	8
f_n	f	$\frac{2^4}{3\times5}f$	$\frac{3^2}{2^3}f$	$\frac{2\times3}{5}f$	$\frac{5}{2^2}f$	$\frac{2^2}{3}f$	$\frac{5^2}{2\times3^2}f$	$\frac{3}{2}f$	$\frac{2^3}{5}f$

9	10	11
$\frac{5}{3}f$	$\frac{3^2}{5}f$	$\frac{3\times5}{2^3}f$

Figure 7 - Notes de la gamme chromatique de Zarlino (Pascal & Tomas, 2000)

En voici la représentation linéaire page suivante.

Figure 8 - Représentation linéaire de la gamme chromatique de Zarlino (Pascal & Tomas, 2000)

Euler, à son tour, proposera une autre méthode permettant de définir de nouvelles gammes, dont la gamme a 12 notes suivante:

n	0	1	2	3	4	5	6	7	8	9	10	11
f_n	f	$f\frac{3^3*5}{2^7}$	$f\frac{3^2}{2^3}$	$f\frac{3*5^2}{2^6}$	$f\frac{5}{2^2}$	$f\frac{3^3*5^2}{2^9}$	$f\frac{3^2*5}{2^5}$	$f\frac{3}{2}$	$f\frac{5^2}{2^4}$	$f\frac{3^3}{2^4}$	$f\frac{3^2*5^2}{2^7}$	$f\frac{3*5}{2^3}$

Figure 9 - Notes de la gamme chromatique d'Euler

La représentation linéaire correspondante est visible sur la Figure 8:

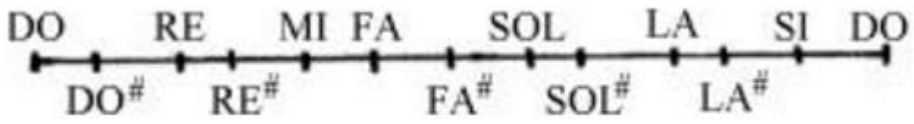

Figure 10 - Représentation linéaire de la gamme chromatique d'Euler (Pascal & Tomas, 2000)

III.1.c) Outils de représentation musicale

Le concept d'accord repose sur la définition de différentes manières de jouer des sons de façon simultané. Malgré les nombreuses possibilités de combinaisons de notes, des systèmes de notations d'accords existent afin d'en catégoriser une grande majorité.

La tonalité constitue également un concept important dans la création de musique. Il permet de repérer une gamme de notes, c'est-à-dire un sous-ensemble des notes de la gamme tempérée, dans laquelle les notes seront en consonnance. La détermination d'une tonalité vient souvent de l'identification d'accords dans un morceau de musique. Le fait de jouer un accord donne une certaine couleur, imposant des règles de consonance. De manière générale, il est possible d'utiliser des gammes à 7 notes (sur les 12 possibles) qui seront en consonance avec l'accord joué.

Différents formats permettent de stocker des données musicales tels que les formats MIDI, WAV ou MP3. Cependant, seul le format MIDI permet d'avoir des informations sur les notes jouées par différents instruments. Les notes MIDI y sont des entiers numérotés de 0 à 127 dont les fréquences suivent le découpage de la gamme tempérée.

III.2. Techniques de génération de musique

III.2.a) Génération par apprentissage

Le Deep Learning a permis d'avoir un modèle de génération de musique qui prend en compte les données musicales existantes et permettant de créer des clips musicaux de même taille sans règles prédéfinie. Cette méthode se base sur la modélisation de réseaux de neurones interconnectés, dont les poids des liens permettent de définir une relation déterministe entre une donnée d'entrée et une donnée de sortie. La phase d'entraînement du réseau de neurones permet de régler ces poids de façon à ce que la relation entrée-sortie soit représentative des données présentées.

Du fait du caractère déterministe d'un réseau de neurones, la création de musiques différentes se fait par l'injection d'entrées aléatoires dans le réseau de neurones. La notion de création ici réside dans le fait de pouvoir générer des données de sortie en envoyant en entrée d'autres valeurs que

celles du dataset d'apprentissage. Il s'agit idéalement de proposer un espace décrivant tous les points intermédiaires entre toutes les données d'entraînement.
Par analogie, le projet de la chaîne Youtube Code Parade montre l'influence de la variations de données d'entrées sur le résultat d'un réseau de neurones entraîné. Il est plus facile de comprendre le comportement de la sortie lorsqu'il s'agit d'images, et grâce à cela, le fait de déplacer les curseurs sur la droite de la Figure 11 permet de régler certaines caractéristiques du visage.

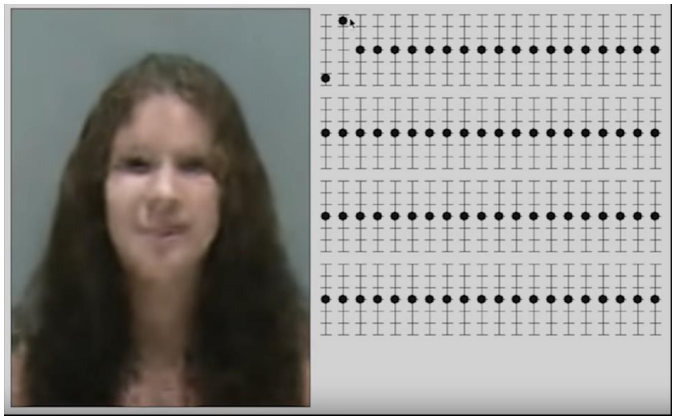

Figure 11 - Génération de visages par Deep Learning (Code Parade, 2018)

Le réseau de neurones de cet exemple illustre donc en quoi il est possible de créer de nouvelles musiques, par le choix aléatoire de valeurs sur les curseurs d'entrée.

Le groupe travaillant sur la génération libre s'est donc basé sur cette méthode pour générer de courtes introductions de musique. Ils ont ensuite envoyé cette musique, sortie du réseau de neurones, à nouveau dans l'entrée pour en générer la suite, conformément à un apprentissage sur des portions de musiques successives.

Le travail démarré par Louis à mon arrivée utilisait une matrice de probabilités appelée matrice de Markov afin d'apprendre des enchaînements d'accords et d'en générer de façon pondérée par rapport aux poids stockés dans la matrice. Cette méthode d'apprentissage repose sur le fait de considérer la génération de musique comme une succession d'états ayant chacun une certaine chance d'être atteint en fonction de l'état courant. Elle sera expliquée plus en détail dans la partie VI.2.b).

III.2.b) Utilisation de datasets préparés

De nombreux travaux de recherche ont consisté à trier, nettoyer, annoter et documenter des données afin de pouvoir s'en servir comme base d'apprentissage. Il est désormais possible de trouver de nombreux datasets avec chacun leurs spécificités. L'important étant de choisir celui qui répond le mieux à nos besoins, le recensement de datasets musicaux permet d'avoir une meilleure visibilité sur les données d'apprentissage, en évitant de perdre du temps lorsqu'il faut changer de dataset ou choisir le plus approprié.
Le travail de recensement que j'ai effectué, présent en annexe, a servi à la fois pour le choix d'un dataset pour l'entraînement de matrices de Markov et aussi pour l'entraînement de réseaux de neurones.
Les différents datasets répertoriés sont les suivants :

- JAAH (Eremenko, Demirel, Bozkurt, & Serra, s.d.)
- Isophonics (Isophonics, s.d.)
- McGill Billboard (DDMAL, s.d.)
- MAESTRO (Magenta, s.d.)
- 130 000 (midi_man, 2015)
- RS200 (de Clercq, Temperley, Lusting, & Tan, s.d.)
- Performed MIDI Dataset (fosfrancesco, s.d.)
- Band-in-a-Box Jazz (Tricotism, s.d.)
- Weimar Jazz (Pfleiderer, Frieler, Abeßer, Zaddach, & Burkhart, 2017)

- JazzParser (Jazz Parser, s.d.)
- Million song dataset (Bertin-Mahieux, P.W. Ellis, Whitman, & Lamere, 2011)
- Lakh MIDI Dataset (Raffel, s.d.)
- Lakh Pianoroll Dataset (Dong, Hsiao, Yang, & Yang, s.d.)

Le dataset retenu pour le POC de génération de mélodie a été le Weimar Jazz pour sa présence d'annotation de tonalité.

IV. Organisation

IV.1. Diagrammes de Gantt

IV.1.a) Diagramme de Gantt prévisionnel

Ce diagramme a été effectué lors de la première semaine de stage, avec toutes les informations disponibles du côté d'Aubay. Les estimations des dates butoirs et des événements organisés par Aubay se basent sur ce qui a été fait l'année précédente. Elles incluent :

- La période de présentation des preuves de concept de tous les stagiaires INNOV pendant une durée de deux semaines, pendant le mois de Juin.
- La Journée Des Stagiaires (JDS) qui est une journée dédiée à la présentation des projets INNOV à des personnes internes à Aubay. La période dédiée est généralement fin juillet.
- La soutenance de PFE : prévue entre fin août et début septembre.

A ce stade, et étant donné les circonstances de crise sanitaire, l'organisation des oraux des preuves de concept et de la journée des stagiaires était mise en cause dans leur format, mais maintenus malgré tout.

IV.1.b) Diagramme de Gantt réel

Au cours du stage, de nombreuses précisions et rectifications ont été apportées.
Parmi celles-ci, la JDS étant sensé durer une journée, s'est déroulé sur une semaine entière afin de prendre plus de temps pour les présentations et les questions et avoir un rythme moins soutenu.

De plus, et ce, par rapport à mes estimations, les périodes initialement dédiées à l'état de l'art, à la conception et à la preuve de concept ont été modifiées à cause de l'incertitude sur le choix d'une solution technique pour la génération de mélodies. Il a fallu une succession de deux séquences « état de l'art, preuve de concept » pour aboutir à un résultat satisfaisant. Une autre preuve de concept a succédé la deuxième pour la poursuivre et une quatrième a été entamée sans pouvoir aboutir avant la phase projet.

Les deux états de l'art identifiables sont les suivants :

- Etat de l'art 1 : intégration personnelle au sein du projet tel qu'il a été entamé, identification des choix techniques de chacun des 7 autres stagiaires du groupe.
- Etat de l'art 2 : Documentation de datasets musicaux afin de choisir lequel serait le plus adapté pour la génération de mélodie

Les preuves de concept seront détaillées dans la partie VI du rapport, en voici les correspondances sur le diagramme de Gantt :

- Preuve de concept 1 : Modélisation mathématique génération de mélodie + Sonic PI
- Preuve de concept 2 : Génération de mélodie par matrice de Markov + normalisation de mode
- Preuve de concept 3 : Génération d'accords par matrices de Markov
- Preuve de concept 4 : Génération de mélodies d'une mesure par Machine Learning
- Preuve de concept 5 : Modélisation mathématique de la consonnance et calcul des gammes automatisé.

Pour la phase de post projet, mes tâches étaient centrées sur la séparation des notes de mélodie et celles d'accompagnement dans la communication du serveur vers l'application. Cela a permis d'afficher avec différentes couleurs les notes de la mélodie et de l'accompagnement sur l'application. La partie documentation consistait à commenter correctement le code du serveur et de compléter le protocole de sa mise en place via AWS (Amazon Web Services).

Les périodes liées à la crise sanitaire causée par le Coronavirus sont également précisées et ont présenté chacune leurs contraintes particulières :

- **Confinement National :** Télétravail dès le mardi 17 mars, le lendemain de mon premier jour de stage. Toutes les réunions se sont effectuées par visioconférences, puis audioconférences pour limiter la consommation de la bande passante du réseau d'Aubay. L'ensemble de nos conversations de groupe ont été effectuées via Microsoft Teams, ainsi que nos appels et réunions avec l'équipe d'encadrement.
- **Déconfinement phase 1 :** Télétravail maintenu par Aubay. Préparation de la mise en place de VPN sur les ordinateurs d'Aubay. Le VPN , afin de pourvoir travailler à la fois à Aubay et chez soi sur l'ordinateur du site
- **Déconfinement phase 2 :** Tests VPN effectués par chacun des stagiaires, reprise du travail sur site à temps partiel, selon un emploi du temps permettant de travailler à une certaine distance des autres stagiaires. Notre groupe retourne travailler sur le site de Boulogne les lundi et mardi, puis les mercredi et jeudi après la phase projet.

IV.2. Ressources humaines

Mon tuteur de stage, Eric Remilleret étant responsable de tous les projets de la cellule INNOV, nous avions un responsable plus spécifique à notre projet, Didier Giot.
Des personnes référentes étaient disponibles pour nous aider sur des aspects précis :

- Assistance technique sur le projet : Arjen Kraneveld.
- Assistance documentation et état de l'art : Morgan Buisson.
- Accueil, intégration et vie sur le Campus : Céline Gouzin.
- Assistance administrative : Sandrine Gouirand.
- Design pour interface utilisateur : Marianne Fernandez.
- Support informatique : contact ESI (Equipe de Support Informatique).

Les premiers stagiaires à rejoindre le projet ont été Marine Chemeque Rabel, Gautier Peuziat et Florian Lafarge et ont commencé en janvier. Marine et Gautier ont travaillé sur de la recommandation de musique tandis que Florian s'est dirigé vers la génération de musique correspondant à l'approche de génération libre (usage de réseaux de neurones).
Ensuite, Mathieu Diep et Nathan Fradet l'ont rejoint à partir de février et ont mis en œuvre d'autres méthodes dans le même domaine.
A mon arrivée, le 16 mars, Louis Couturier travaillait sur la génération guidée depuis deux semaines et Arthur Feyt, depuis une semaine. Clarisse Charrier nous a rejoint par la suite, le 7 Avril, et a travaillé plus particulièrement sur la fusion de mélodies. Louis et moi avions tous les deux une dizaine d'années d'expérience en théorie musicale.

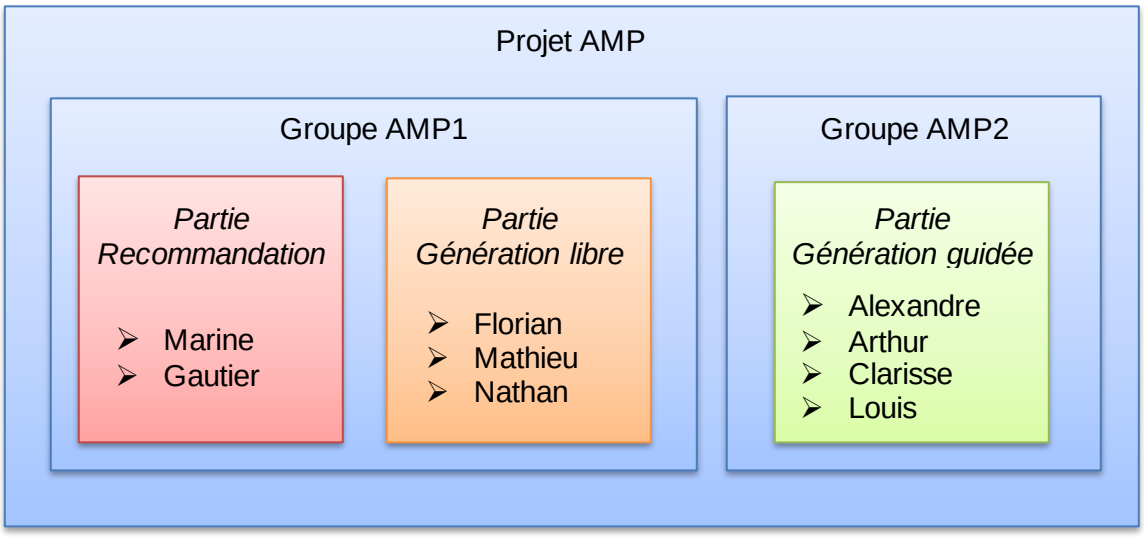

Figure 12 - Schéma récapitulatif du découpage du projet AMP

Les encadrants d'Aubay ont mis en place un suivi adapté à la période du stage :

- Preuves de concept (16/03 – 05/06) : deux réunions par semaine avec le groupe AMP2, une réunion « Comité de suivi » le mardi, de 11h20 à 12h dédiée à des discussions plus techniques, et une réunion « Stand up » le jeudi, de 16h à 16h15 afin de faire un point rapide et de signaler d'éventuels blocages. Présence d'Eric, de Didier et d'Arjen et occasionnellement Morgan.
- Phases projet (22/06 – 24/07) et post projet (03/08 – 21/08) : une réunion commune à toute l'équipe AMP le mercredi, de 10h30 à 11h30.

IV.3. Communication, stockage et partage de données

Divers supports ont servi à échanger et mettre à disposition différents types de données. Ces supports ont chacun eu des rôles différents dans la réalisation du travail de groupe. En voici la liste, avec leurs spécificités :

- Espace en ligne wiki-innov.aubay.io : espace interne à Aubay dédié à la présentation de la cellule INNOV et de son activité (responsables, stagiaire, projets, …). Des informations relatives aux pratiques de l'entreprise, à l'ensemble des outils utiles et aux contacts de l'administration y sont présentes. Des personnes externes peuvent y avoir accès avec un identifiant.
- Espace en ligne gipssi.aubay.com : saisie du temps de travail effectué, des absences et des notes de frais pour le calcul des salaires.
- Boîte mail webmail.aubay.com : messagerie interne, diffusion de messages du comité d'entreprise et de la direction, gestion d'événements sur calendrier avec liens vers l'application Teams.
- Application Teams (Microsoft) : Outil de discussion instantanée avec la possibilité de créer des groupes de discussion. Stockage de documents répertoriés par équipes de travail avec des arborescences de fichiers. Les fichiers de chaque groupe respectent le rangement suivant :
  - 1 – Prise en Main
  - 2 – Etat de l'art
  - 3 – Preuve de Concept
  - 4 – Projet
  - 5 – UX-UI
  - 6 – Suivi
  - 99 – A Classer

  UX fait référence à « **U**ser E**x**perience » et UI, à « User Interface », le dossier en question est destiné aux visuels de l'application créés en amont du développement. Le dossier de suivi est destiné aux comptes-rendus de réunion.
  Cette organisation permet essentiellement de répartir les documents produits en fonction de notre avancement dans le projet.
- Plateforme de développement gitlab.aubay.io : outil de sauvegarde des codes avec la gestion de version via les commandes Git.
- VPN Global Protect : accès sécurisé à l'ordinateur d'Aubay par l'intermédiaire de notre ordinateur personnel.

IV.4. Développement en équipe

IV.4.a) Gestion des tâches

Pour avoir une meilleure visibilité sur l'avancement de projets, nous avons utilisé un tableau tiré de la méthode agile : le tableau Kanban.

Le principe de la méthode agile est de décomposer son projet en plusieurs fonctionnalités ou niveaux de raffinements afin de viser d'abord le fonctionnement principal, puis d'en améliorer soit les performances, soit son domaine de fonctionnement.

Le remplissage du tableau pour gérer les tâches du projet implique de le décomposer en « épopées » contenant des « récits » qui se décomposent en « tâches ». Chacune de ces tâches peut se trouver dans une des quatre colonnes suivantes :

- Backlog : Tâches proposées par le groupe mais non prioritaires
- Sélectionné pour le développement : Tâches à réaliser
- En cours : Tâches en cours de réalisation
- Fini : Tâches terminées

Une tâche ne peut être assignée qu'à une seule personne, c'est pourquoi il faut bien décomposer le travail du projet de façon à avoir des tâches les plus indépendantes possibles les unes des autres.

La Figure 13 montre comment est représenté le tableau Kanban, avec la possibilité de déplacer les tâches d'une colonne à une autre.

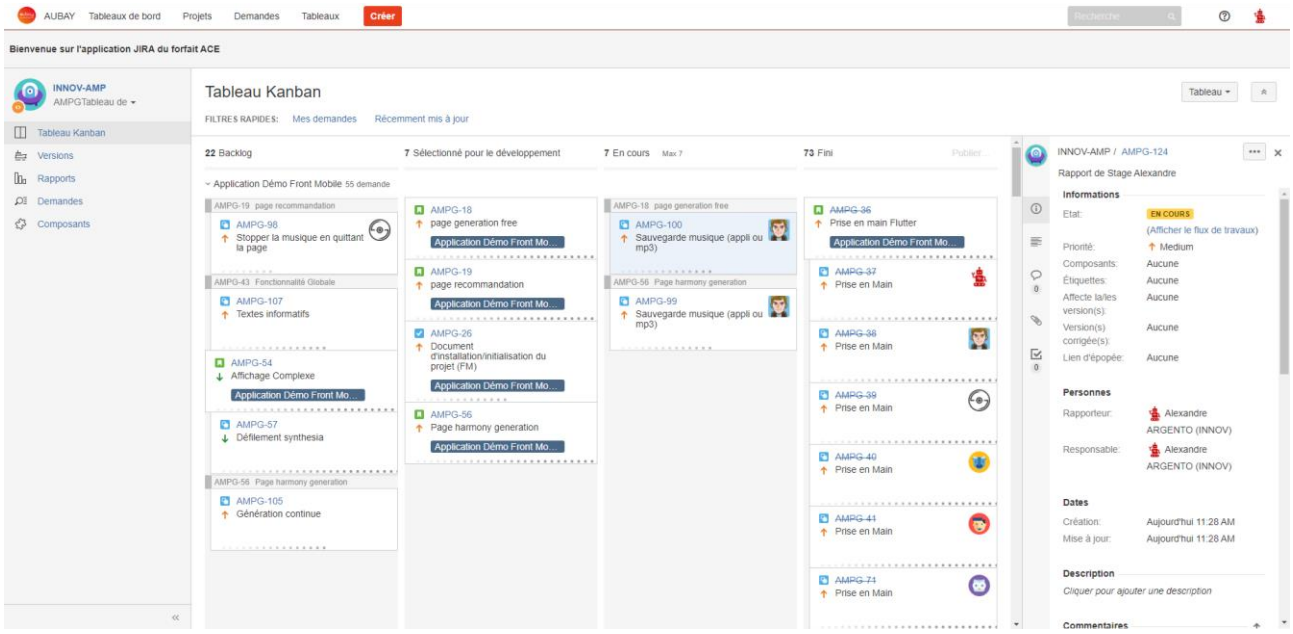

Figure 13 - Vue d'un tableau Kanban du projet AMP

Ce tableau a servi de support pour chacune de nos réunions pour donner un aperçu de notre avancement aux encadrants, et aussi particulièrement pendant la phase de projet, afin de réassigner rapidement des tâches à ceux ayant fini les leurs.

IV.4.b) Développement collaboratif

L'utilisation de Git et le codage en équipe sur un même projet nécessite certaines règles afin de gagner du temps et d'éviter certains problèmes.

Pour les codes des preuves de concept, étant donné que nous avions des parties plutôt indépendantes, les codes de chacun étaient dans une seule branche, séparés dans un dossier comportant le nom de la personne.

Dans le cas du développement de l'application, tout le groupe devait travailler sur des portions de codes présentes dans les mêmes fichiers, ce qui implique l'usage de différentes branches.

La composition générale des branches est la suivante :

- Master (niveau 1) : branche sur laquelle le code est connu pour fonctionner correctement, dans tous les cas de figures possibles.
- Develop (niveau 2) : branche sur laquelle un certain nombre de portions de code fonctionnent individuellement. Elle sert à la sauvegarde du code dans lequel on a regroupé correctement les derniers ajouts de fonctionnalités, mais dont le fonctionnement global doit être testé.

- Partie X (niveau 3) : le nom de cette branche est spécifique au développeur par rapport aux fonctionnalités qu'il doit coder. Cette branche sert à tester, à partir d'un code sans bug et dont personne d'autre ne modifie le contenu, le fonctionnement d'une ou plusieurs fonctionnalités ajoutées par une seule personne.

A partir de cette organisation, les règles sont les suivantes :

1) Chaque développeur n'effectue des modifications que sur sa branche personnelle, de niveau 3.
2) Lorsque tous les développeurs sont prêts et avertis, l'un d'entre eux, si sa branche marche, peut effectuer une fusion de sa branche dans la branche Develop et ainsi lui ajouter les fonctionnalités qu'il a testé.
3) Lorsque tous les développeurs sont prêts et avertis, l'un d'entre eux peut, si possible avec la supervision d'au moins une autre personne, tester si la branche Develop fonctionne dans toutes les configurations possibles. Si la branche fonctionne, la branche Master peut être remplacée par la nouvelle version de Develop.

La règle 1) sert à garantir l'intégrité des branches de niveau 1 ou 2 en minimisant les erreurs dans les branches regroupant du travail partagé. Elle sert également, avec la règle 2) à garantir que les branches de niveau 2 n'ont été modifiées qu'avec la consultation de chacun et donc, que tous sachent si la branche de niveau 2 possèdent ou non du travail qu'ils ont sur leur propre branche.

La règle 2) a pour but principal de cadrer les opérations de fusion de code. Etant donné qu'une fusion dans Develop fait avancer sa version par rapport à toutes les autres branches de niveau 3, il est nécessaire de limiter un maximum ces opérations. De plus, avec la règle 1), elle garantit que chaque développeur sache s'il est à jour avec la branche Develop et lui ajoute des fonctionnalités, ou s'il est en retard de certains ajouts dont il devra faire attention lors d'une fusion future.

La règle 3) permet d'être certain que la version qui est envoyée dans la branche Master est une version stable, sans bug. Ainsi, pendant tout le temps de développement, l'équipe conserve une version de leur travail fonctionnelle. Cette version conservée dans Master peut servir de référence si jamais il a été impossible de débugger la branche Develop et nécessaire de rétrograder la version jusqu'à la dernière qui marche. Elle sert aussi de référence lorsque le développement s'approche d'une date butoir, pour éviter que des ajouts de dernière minute dans le code ne prive l'équipe de toute version stable pour une démonstration.

Ces règles n'ont pas été établies systématiquement au début du projet. En réalité, nous nous sommes petit à petit tous entendus sur leur application pour éviter certains risques ; risques que j'ai eu l'occasion de prendre de façon maladroite en développant pour la première fois avec une telle organisation. Je crois qu'il sera bon, à l'avenir, de convenir formellement d'habitudes de développement dès le départ, pour éviter tout malentendu.

V. Conception et Analyses système

V.1. Approches en composition musicale

Lors de mon arrivée sur le stage, l'objectif était d'identifier de quelle manière, autre que le réseau de neurones, il était possible de créer de la musique. J'ai donc représenté, d'après mon expérience, les différentes approches possibles de traiter lorsqu'un compositeur crée de la musique. Par cette première réflexion Figure 14, il était possible de représenter les aspects déjà en cours de traitement par Louis et Arthur et de trouver où serait mon apport.

La partie « temporelle » se concentre sur la progression des notes constituant une mélodie ou un accompagnement. Cette progression possède trois caractéristiques : la mélodie elle-même et ce qui la caractérise, le son de cette mélodie et sa structure en termes de répétitions.
La partie « instantanée » est orientée sur l'étude, à un instant précis, de tous les sons qui se superposent et qui sont perçus en même temps. Il s'agit alors d'étudier soit la consonnance des notes jouées en même temps, soit le mariage que représente l'ensemble des instruments, sons et percussions utilisés.

Plus l'on descend dans l'arbre, plus les termes utilisés font référence à des notions techniques, spécifiques. Les pictogrammes présents ont pour objectif d'imager cette décomposition pour améliorer sa compréhension. A cela, nous ajoutons des encadrés, dont la légende figure en haut à gauche, afin de préciser le rôle du travail des membres de l'équipe dans le projet.

L'ensemble des points terminaux de ce diagramme constitue représente des points d'intérêt dans la composition de musique. Cependant, il n'est pas nécessaire de tous les traiter. L'importance est d'aborder des points de vue complémentaires afin d'avoir un produit fini intéressant musicalement. Par exemple, combiner approche temporelle et instantanée (au moins la consonnance) permet d'avoir une musique construite dans le temps tout en ayant, à chaque instant, une certaine qualité d'écoute.

En identifiant où se positionne le projet dans le réglage, ou l'apprentissage de caractéristiques musicales, certaines caractéristiques peuvent être fixées arbitrairement de façon à garder un modèle de composition simple. L'ordre dans lequel chacune des caractéristiques doit être prise en compte dans la composition n'est pas précisé ici et constitue un autre sujet de discussion. En effet, le compositeur peut partir d'une mélodie qu'il a en tête avant d'y ajouter un accompagnement, une rythmique, etc…, ou alors, le point de départ peut être une progression d'accord donnant lieu à une introduction, puis un couplet, etc…

Pour finir, cette représentation est basée sur une séparation simpliste entre les points de vue purement temporels et fréquentiels. Elle présente la limite de mal décrire des notions combinant approches horizontales et verticales. De même, elle n'est pas forcément exhaustive dans la décomposition qu'elle propose.

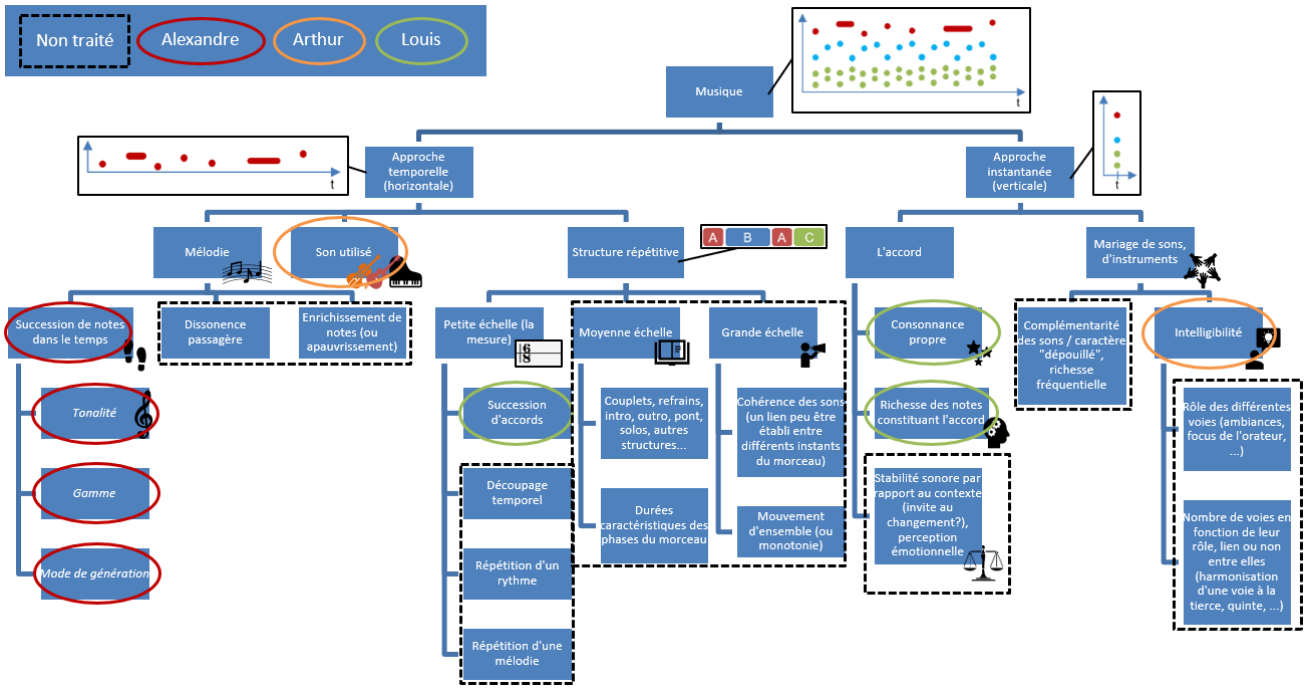

Figure 14 - Diagramme d'approches en composition musicale

V.2. Légitimité d'une génération créative

Comment légitimer une démarche de production musicale ? Cette question devient cruciale lorsque l'on est amené à faire des choix qui peuvent biaiser la création de musique. Le but n'est pas de demander à des compositeurs d'enregistrer de la musique selon leurs propres critères. Il est nécessaire de trouver des moyens de prendre en compte des données extérieures pour donner un résultat cohérent avec des musiques déjà existantes.

L'aléatoire est visiblement la seule traduction du caractère créatif des algorithmes (bien qu'il ne soit qu'apparent pour les ordinateurs non quantiques). Cet élément apparaît toujours dans un système créatif. Pour créer ce système, nous modélisons toute la chaîne de traitement de la donnée, puis nous utilisons l'aléatoire en tant qu'entrée du système en espérant avoir un résultat à la fois nouveau, mais conforme au format de données que nous avons voulu lui donner.

Deux options s'offrent à nous. Soit, nous utilisons des données existantes pour les « apprendre » et ainsi créer quelque chose qui s'en « inspire » (flèche en haut à gauche de la Figure 15). Soit, nous modélisons mathématiquement l'idée que nous avons du type de création désiré, tout en laissant au modèle la place à l'aléatoire (flèche en bas à gauche de la Figure 15). Le résultat (la création du système en question) sera alors contenu dans un ensemble respectant des exigences fixées comme axiomes, dont leur définition concrète est à débattre.

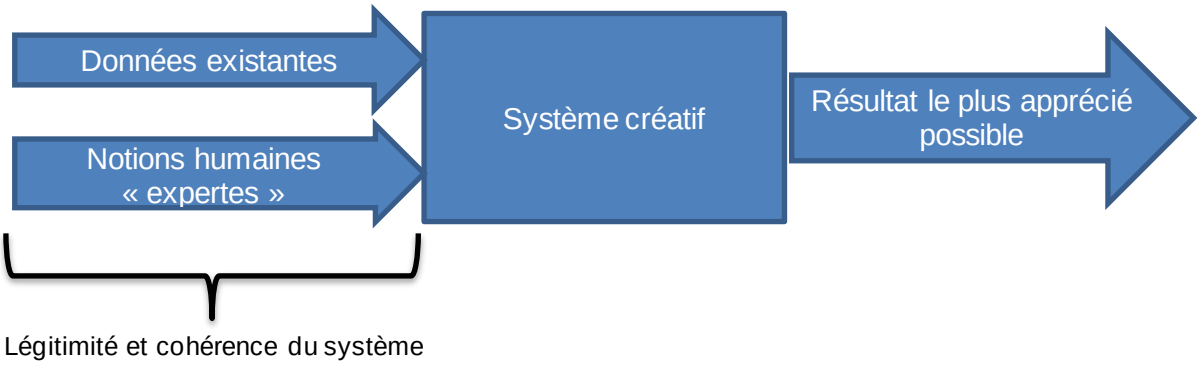

Figure 15 - Première représentation d'un système créatif

A cela, nous y ajoutons les hypothèses simplificatrices de travail visant à limiter la complexité du cadre dans lequel nous travaillons, pour le rendre compréhensible par un ordinateur et par des personnes.

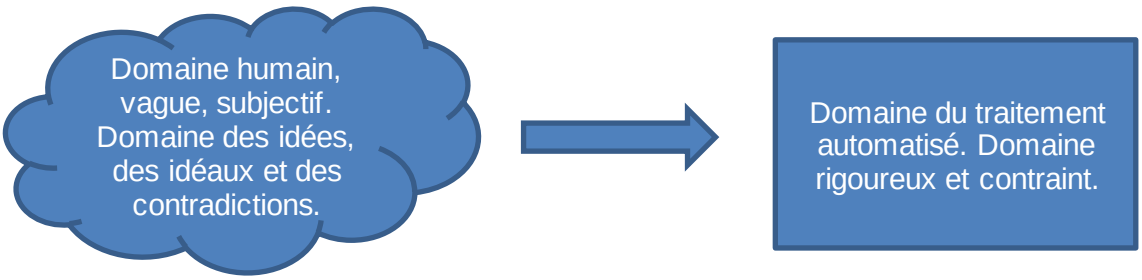

Figure 16 - Paradoxe de l'automatisation

L'application du Machine Learning avec les réseaux de neurones reprend la première option, mais inclue partiellement la deuxième dans le choix du modèle de réseau de neurones, ceci de façon plus ou moins lié à des critères humains (mémoire, régression des données apprises, …).

L'intérêt de la modélisation est d'orienter le débat sur la façon dont on a traduit mathématiquement des notions humaines (consonnance des notes, perception positive des répétitions), tandis que l'apprentissage par réseaux de neurones nous permet de relayer la modélisation technique à la phase d'entraînement. Ainsi, le Machine Learning permet de justifier une partie du modèle conçu (les poids des liens du réseau de neurones et des matrices de convolution) par la phase d'apprentissage avec des données existantes.

Enfin, une dernière façon de justifier une modélisation est de tirer parti des retours de l'utilisateur en fonctionnement (flèche en bas à droite de la Figure 17). C'est notamment le cas de l'apprentissage par renforcement qui met en place un retour sur l'appréciation du résultat produit. Cette appréciation sera prise en compte pour améliorer la production suivante. Le système tirera donc de l'expérience de la qualité de ses productions.

L'apprentissage par algorithme génétique met aussi en œuvre l'idée de « retour », mais le nombre d'élément à examiner étant généralement trop grand, la définition d'un « critère de sélection » est nécessaire pour automatiser cette tâche. Cette définition doit modéliser notre approbation ou non à l'idée de garder certaines productions pour la génération suivante.

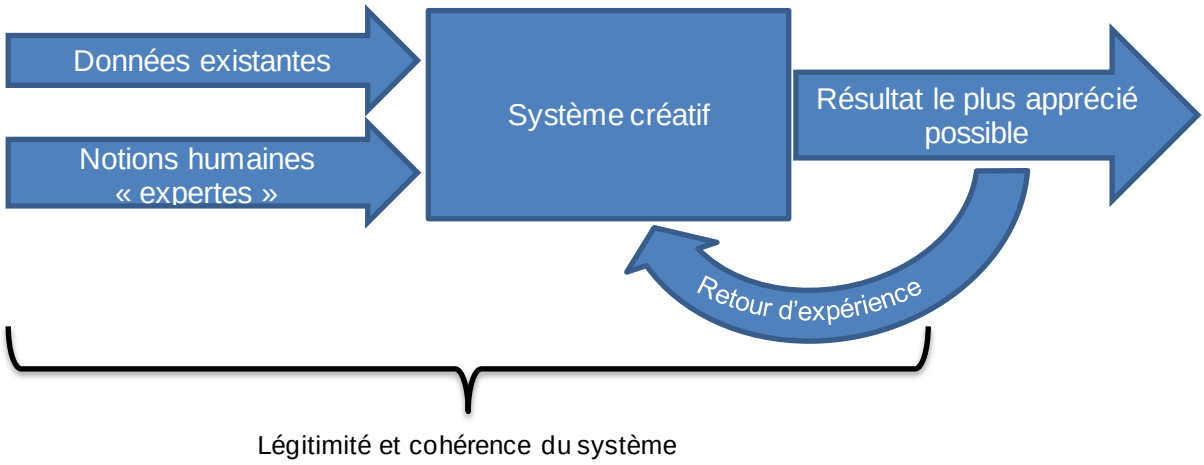

Figure 17 - Représentation d'un système créatif rendu légitime

V.3. Créativité et interface utilisateur

Alors que certains choix de modélisation de la musique peuvent être biaisés et justifiés uniquement par des besoins de simplification ou de budget, certains choix peuvent être laissés réglables en permanence pour adapter la génération au client concerné.
De cette manière, soit c'est le développeur qui effectue les réglages par rapport au client (colonne de gauche de la Figure 18), soit c'est le client lui-même qui a la possibilité d'influencer en sa faveur la musique générée (colonne de droite de la Figure 18).

La problématique qui apparaît autour de cela est le niveau de complexité du réglage proposé. En effet, il est nécessaire que le client puisse comprendre comment se servir des réglages proposés afin d'obtenir la musique qu'il désire.
La même problématique se pose d'un point de vue des développeurs du générateur de musique : soit le code n'a besoin d'aucun réglage à adapter en fonction du client (les données qui récupère lui suffisent pour être pertinent), soit il est nécessaire de régler certains paramètres et donc de connaître leur fonction.

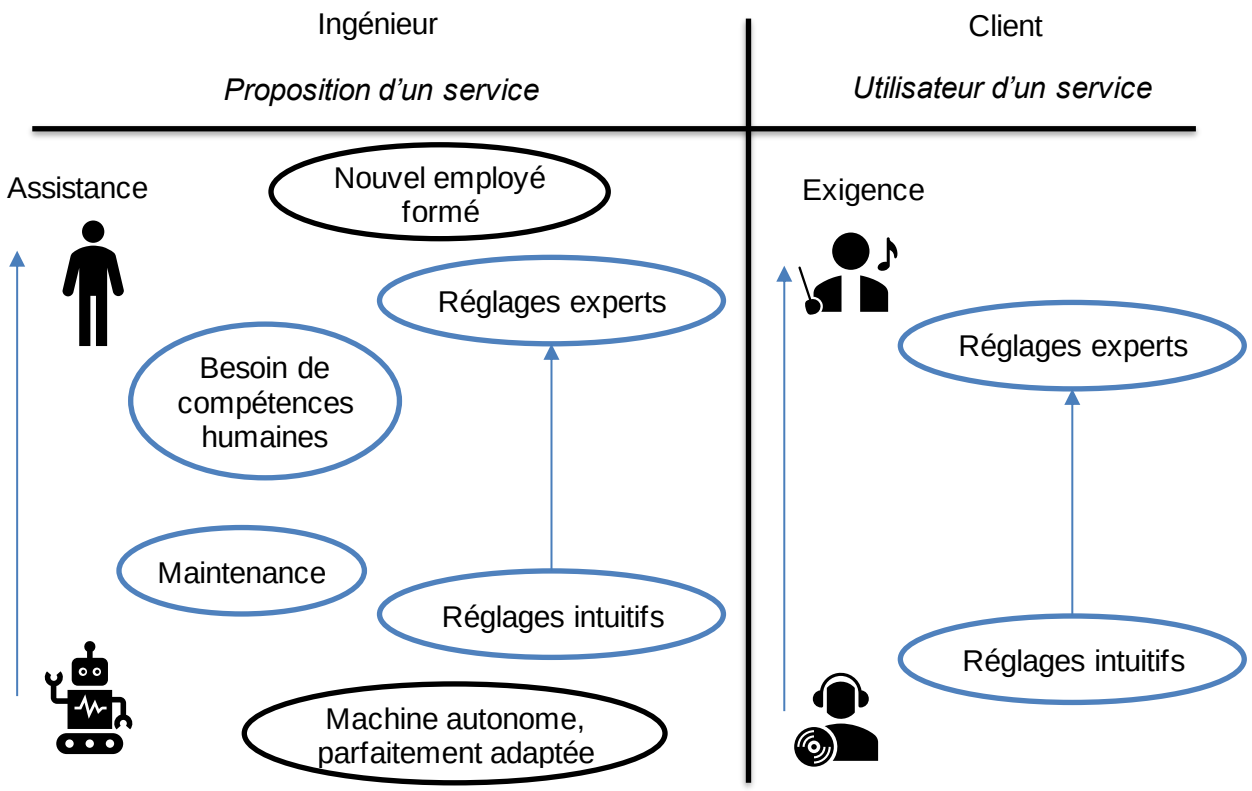

Figure 18 - Représentation des niveaux de réglages

D'un point de vue de l'utilisateur du générateur de musique, l'aspect subjectif et changeant de son besoin implique une zone de battement forcément non vide entre le produit qui lui sera livré et ses attentes. L'idéal étant de livrer une « boîte noire » dont l'usage est intuitif et le résultat est celui attendu par le client, la présence systématique de réglages dans le produit proposé demande une certaine mise à niveau. La Figure 19 montre en bleu la partie dont s'occupe l'ingénieur, faisant partie de ce que le client ne sait pas ou ne veut pas faire, en vert la partie qu'il sait déjà faire et en orange, la partie qu'il peut faire.

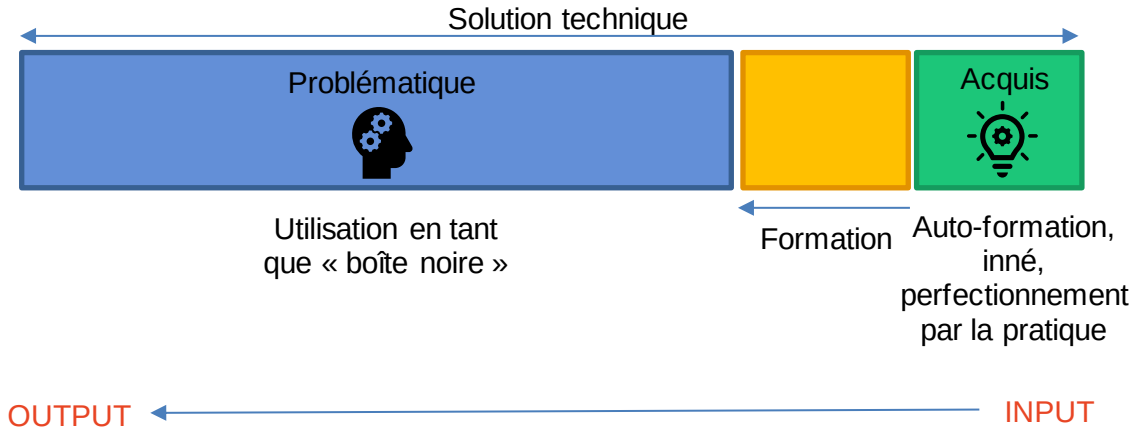

Figure 19 - Représentation de l'écart entre le besoin du client (input) et sa solution (output)

Dans le cas d'un générateur de musique, plus les concepts derrières les réglages proposés au client sont simples à comprendre, plus la zone orange sera petite. Toutefois, il ne faut pas oublier d'identifier le type de client, s'il s'agit d'un expert en musique qui veut des réglages précis ou s'il s'agit d'une personne sans expérience, qui ne veut pas avoir à se casser la tête.

Pour aller plus loin, la Figure 20 propose une décomposition non du produit qu'est le générateur, mais de la création musicale qui en résulte. Cette création possède d'abord deux aspects antagonistes :

- La modélisation fixe, qui ne changera jamais d'une génération à une autre (voir Figure 21).
- L'espace libre, partie fonction de deux facteurs :
  - Les choix utilisateurs, sélection de données d'apprentissage de référence (style de musique, …), sélections de poids stockés dans des matrices de Markov ou paramétrage par rapport à des fonctionnalités proposées.
  - La partie purement « créative » de la production, qui peut changer pour des mêmes choix utilisateurs et qui se base sur de l'aléatoire.

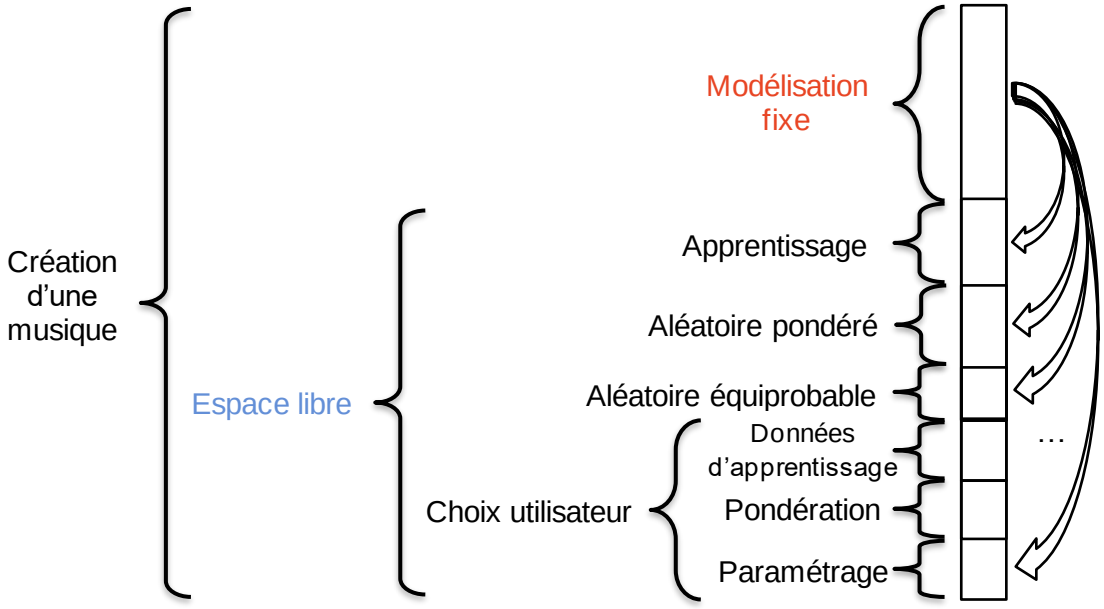

Figure 20 - Modélisations, créativité et choix

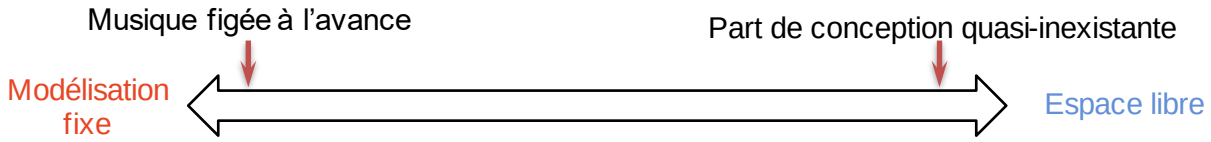

Figure 21 - Espace libre et modélisation fixe

VI. Preuves de concept

VI.1. Modélisations successives avec Sonic Pi

VI.1.a) Développement

L'objectif de cette partie est de vérifier l'intérêt d'implémenter un générateur de mélodies basé uniquement sur des modèles mathématiques et de l'aléatoire.
L'aléatoire (ou plus exactement, le pseudo-aléatoire pour un ordinateur classique) est la base de la partie créative du générateur puisqu'il s'agit du processus non déterministe permettant d'avoir un aspect imprévisible dans l'évolution de la mélodie.

Afin de contrôler un minimum la génération de mélodie et de la faire respecter certains critères en accord avec notre conception d'une mélodie, nous ajoutons des modélisations visant à conditionner ce choix aléatoire.
Trois modélisations ont été implémentées :

- **Restriction de notes** : L'accompagnement joué pour ce test est composé d'accords basés sur la tonalité C majeur. Cela implique que certaines notes vont généralement être en dissonance avec l'accompagnement. Le modèle ci-dessous montre le niveau de dissonance de la note, du moins dissonant en vert foncé, au plus dissonant en rouge, en passant par le vert clair, le jaune, puis l'orange :

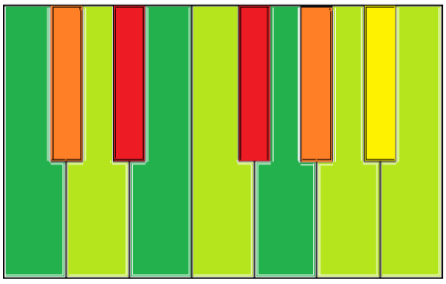

Figure 22 - Niveaux de consonnances des notes avec C majeur

  Ce modèle nous permet de restreindre notre choix de notes mélodiques afin d'avoir toujours un certain niveau de consonance. Les notes que l'on retiendra (celles en vert foncé et vert clair) seront en fait ce que l'on appelle communément une gamme. Cette gamme s'appelle ici la gamme de C majeur en raison de sa position d'origine, la note C, et du mode que favorise l'accompagnement, le mode majeur. Nous opérons alors à une restriction du choix de notes à seulement 7 notes indiquées en vert (C, D, E, F, G, A et B).

- **Pondération** : D'après le clavier coloré représenté ci-dessus, certaines notes ont une plus grande importance. D'un point de vue de la consonance, les notes en vert foncé sont celles qui sont les plus stables, qui ont un rôle plus important dans la gamme, tandis que les notes en vert clair sont des notes de passage. Il est alors possible d'améliorer la génération de mélodie en augmentant la probabilité de choisir une note vert foncé par rapport à une note vert clair. Cela se traduira par la pondération du choix des notes, déduite directement du modèle de consonance défini dans le code (en tant que liste de nombres entiers).

- **Limitation mélodique** : Les restrictions et les pondérations que l'on peut faire sur les notes ne relèvent que de l'aspect de consonance générale de la mélodie, il nous faut aussi prendre en compte une notion de cohérence dans le déplacement des notes. Cette notion de cohérence peut se traduire par une certaine proximité entre des notes successives, mais aussi par une notion de répétitions que nous ne traiterons pas ici, par souci de simplicité. La limitation mélodique a donc pour fonction de réduire la portée du choix des notes autour de la note précédemment jouée. On définira cette portée par un entier "delta"

(portée mélodique) dans le programme Sonic Pi. Par exemple, si la note E vient d'être jouée et que delta vaut 2, alors nous avons le choix parmi les notes suivantes: C, D, E, F ou G (en prenant en compte la modélisation de restriction de notes).

Notons que seul l'ordre d'application de la limitation mélodique et de la restriction de notes n'a d'importance du point de vue du résultat :

- Pour la pondération et la restriction, le fait de restreindre le choix des notes ne modifie par le poids de ces notes car il est indépendant de leur nombre. De même, la restriction de notes est effectuée par rapport aux noms des notes et non par rapport à leur poids.
- Pour la pondération et la limitation mélodique, la même propriété s'applique car la limitation mélodique est une restriction particulière. Le fait que cette restriction soit dépendante de la note en cours n'a pas d'influence sur la pondération qui elle est intemporelle.
- Pour la restriction de notes et la limitation mélodique, la restriction change la portée de la limitation, et ce de façon (généralement) asymétrique autour de la note considérée, ce qui empêche (généralement) l'équivalence avec le choix de delta.

Pour résumer, si nous considérons que nous appliquons les modélisations de gauche à droite, nous avons (propriété de commutativité) :

- Pondération + Restriction = Restriction + Pondération
- Pondération + Limitation mélodique = Limitation mélodique + Pondération
- Limitation mélodique + Restriction ≠ Restriction + Limitation mélodique

Ce dernier point doit donc être résolu de façon arbitraire. Nous choisirons ici de ne pas compter les notes que nous avons supprimé dans le réglage de delta (la portée mélodique). De plus, nous ne souhaitons pas qu'il y ait d'asymétrie autour de la note en cours lors du choix de la prochaine note (autant de notes plus basses que plus autres). Cette asymétrie pourra faire l'objet d'une modélisation complémentaire, nous permettant d'avoir un vrai contrôle sur cette asymétrie.
Lorsqu'il y a commutativité, nous choisirons sans hésiter la solution la moins coûteuse en temps de traitement.

VI.1.b) Analyse et conclusions

D'après les résultats audios obtenus, il est possible de constater la valeur ajoutée de chacune des modélisations utilisées, malgré la dépendance par rapport à l'auditoire :

- **Restriction de notes**: Grande valeur ajoutée. Plus aucune note ne dissone, l'oreille n'est plus "agressée" à l'écoute du résultat.
- **Limitation mélodique**: Grande valeur ajoutée. Sensible structuration de la mélodie donnant une sensation de cohérence.
- **Pondération**: Valeur ajoutée plus difficile à percevoir, généralement moins importante.

D'un point de vue orienté sur la modélisation, nous pouvons constater que la charge du travail d'analyse grandit de façon exponentielle avec le nombre de modélisations choisies. En effet, pour n modélisations, il faudra considérer tous les arrangements possibles sans répétition de ces dernières, nous menant à "k parmi n" audios à générer et à analyser, pour k allant de 1 à n.

$$\sum_{k=1}^{n}\binom{n}{k} = \sum_{k=0}^{n}\binom{n}{k} * 1^k * 1^{n-k} - \binom{n}{0} = (1+1)^n - 1 = 2^n - 1$$

Équation 1 - Nombre d'arrangements sans répétition en fonction du nombre de modélisations

Ces arrangements sont donc au nombre de 2^n - 1 (la simplification de la somme a été faite en utilisant la formule du binôme de Newton). Certains choix, comme nous l'avons vu, nous permettent de réduire ce nombre, comme le choix de fixer des ordres de modélisations, mais il ne peut pas être fait systématiquement sans justifications préalables.

Il est donc difficile de s'aventurer dans le chemin des superpositions de couches de modélisations étant donné le nombre de points de discussions qui se profilent et qui peuvent tous faire débat. Par exemple, l'usage d'un autoencodeur (Deep Learning) représente une solution, il permet de s'affranchir de ce niveau de modélisation car elle est partie intégrante du processus d'abstraction dans ce qui est appelé l'espace latent. Toutefois, cette nouvelle modélisation implique le réglage de nouveaux paramètres dont l'influence sur le résultat est moins évidente.

Etant donné l'objectif de notre projet portant principalement sur l'usage de connaissances (modélisations) afin de mieux structurer la création musicale, l'objectif qui se profile est de trouver un juste milieu entre l'assimilation des données existantes (Deep Learning) et la définition de règles.

VI.2. Génération de mélodies par matrice de Markov

VI.2.a) Approche préliminaire

Contrairement au Deep Learning qui consiste à apprendre à générer des données en bloc, sans avoir de contrôle particulier sur les différentes caractéristiques apprises au sein du réseau de neurone, l'utilisation de matrices de Markov concentre l'apprentissage sur une caractéristique précise. Pour Louis, il s'agit des progressions d'accord et ici, des progressions de notes constituant une mélodie.

Le désavantage de cette méthode est de ne permettre un apprentissage ciblé sur une seule caractéristique et non toutes en même temps. La Figure 23, tirée de mon document de POC en anglais pour Aubay, montre le travail avec du Deep Learning en haut et le travail sur des caractéristiques spécifiques en bas. Il est facile à deviner sur cette figure que les générations ne seront pas équivalentes puisque je ne traiterai que du générateur de mélodie (partie encadrée en rouge). Heureusement, l'ajout du travail de Louis et d'Arthur sur le générateur d'accord permet de générer un accompagnement donnant plus de consistance à la musique.

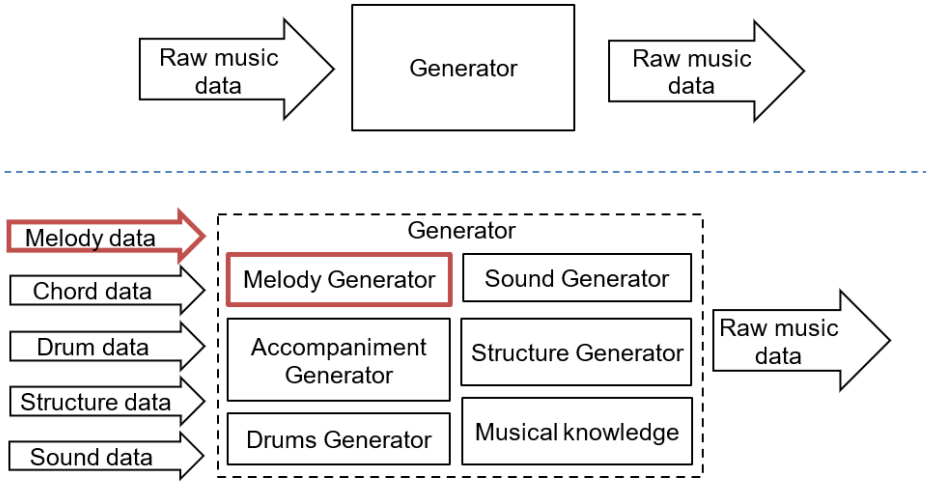

Figure 23 - Différences entre l'usage du Deep Learning et l'apprentissage ciblé

De plus, alors qu'il est aussi possible d'utiliser le Deep Learning pour générer des mélodies, je me suis d'abord dirigé sur les matrices de Markov pour diversifier les approches du groupe AMP. Les matrices de Markov présentent aussi l'avantage de générer des données de tailles quelconques.

VI.2.b) Principe

Si nous ne considérons que deux notes différentes (A et B) pour la génération, il s'agit de considérer le graphe d'état de la Figure 24 avec les poids $P_X(Y)$, représentant la probabilité d'aller sur Y sachant que l'on se trouve sur X.

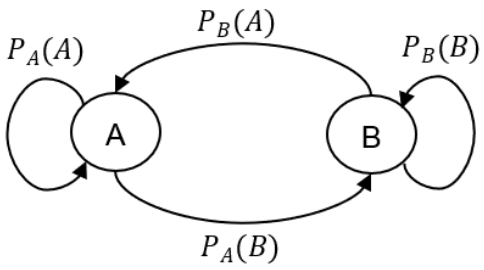

Figure 24 - Graphe d'états élémentaire

L'ensemble des poids de ce graphe peuvent être disposés dans une matrice que l'on appelle Matrice de Markov :

$$M = \begin{pmatrix} P_A(A) & P_B(A) \\ P_A(B) & P_B(B) \end{pmatrix}$$

Équation 2 - Expression d'une matrice de Markov

D'autre part, il est possible d'obtenir les valeurs des probabilités à partir d'une matrice qu'on appellera matrice de recensement :

$$M^* = \begin{pmatrix} N(A \to A) & N(B \to A) \\ N(A \to B) & N(B \to B) \end{pmatrix}$$

Équation 3 - Expression d'une matrice de recensement

Les valeurs $N(X \to Y)$ représentent le nombre de cas où X est suivi de Y dans le dataset considéré. Ainsi, nous pouvons calculer la matrice de Markov avec la formule suivante, sous réserve qu'aucun dénominateur ne s'annule :

$$M = \begin{pmatrix} \dfrac{N(A \to A)}{N(A \to A) + N(A \to B)} & \dfrac{N(B \to A)}{N(B \to A) + N(B \to B)} \\ \dfrac{N(A \to B)}{N(A \to A) + N(A \to B)} & \dfrac{N(B \to B)}{N(B \to A) + N(B \to B)} \end{pmatrix}$$

Équation 4 - Lien entre matrice de Markov et matrice de recensement

L'annulation d'un dénominateur est équivalent au fait d'avoir une colonne complète de zéros dans la matrice de recensement. Un tel cas arrive lorsqu'un état du graphe n'est jamais rencontré en tant qu'entrée dans le dataset. Il est alors impossible de faire un choix quand au passage vers un autre état lorsque l'on est dans celui-ci.

En plus du calcul de la matrice de Markov servant de transition entre les notes, il est possible de calculer le vecteur d'état initial :

$$I = \begin{pmatrix} \dfrac{N(A)}{N(A) + N(B)} \\ \dfrac{N(B)}{N(A) + N(B)} \end{pmatrix}$$

Équation 5 - Equation du vecteur d'état initial

Avec N(X) représentant le nombre de fois où nous commençons par X dans le dataset.

Le vecteur I donne ainsi les probabilités de commencer par l'état A ou B en regardant la colonne correspondante.

Avec les informations I et M, nous avons donc la possibilité de générer des notes selon les étapes suivantes :

1. Choisir un état initial A ou B selon les probabilités dans I. Cet état devient l'état courant.
2. Si l'état courant est A, choisir l'état suivant en fonction des probabilités données par la première colonne de la matrice M. Si l'état courant est B, choisir l'état suivant en fonction des probabilités données par la deuxième colonne de la matrice M.
3. L'état suivant devient l'état courant.
4. Retourner à l'étape 2.

Dans notre cas, nous utiliserons des définitions similaires, mais généralisées, car nous travaillons avec plus de deux états : A et B peuvent être remplacés par n'importe quelle note MIDI entre 0 et 128. Si n est le nombre d'états possibles, alors la matrice de Markov sera de taille (n,n).

Notons que nous n'avons, à aucun moment pris en compte le temps entre deux notes successives. Il sera constant entre chacune des notes afin de simplifier notre étude.

La même méthode peut être appliquée en prenant en compte un nombre quelconque, μ (la mémoire), de notes en entrée dans le dataset, ce qui donne une matrice de Markov de taille (n,n^{μ}) en prenant en compte toutes les permutations de notes possibles dans leur ordre d'apparition.

VI.2.c) Normalisation tonale

Etant donnée qu'il faut pouvoir adapter la génération de mélodie à l'accord généré, nous devons nous placer dans une gamme par défaut afin d'effectuer une transformation pour passer dans une nouvelle gamme.
Comme dans la partie V1.1.a), nous nous restreindront les notes à celles de la gamme de C majeur (voir Figure 25). Cela implique un prétraitement des données d'apprentissage afin de détecter leur gamme, puis de les ramener dans la gamme de C majeur. Nous appellerons cette étape, la normalisation tonale.

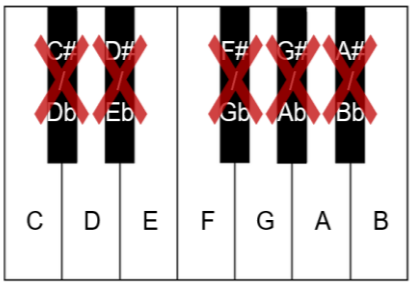

Figure 25 - Notes de la gamme C majeur

Par exemple, en s'assurant de ne travailler qu'avec des gammes à 7 notes, nous pouvons passer de la gamme de C mineur (voir Figure 26) à la gamme de C majeur en conservant l'ordre des notes et en changeant les notes D# (ou Eb) en E.

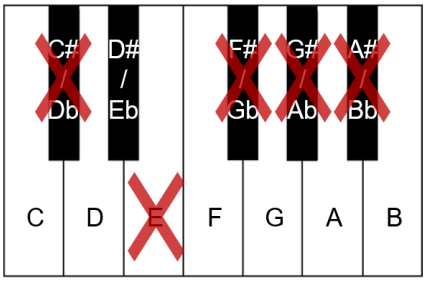

Figure 26 - Notes de la gamme C mineur

VI.2.d) Dataset

Le dataset utilisé pour la génération de mélodies est, comme indiqué dans l'état de l'art, le Weimar Jazz Dataset. Voici une représentation des notes d'une de ses 456 solos de jazz :

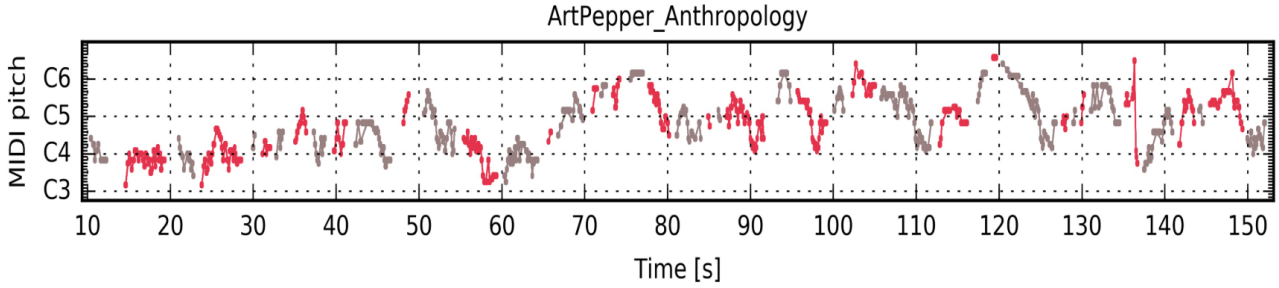

Figure 27 - Représentation des notes du solo de Art Pepper - Anthropology, en fonction du temps

Afin d'avoir une meilleure consistance dans la succession des notes et à cause du fait que nous négligeons le rythme, les solos ont été découpés en fonction de leurs plus grandes pauses. Cette opération a consisté à détecter des pauses « outliers » au-dessus d'un écart-type de la régression linéaire, comme celle qui apparaît Figure 28.

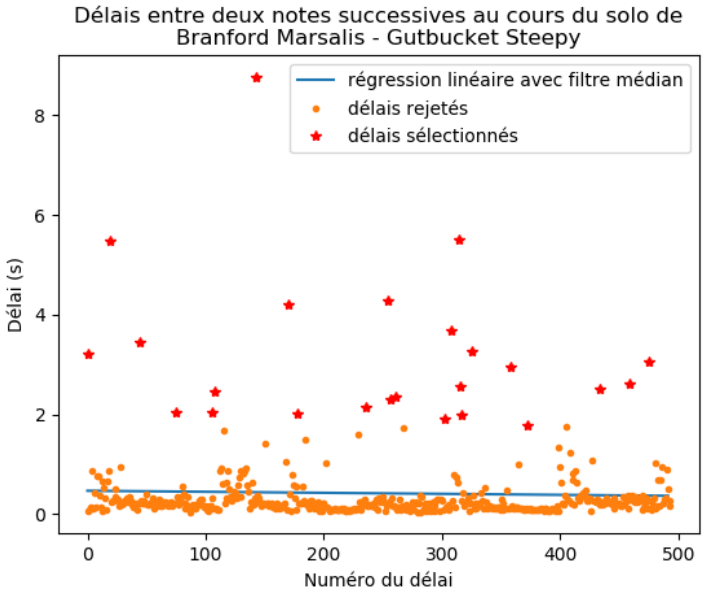

Figure 28 - Détection d'outliers pour le découpage de solo

Les groupes de notes successives obtenus sont appelés des « phrasés musicaux ».
La deuxième étape du traitement est alors de normaliser les phrasés musicaux en recalant les notes hors de leur gamme, puis en transformant la gamme vers celle de C majeur.

Les annotations de gamme étant très inexactes par rapport aux notes des solo correspondant, il a fallu coder un détecteur de gamme basé sur la maximisation du nombre de notes qui rentrent dans la gamme.

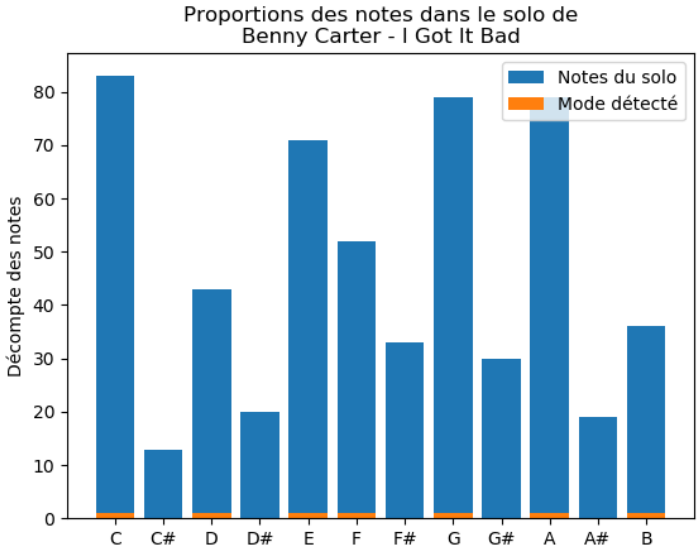

Figure 29 - Affichage de la gamme détectée par rapport aux notes d'un solo

Précisons ici que toutes les gammes testées sont des gammes a 7 notes couramment présente en musique.

VI.2.e) Résultats

L'apprentissage de la matrice de Markov pour une mémoire d'une note Figure 30 possède trois propriétés remarquables :

- **Emphase sur le mouvement** : la diagonale centrale où l'état « note i » donne un état « note i » apparaît plus claire que ses diagonales voisines, ce qui montre que les chances de générer une future note différente est plus grande que celle de rester sur la même note.
- **Limitation de l'amplitude de mouvement mélodique** : présence de valeurs négligeables, voire nulles à partir des quatrièmes diagonales, supérieures comme inférieures.
- **Contrebalance du mouvement des notes extrêmes** : cela est caractérisé par des plus grandes probabilités en bas de la diagonale supérieure n°2, tout comme des plus grandes probabilités en haute de la diagonale inférieure n°2. Cela peut être vu comme un « effet de cuvette » sur l'échelle des notes MIDI.

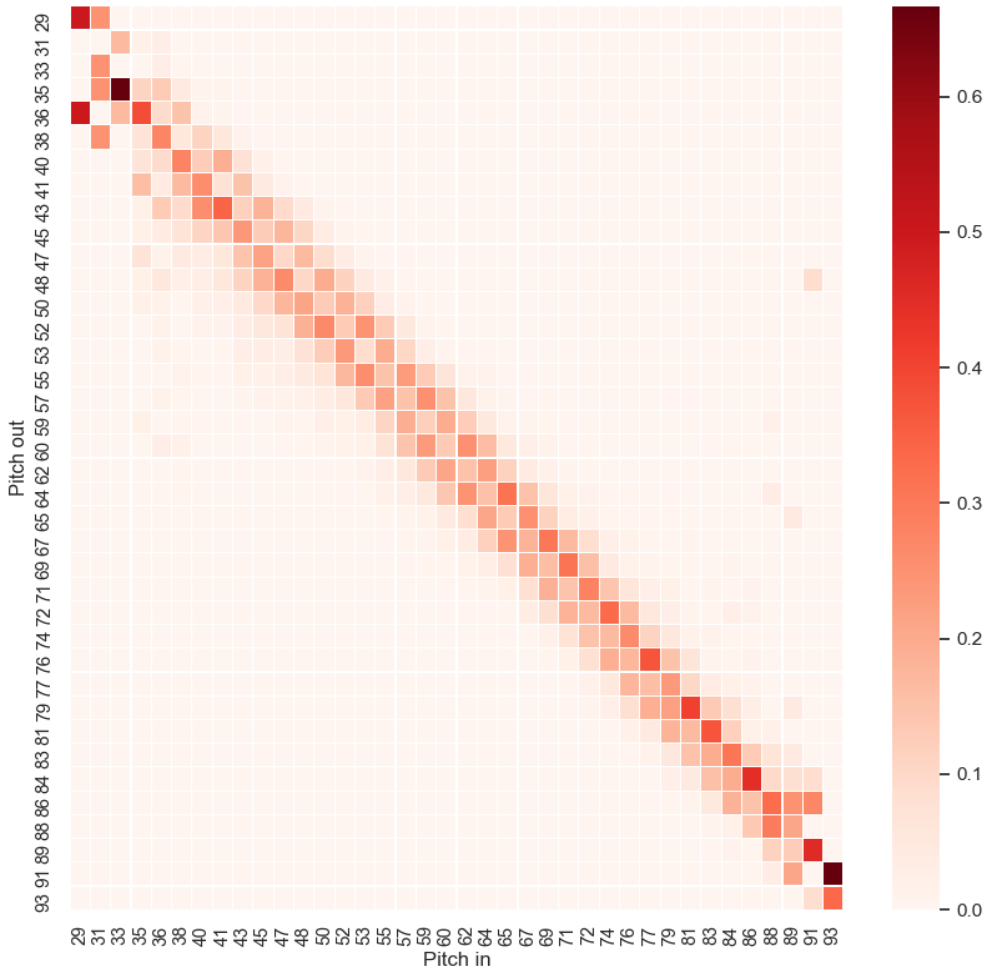

Figure 30 - Matrice de Markov pour une mémoire µ=1

Des résultats moins conclusifs sont tirés de l'apprentissage pour µ>2, d'une part car il devient compliqué d'afficher matrice entière et d'autre part car l'ajout d'information apprise ressemble à la fois peu discernable et peu interprétable.

Du fait que les informations manquantes dans les colonnes de la matrice de Markov aient été remplacées par celles avec un µ inférieur, le phénomène de diagonales que pour µ=1 réapparaît de manière générale (voir Figure 31), comme si l'on n'avait retenu que la dernière note dans les entrées de la matrice. Il est cependant possible de remarquer que quelques fortes valeurs apparaissent hors des diagonales principales.

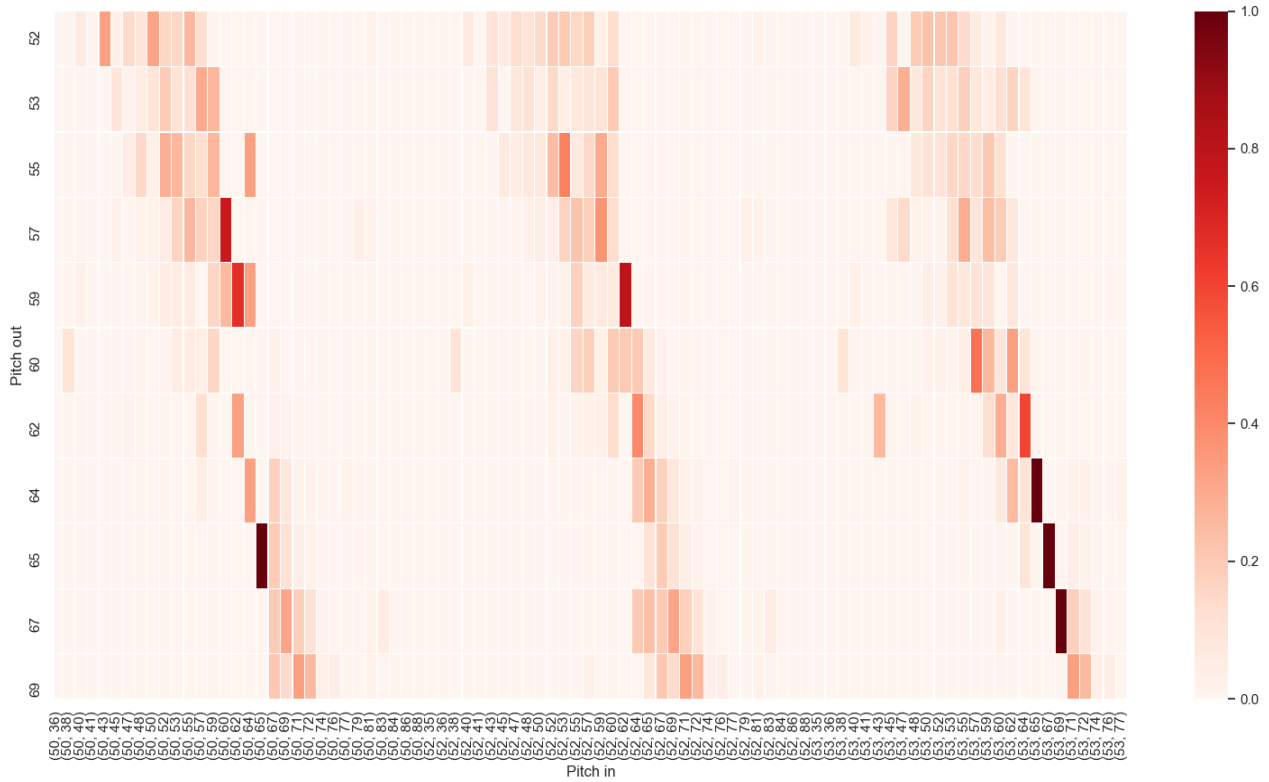

Figure 31 – zoom sur la matrice de Markov pour µ=2

Mes tests ne se sont pas étendus au-delà de µ=3 à cause de la limite de stockage. Nous appellerons désormais une unité générant des notes avec une matrice de Markov, un générateur de Markov.

VI.3. Génération d'accords par générateurs de Markov

Une autre particularité des matrices de Markov est de pouvoir, malgré l'apprentissage qui a été fait, bloquer la possibilité de tirer certaines notes en les supprimant du tirage aléatoire pondéré.
Avec cette technique, il est possible de former des accords à partir de trois générateurs.

Les critères choisis pour contraindre les générateurs sont déterminés en cascade. Autrement dit, il faut qu'un générateur génère une note pour savoir comment contraindre le générateur suivant avant qu'il ne génère une note à son tour, et ainsi de suite.
Nos trois générateurs ont donc un ordre, et plus le rang du générateur est élevé, plus il sera contraint:

- Premier générateur :
  - Génère un pitch de "basse" entre 48 et 59 (choix d'une note sur une octave).
  - Génère : pitch1.
- Deuxième générateur :
  - Doit choisir un pitch sur l'octave de pitch1, c'est-à-dire entre pitch1 et pitch1 + 12.
  - Ne doit pas choisir pitch1.
  - Ne soit pas choisir un pitch trop près de pitch1, soit ni pitch1 – 1 ni pitch1 + 1.
  - Génère : pitch2.
- Troisième générateur :
  - Doit choisir un pitch sur l'octave de pitch1, c'est-à-dire entre pitch1 et pitch1 + 12
  - Ne doit pas choisir pitch1.
  - Ne soit pas choisir un pitch trop près de pitch1, soit ni pitch1 – 1 ni pitch1 + 1.
  - Ne doit pas choisir pitch2.
  - Ne soit pas choisir un pitch trop près de pitch2, soit ni pitch2 – 1 ni pitch2 + 1.

Ici, le nombre de notes choisies est alors fixé à trois, bien qu'il soit aussi possible d'obtenir des accords intéressants avec deux ou quatre notes. Cette extension pourra être implémentée avec la notion de rythme intégrée au générateur ou la notion de sur-contrainte.

VI.4. Génération de mélodies par réseaux de neurones (inachevé)

Un certain temps entre la fin du POC sur la génération par matrices de Markov et le développement du démonstrateur mobile a été consacré à revenir sur les réseaux de neurones pour générer des mélodies.

L'idée principale est d'entraîner un autoencodeur variationnel (VAE pour Variational Auto-Encoder, réseau de neurone symétrique, encodeur puis décodeur) sur des mélodies d'une mesure et d'utiliser la génération de quelques mesures pour créer une musique avec une structure répétitive.

Etant donné la difficulté qu'il y avait à analyser les capacités des VAE entraînés, notamment pour noter les différences entre deux valeurs proches de l'espace latent (espace entre l'encodeur et le décodeur), j'ai passé un certain temps à développer une interface pour régler l'espace latent, visualiser et écouter directement le résultat.

En voici un exemple visuel pour un espace latent à deux dimensions (deux curseurs réglables) Figure 32.

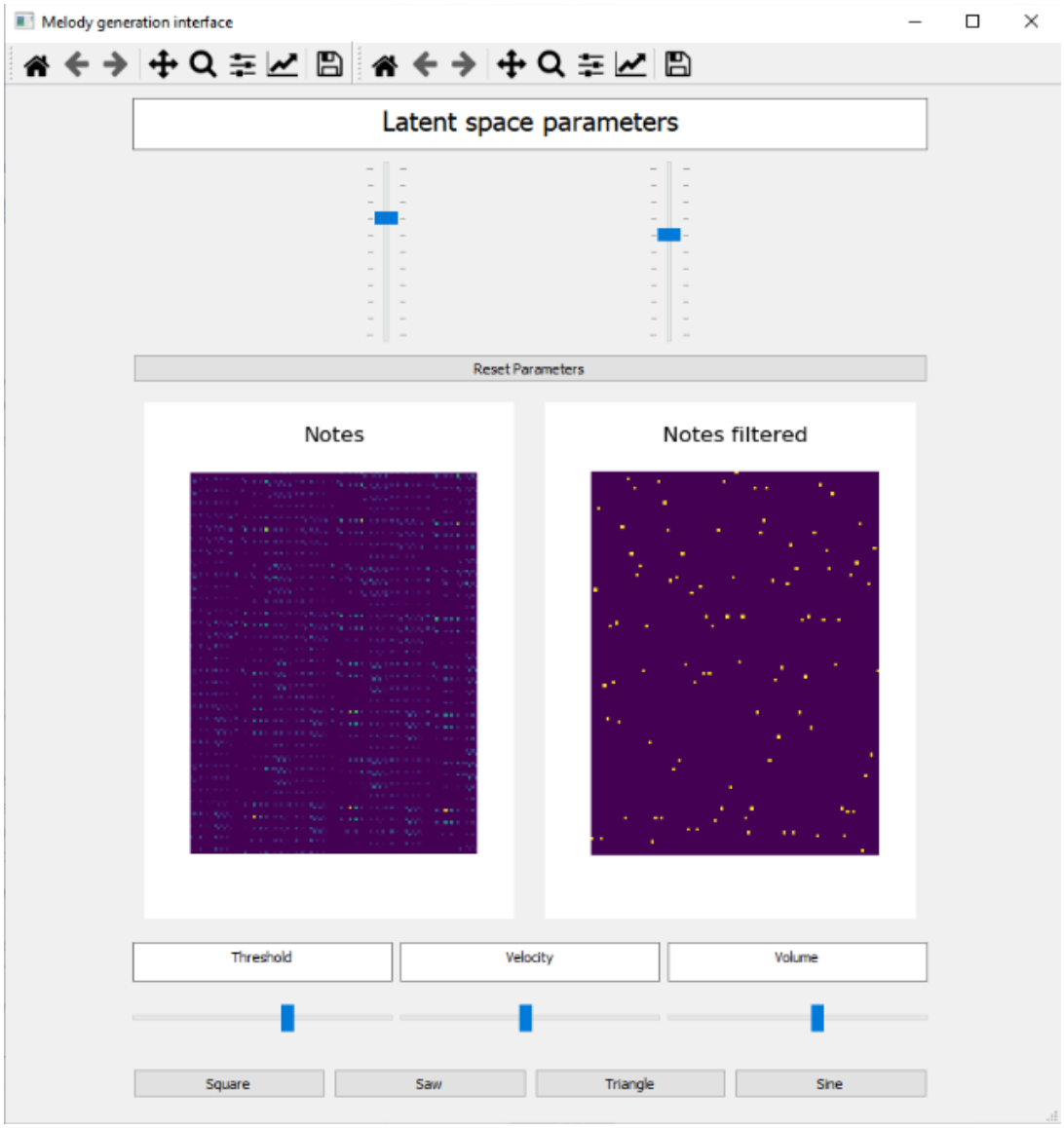

Figure 32 - Interface audiovisuelle pour VAE

Cette interface a été développé à l'aide de la librairie Python PyQt5. Elle détecte la dimension de l'espace latent du VAE pour en afficher les curseurs, rafraichis l'affichage des notes (en format pianoroll) à chaque modification et joue le résultat en cliquant sur une des formes d'ondes proposées en bas de la fenêtre.

A défaut d'obtenir des modèles entraînés décodant correctement les mesures du dataset, les résultats audios n'ont pas été satisfaisant. On peut s'en rendre compte en regardant les notes jaunes après filtrage, dont l'écoute se traduit quasiment par une cacophonie de notes.

VI.5. Harmonie mathématique

Grâce au fond théorique développé dans la partie III.1.b), nous avons la possibilité d'utiliser des informations supplémentaires sur les 12 notes de la gamme tempérée. En effet, avec une erreur que nous négligerons pour la majorité des auditeurs, nous considèrerons que les propriétés des notes de la gamme tempérée sont similaires à celles de la gamme chromatique de Pythagore. Le passage de l'une à l'autre se fait par correspondance des notes par proximité fréquentielle, en conservant leur nombre et leur ordre.

Les deux propriétés intéressantes des notes de la gamme de Pythagore sont qu'elles sont éloignées d'un certain nombre de quintes (multiples de 3/2 ou son inverse) et d'un certain nombre d'octaves (multiples de 2 ou son inverse).
En écrivant ces notes sous la forme $f*3^n/2^{n+m}$, il est possible d'identifier de combien de quintes elles sont éloignées de f et de combien d'octaves ont été nécessaires pour les ramener dans l'octave, on notera le nombre de quintes q et le nombre d'octaves o.
Nous pouvons utiliser ces informations pour savoir, parmi les notes de la gamme chromatique, lesquelles sont les plus "proches" en termes de quintes et d'octaves des notes de l'accord considéré. Le nombre de quintes, ascendant ou descendant servira de première mesure de distance entre les notes pour minimiser leur éloignement sur l'échelle des quintes. Le nombre d'octaves servira pour départager des gammes candidates avec le même score de distance, le sens des octaves n'importera pas.
Ainsi, si $r = f*3^n/2^{n+m}$, alors $q(r) = n$ et $o(r) = |m|$. On obtient, pour toutes les notes, le tableau suivant :

	1	2	3	4	5	6	7	8	9	10	11	12
r	f	$f\frac{2^8}{3^5}$	$f\frac{3^2}{2^3}$	$f\frac{2^5}{3^3}$	$f\frac{3^4}{2^6}$	$f\frac{2^2}{3}$	$f\frac{3^6}{2^9}$	$f\frac{3}{2}$	$f\frac{2^7}{3^4}$	$f\frac{3^3}{2^4}$	$f\frac{2^4}{3^2}$	$f\frac{3^5}{2^7}$
$q(r)$	0	-5	2	-3	4	-1	6	1	-4	3	-2	5
$o(r)$	0	3	1	2	2	1	3	0	3	1	2	2

Figure 33 - Tableau des valeurs du nombre de quintes et d'octaves

Les valeurs q(r) et o(r) sont stockées dans le programme dans deux listes.
Ces valeurs sont alors utilisées pour effectuer des calculs de distances entre les notes de l'accord considéré et toutes les gammes de 7 notes possibles afin de trouver celle dont la distance est la plus petite.
Nous définissons la distance sur les quintes entre un accord et une gamme comme suit :

$$D_f(Chord, Scale) = \sum_{scale_note \in Scale} \min_{chord_note \in Chord} |q(scale_note) - q(chord_note)|$$

Équation 6 - Formule de distance de quintes entre accords et gammes

La gamme possédant la distance minimale sera le meilleur candidat, car composée de notes proches de celles de l'accord en termes de sauts de quintes.
Si plusieurs gammes ont la même distance minimale, il est encore possible de les départager selon le nombre de sauts d'octaves avec la distance suivante:

$$D_o(Chord, Scale) = \sum_{scale_note \in Scale} o(scale_note)$$

Équation 7 - Formule de distance entre accords et gammes

Enfin, si aucune de ces distances ne suffit pour départager des gammes candidates, alors nous seront à cours de critères de sélection et nous feront un tirage aléatoire.

Le tableau ci-dessous présente un récapitulatif des résultats du programme implémenté (gamme trouvée par algorithme), comparés aux sélections de gammes effectuées auparavant (gamme sélectionnée à l'oreille). Les gammes ici sont les numéros de notes sélectionnées entre 0 et 11, le 0 étant le Do, et ainsi de suite.

Type d'accord	Gamme trouvée par algorithme	Critère algorithmique déterminant	Gamme sélectionnée à l'oreille	Nombre de notes différentes
Maj	0, 2, 4, 5, 7, 9, 11	Quintes	0, 2, 4, 6, 7, 9, 11	1
Min	0, 2, 3, 5, 7, 8, 10	Quintes	0, 2, 3, 5, 7, 9, 10	1
Dom	0, 2, 4, 5, 7, 9, 10	Quintes + octaves	0, 2, 4, 6, 7, 9, 10	1
Hdim7	0, 3, 5, 6, 7, 10, 11	Quintes + octaves	0, 2, 3, 5, 6, 8, 10	4
Dim	0, 3, 5, 6, 7, 10, 11	Quintes + octaves	0, 2, 3, 5, 6, 8, 9	4
Dim7	0, 2, 3, 5, 6, 7, 9	Quintes + octaves	0, 2, 3, 5, 6, 8, 9	1

Figure 34 - Tableau comparatif entre l'algorithme et le choix à l'oreille

On peut observer qu'il n'y a pas unanimité entre l'humain et le "mathématique", mais une majeure partie des cas ne présente qu'une seule note de différence. Dans ces cas-là, l'écoute de la gamme trouvée grâce aux mathématiques n'est pas déplaisante et sonne peut-être plus "standard" que celles qui ont été choisies, notamment pour les accords Maj et Min.

Les deux cas où beaucoup de notes diffèrent se produisent dans des cas où à la fois les quintes et les octaves sont nécessaires pour choisir la gamme et l'accord en question est plus sophistiqué (ou plus rare en musique), à l'exception de Dim7 où le résultat était plus en accord avec les attentes. Dans les deux cas présentant 4 notes différentes des attentes, le résultat par les mathématiques était tout aussi intéressant auditivement. Une simple remarque est que les gammes calculées dans ces cas-là comportent une succession de trois notes qui se suivent directement (5,6,7). Cela explique sans doute pourquoi elle n'a pas été sélectionné à l'oreille puisqu'un désir d'écarter les notes pour couvrir un maximum la gamme chromatique pouvait se justifier. Une telle contrainte peut être rajoutée dans le calcul des scores des gammes, encore faut-il lui donner un certain poids par rapport aux quintes et aux octaves.

En ce qui concerne les légères différences d'au moins une note entre les gammes choisies à l'oreille et celles choisies par algorithme, trois raisons peuvent expliquer ces écarts:

- La préférence pour l'effet d'une certaine dissonance, lié à un biais culturel.
- Plus généralement, le biais humain lié au fait que ces gammes aient été choisies que par deux personnes avec leur propre expérience et culture musicale,
- Les incertitudes liées au passage de la gamme de Pythagore à la gamme tempérée donnant un léger décalage entre notre perception en écoutant les notes de la gamme tempérée et le traitement de l'algorithme sur la gamme de Pythagore. Ce léger décalage peut être en fait significatif étant donné le nombre de fractions rationnelles présentes autour d'une fréquence de la gamme tempérée.

Nous remarquerons qu'aucun des accord n'a eu besoin d'un choix aléatoire pour lui assigner une gamme. A chaque fois, soit le nombre de quintes ou d'octaves a suffi.

VII. Application mobile

VII.1. Cahier des charges

VII.1.a) Fonctionnement et organisation générale

Le fonctionnement prévu de l'application mobile était divisé en deux parties :

- Le « backend » : utilisation d'un serveur web permettant de décentraliser certaines étapes lourdes en calculs tout en ayant la possibilité d'utiliser des données plus volumineuses.
- Le « frontend » : code de l'interface utilisateur présent dans l'application, responsable de toute la partie visuelle et de la gestion des interactions entre l'utilisateur et l'application.

Un critère imposé par Aubay INNOV était de développer le frontend avec Flutter. Flutter est un logiciel multi-plateforme, utilisé pour développer des applications à la fois sur Android, Max, IOS, Windows et Linux.
Le choix de coder le backend en python se justifiait par la facilité d'intégration des preuves de concept déjà développées en python. L'utilisation du framework Flask permettait de créer un serveur web en utilisant le protocole http pour répondre aux requêtes venant de l'application. Le service Amazon EC2 a permis de d'héberger le code du backend en l'exécutant sur une machine virtuelle.

L'organisation des vues de l'application a été faite de façon à présenter séparément les trois approches, recommandation, génération libre et génération guidée :

- Vue du menu principal :
  - Vue Recommandation
  - Vue Génération libre
  - Vue Génération guidée :
    - Bouton « play »
    - Volet d'options :
      - Transposition
      - Mode (Majeur ou mineur)
      - Vitesse
      - Choix d'instrument
    - Visuel des notes jouées

VII.1.b) Fonctionnalités

Les fonctionnalités à implémenter ont été organisées, après concertation entre les membres de la partie génération guidée, dans le diagramme de la page suivante.

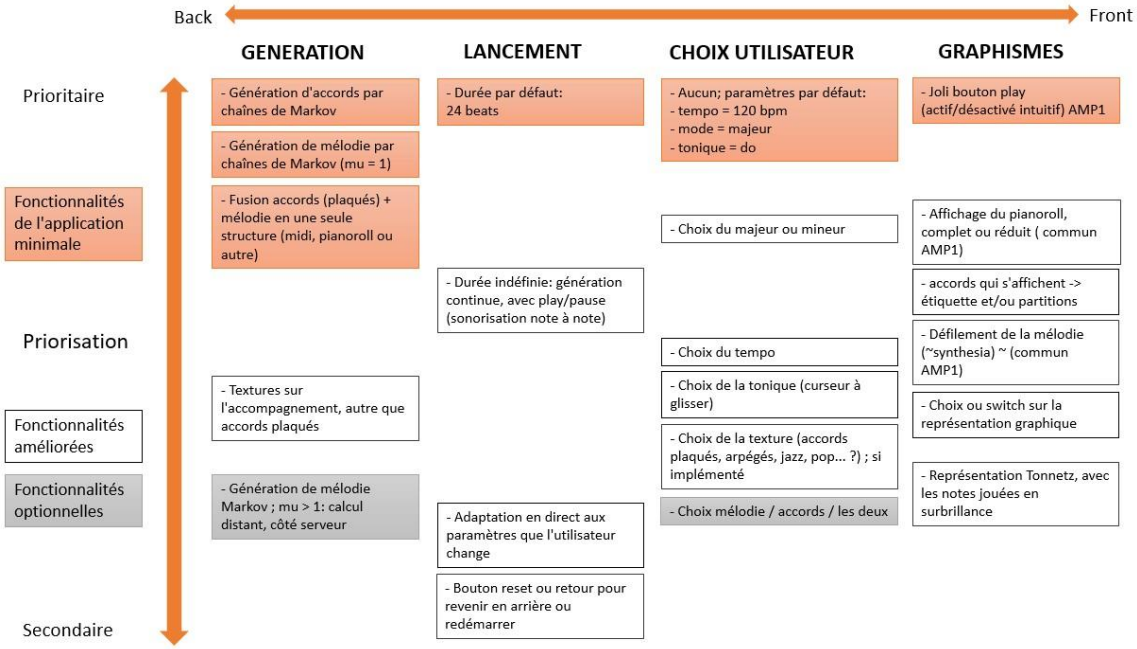

Figure 35 - Cahier des charges adaptatif du projet de génération guidée

Les blocs orange correspondent à l'application minimale à implémenter. Elle fera ensuite l'objet d'améliorations successives par ordre de priorité, selon le temps restant.

VII.2. Serveur web

Flask est un Framework en python, qui va être utilisé pour faire tourner les modèles sur un serveur dans le backend. Cette technologie est adaptée pour car nos travaux ont été réalisé en python, ce qui nous permettra d'exécuter directement nos modèles sur le serveur.

Nous nous sommes fixés l'état de développement de la partie génération de musique tel que décrivant le processus de la Figure 36. Le résultat était assez satisfaisant pour ne pas viser plus d'améliorations, cinq exemples d'audios générés ont été mis en annexe pour écouter le résultat.

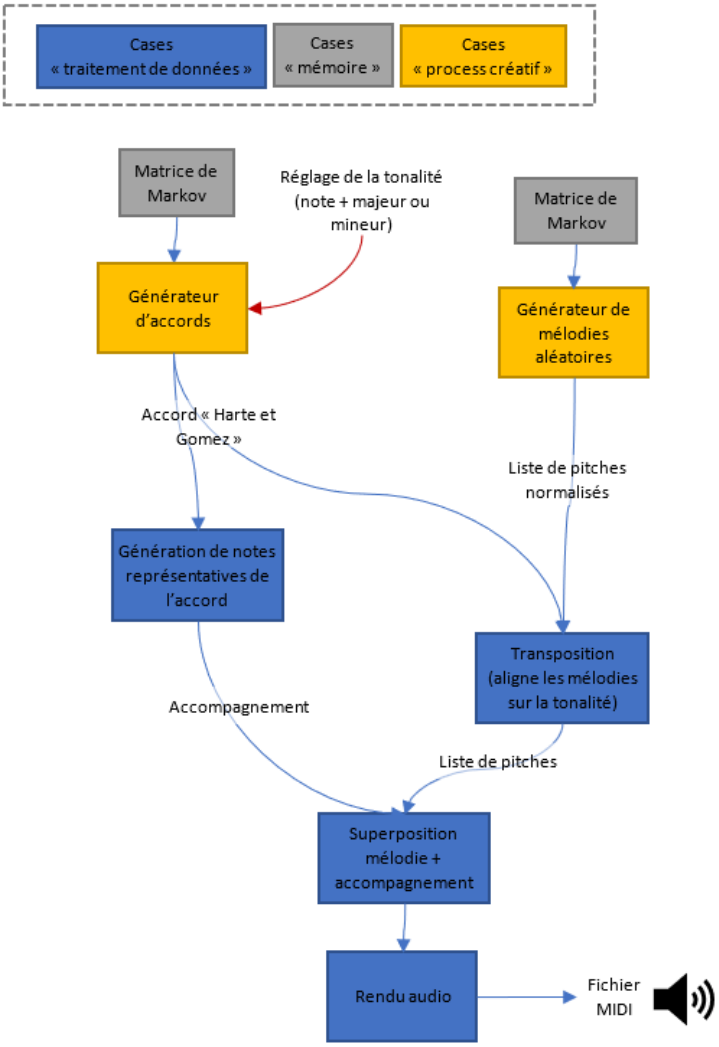

Figure 36 - Analyse système du générateur de musique

Les informations reçues par le serveur via le protocole http sont donc la note et le mode (majeur ou mineur). Le fichier MIDI obtenu a été convertis dans un format JSON personnalisé afin de pouvoir en extraire les données facilement sur Flutter. La méthode GET utilisée sur le serveur a permis à l'application, une fois la requête http envoyée, de télécharger automatiquement le fichier JSON généré par le code du backend.

VII.3. Frontend

Les fonctionnalités, autres que la tonalité, ont été gérées dans le code du frontend, avec Flutter. Par exemple, ni le choix de la vitesse de jeu d'une génération de musique ni le son n'ont une influence sur la génération guidée, il est donc inutile de les faire transiter entre le backend et le frontend.

D'un point de vue du design, nous avons tout d'abord contacté Marianne Fernandez une fois nos fonctionnalités listées afin qu'elle nous propose une maquette visuelle. Nous nous sommes ensuite appuyés sur ses productions pour coder l'application. Les figures qui suivent montrent le résultat final en lien avec le projet génération guidée.

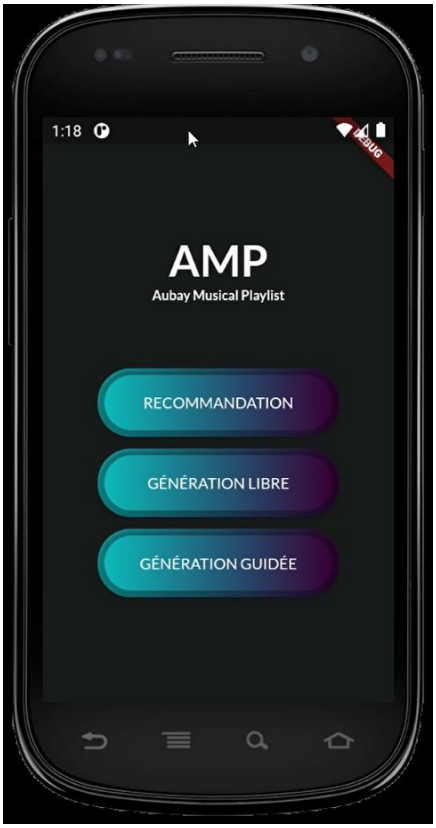

Figure 37 - Visuel de l'application - Ecran d'accueil

Figure 38 - Visuel de l'application - Page génération guidée

Figure 39 - Visuel de l'application - Volet d'options

Je précise que mon rôle dans le codage des aspects visuels et audios a été minime. L'essentiel de mes tâches de la phase projet étaient centrées sur les communications entre le frontend et le backend et la conversion des données dans des formats exploitables pour les autres stagiaires.

Le support de présentation que nous avons créé afin de présenter lors de la journée des stagiaires est présent en annexe.

VIII. Conclusion

Je tire de ces cinq mois de stage beaucoup d'expérience dans le développement de solutions techniques et dans l'organisation d'un travail au sein d'un groupe de 7 à 9 personnes. J'ai pu m'épanouir en profitant d'une grande autonomie dans le choix d'axes de recherches, tout en contribuant à un travail de collaboration entre musiciens et non-musiciens. Le contexte de crise sanitaire ayant éloigné l'équipe physiquement, nous avons fait en sorte, avec le support d'Aubay, de garder un contact constant et de bonnes conditions de travail.

J'ai eu l'occasion, à de nombreuses reprises, de présenter des résultats de recherche et des analyses dans un cadre professionnel et constructif. La présence du facteur humain dans l'objectif même du sujet de stage a nécessité une grande rigueur dans la formalisation d'une démarche de résolution de problème et dans l'exposition de résultats.

Enfin, si toutefois les résultats d'un générateur de musique sont indéniablement subjectifs, mes collègues et moi pensons avoir obtenus des musiques agréables à entendre, avec une certaine part d'imprévisibilité. Je dirais que notre travail a permis de faire avancer celui qui a été fait sur les réseaux de neurones. D'un point de vue de leurs particularités respectives, le premier étant plus prévisible que l'autre, l'autre étant moins subjectif que le premier, il serait envisageable de combiner les deux à l'avenir.

IX. Bibliographie

Agence culturelle Grand Est. (2009). *Définition du son.* Récupéré sur sonorisation-spectacle.org: http://www.sonorisation-spectacle.org/definition-du-son.html

Aubay. (s.d.). *About Aubay*. Récupéré sur Aubay: https://www.aubay.com/index.php/language/en/aubay-group/about-aubay/

Bertin-Mahieux, T., P.W. Ellis, D., Whitman, B., & Lamere, P. (2011). Récupéré sur Million Song Dataset: http://millionsongdataset.com/

Code Parade. (2018, 07 24). *Generating Songs With Neural Networks (Neural Composer).* Récupéré sur Youtube: https://www.youtube.com/watch?v=UWxfnNXlVy8

DDMAL. (s.d.). *The McGill Billboard Project*. Récupéré sur DDMAL: https://ddmal.music.mcgill.ca/research/The_McGill_Billboard_Project_(Chord_Analysis_Dataset)/

de Clercq, T., Temperley, D., Lusting, E., & Tan, I. (s.d.). Récupéré sur A Corpus Study of Rock Music: http://rockcorpus.midside.com/

Dong, H.-W., Hsiao, W.-Y., Yang, L.-C., & Yang, Y.-H. (s.d.). Récupéré sur Lakh Pianoroll Dataset: https://salu133445.github.io/lakh-pianoroll-dataset/

Eremenko, V., Demirel, E., Bozkurt, B., & Serra, X. (s.d.). Récupéré sur Audio-Aligned Harmony: https://mtg.github.io/JAAH/

fosfrancesco. (s.d.). Récupéré sur Github: https://github.com/fosfrancesco/performed-midi-dataset

Gabriel, C. (2019). *Technique de l'image B2&B2.* Récupéré sur Supports pédagogiques de Claude Gabriel: http://www.claudegabriel.be/Acoustique%20chapitre%208.pdf

Gérard, B., Goeldel, C., & Marion-Péra, M.-C. (2000, 01). *https://www.researchgate.net/figure/Attracteur-de-Lorenz_fig1_233979083.* Récupéré sur ResearchGate: https://www.researchgate.net/figure/Attracteur-de-Lorenz_fig1_233979083

Isophonics. (s.d.). Récupéré sur Isophonics.net: http://isophonics.net/datasets

Jazz Parser. (s.d.). *Jazz Corpus*. Récupéré sur Jazz Parser: http://jazzparser.granroth-wilding.co.uk/JazzCorpus.html

Lorenzi, A. (2019). *Localisation.* Récupéré sur Voyage au centre de l'audition: http://www.cochlea.eu/son/psychoacoustique/localisation#:~:text=L'audition%20binaurale%20est%2C%20%C3%A0,utile%20dans%20un%20milieu%20bruyant.

Magenta. (s.d.). *The MAESTRO Dataset*. Récupéré sur Magenta: https://magenta.tensorflow.org/datasets/maestro

Medoucine. (2018, 01). *Musicothérapie.* Récupéré sur Passeport santé: https://www.passeportsante.net/fr/Therapies/Guide/Fiche.aspx?doc=musicotherapie_th

midi_man. (2015). Récupéré sur Reddit: https://www.reddit.com/r/datasets/comments/3akhxy/the_largest_midi_collection_on_the_internet/

Nature Neuroscience. (2011, 01 22). *DOPAMINE ou pourquoi la musique fait du bien….* Récupéré sur santé log: https://www.santelog.com/actualites/dopamine-ou-pourquoi-la-musique-fait-du-bien

Pascal, C., & Tomas, N. (2000). Récupéré sur Ecole partouche: https://www.ecole-partouche.fr/fichiers_documents/MUSIQUE__ET__MA.pdf

Pfleiderer, M., Frieler, K., Abeßer, J., Zaddach, W.-G., & Burkhart, B. (2017). *Weimar Jazz Database*. Récupéré sur The Jazzomat Research Project: https://jazzomat.hfm-weimar.de/dbformat/dboverview.html

Raffel, C. (s.d.). Récupéré sur The Lakh MIDI Dataset v0.1: https://colinraffel.com/projects/lmd/

Remisse, S. (2020, 01 30). *Le pouvoir de la musique sur l'Homme à l'écoute du marketing.* Récupéré sur MBA MCI: https://mbamci.com/pouvoir-musique-homme-ecoute-marketing/

RTBF Tendance. (2019). *La dopamine joue un rôle dans le plaisir musical.* Récupéré sur RTBF: https://www.rtbf.be/tendance/bien-etre/psycho/detail_la-dopamine-joue-un-role-dans-le-plaisir-musical?id=10128360

Tricotism. (s.d.). *Index od /midi*. Récupéré sur Band-in-a-Box: http://bhs.minor9.com/midi/

Wikipédia. (2018, 10 18). *Loi de Weber-Fechner.* Récupéré sur Wikipédia: https://fr.wikipedia.org/wiki/Loi_de_Weber-Fechner

Wikipédia. (2020). *Dopamine.* Récupéré sur Wikipédia: https://fr.wikipedia.org/wiki/Dopamine#:~:text=Dans%20le%20syst%C3%A8me%20nerveux%20central%2C%20la%20dopamine%20joue%20un%20r%C3%B4le,le%20sommeil%20ou%20la%20m%C3%A9morisation.

Wikipédia. (2020, 06 03). *Harmonic Entropy.* Récupéré sur Xenharmonic Wiki: https://en.xen.wiki/w/Harmonic_Entropy

Wikipédia. (s.d.). *Consonance et dissonance.* Récupéré sur Wikipédia.

X. Table des illustrations

Figure 1 – Répartition du chiffre d'affaires d'Aubay par secteur d'activité (en 2018) 6
Figure 2 - Evolution des effectifs d'Aubay depuis 2014 6
Figure 3 - Modèle de perception des dissonances entre deux sons (Wikipédia) 8
Figure 4 - Notes de la gamme chromatique de Pythagore (Pascal & Tomas, 2000) 9
Figure 5 - Représentation linéaire de la gamme chromatique de Pythagore (Pascal & Tomas, 2000) 9
Figure 6 - Notes de la gamme tempérée (Pascal & Tomas, 2000) 9
Figure 7 - Notes de la gamme chromatique de Zarlino (Pascal & Tomas, 2000) 9
Figure 8 - Représentation linéaire de la gamme chromatique de Zarlino (Pascal & Tomas, 2000) 10
Figure 9 - Notes de la gamme chromatique d'Euler 10
Figure 10 - Représentation linéaire de la gamme chromatique d'Euler (Pascal & Tomas, 2000) ... 10
Figure 11 - Génération de visages par Deep Learning (Code Parade, 2018) 11
Figure 12 - Schéma récapitulatif du découpage du projet AMP 14
Figure 13 - Vue d'un tableau Kanban du projet AMP 16
Figure 14 - Diagramme d'approches en composition musicale 19
Figure 15 - Première représentation d'un système créatif 20
Figure 16 - Paradoxe de l'automatisation 20
Figure 17 - Représentation d'un système créatif rendu légitime 21
Figure 18 - Représentation des niveaux de réglages 22
Figure 19 - Représentation de l'écart entre le besoin du client (input) et sa solution (output) 22
Figure 20 - Modélisations, créativité et choix 23
Figure 21 - Espace libre et modélisation fixe 23
Figure 22 - Niveaux de consonnances des notes avec C majeur 24
Figure 23 - Différences entre l'usage du Deep Learning et l'apprentissage ciblé 26
Figure 24 - Graphe d'états élémentaire 27
Figure 25 - Notes de la gamme C majeur 28
Figure 26 - Notes de la gamme C mineur 29
Figure 27 - Représentation des notes du solo de Art Pepper - Anthropology, en fonction du temps 29
Figure 28 - Détection d'outliers pour le découpage de solo 29
Figure 29 - Affichage de la gamme détectée par rapport aux notes d'un solo 30
Figure 30 - Matrice de Markov pour une mémoire μ=1 31
Figure 31 – zoom sur la matrice de Markov pour μ=2 32
Figure 32 - Interface audiovisuelle pour VAE 34
Figure 33 - Tableau des valeurs du nombre de quintes et d'octaves 35
Figure 34 - Tableau comparatif entre l'algorithme et le choix à l'oreille 36
Figure 35 - Cahier des charges adaptatif du projet de génération guidée 38
Figure 36 - Analyse système du générateur de musique 39
Figure 37 - Visuel de l'application - Ecran d'accueil 40
Figure 38 - Visuel de l'application - Page génération guidée 40
Figure 39 - Visuel de l'application - Volet d'options 41

Équation 1 - Nombre d'arrangements sans répétition en fonction du nombre de modélisations 25
Équation 2 - Expression d'une matrice de Markov 27
Équation 3 - Expression d'une matrice de recensement 27
Équation 4 - Lien entre matrice de Markov et matrice de recensement 27
Équation 5 - Equation du vecteur d'état initial 27
Équation 6 - Formule de distance de quintes entre accords et gammes 35
Équation 7 - Formule de distance entre accords et gammes 35

XI. Table des annexes

- Offre de stage - Aubay Musical Playlist.pdf
- Diagramme de Gantt prévisionnel.pdf
- Diagramme de Gantt réel.pdf
- Recensement de datasets musicaux.pdf
- Présentation Journée des Stagiaires.pdf
- Exemple de musique générée 1.mid
- Exemple de musique générée 2.mid
- Exemple de musique générée 3.mid
- Exemple de musique générée 4.mid
- Exemple de musique générée 5.mid