Université de Lille 1 SCIENCES ET TECHNOLOGIES

Projet n°85 : *Développement d'une interface Javascript dédiée à la visualisation harmonique de séquence musicales*

Etudiant : Pierre Caruyer

Encadrant : Louis Bigo - Laboratoire CRIStAL(équipe Algomus)

Table des matières

1.Introduction

Habituellement, lorsque l'on représente un morceau de musique, on le représente sous la forme d'une suite de notes appelée partition.
Dans une partition, chaque note est représentée, dans un ordre chronologique, par son timbre (do, ré, mi, fa, etc ...), sa hauteur – grave ou aiguë - sa durée et son intensité. ce qui permet aux musiciens de reproduire le son comme le compositeur l'a prévu lorsqu'il a écrit son morceau.

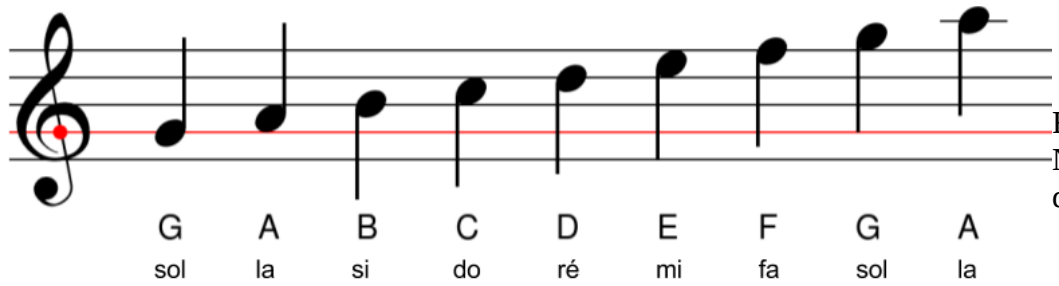
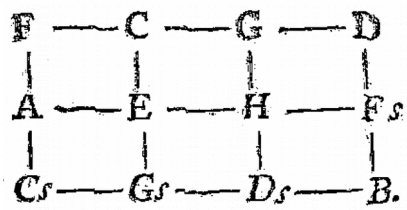


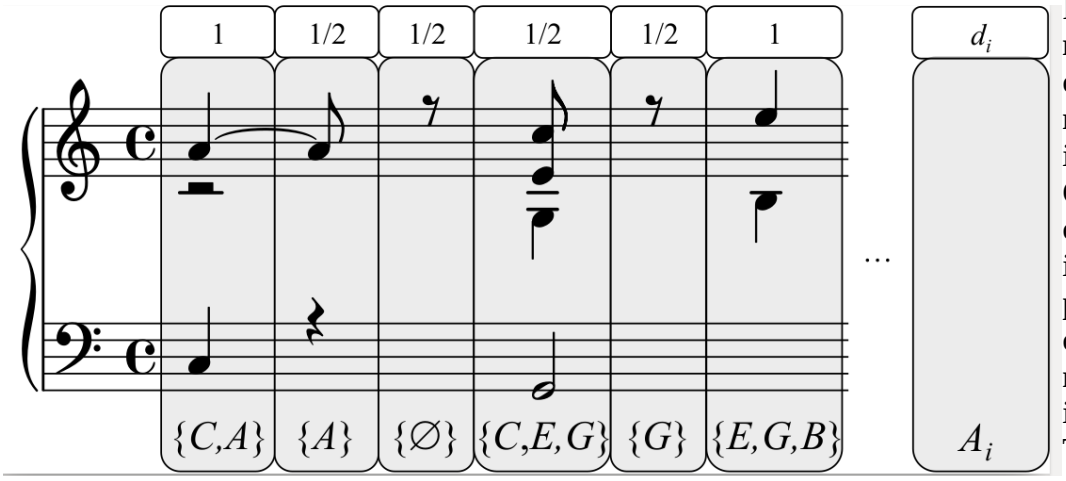
Fig. 1
Note représentées dans une portée

Fig. 2 Tonnetz d'Euler



Il existe d'autres façons de représenter la musique comme le Tonnetz qui a été inventé par Leonhard Euler en 1739.
Le Tonnetz correspond à une organisation des notes de musique et des accords dans l'espace, dans laquelle la distance entre les éléments reflète leur consonance sur le plan musical. La notion de proximité entre les notes permet d'attribuer un score (par exemple, le score sera plus ou moins bas en fonction du nombre d'accord dissonants que comporte la musique) à toute pièce de musique, en offrant ainsi différentes approches pour l'analyse et la classification stylistique.

Il est possible de faire le lien entre une partition et le Tonnetz car le Tonnetz permet de représenter un accord (ensemble de notes) à un moment précis de la musique. Une partition peut donc être décomposée en une suite d'accord pour être représentée dans le Tonnetz.



Ici, on peut voir la représentation d'un cours morceau de musique joué par deux instruments à la fois. Chaque portée correspondant à un instrument. Sous les portées, on peut voir la correspondance des notes jouées par les instruments dans le Tonnetz.

2.Présentation du projet

2.1.L’équipe Algomus

Ce projet est proposé par l’équipe « Algomus », spécialisée dans l’élaboration d’algorithmes dédiés à l’analyse musicale, aussi appelée musicologie assistée par ordinateur.
L’équipe Algomus développe des outils permettant aux musicologues d’étudier des pièces musicales de différents styles sur les plans mélodiques, harmoniques et rythmiques.

2.2.Objectif global

L’analyse et la représentation de progressions harmoniques (suite d’accords) y constitue un axe central. Le logiciel HexaChord développé en Java (site web en footnote) a été développé dans le but de visualiser et d’analyser ces progressions durant leur écoute à l’aide de représentations dérivées du *Tonnetz*.
Le présent projet a pour but d’étudier les outils disponibles pour l’implémentation de fonctionnalités de bases de HexaChord en langage JavaScript, pour cela nous allons nous baser sur deux applications existantes, TonnetzViz et mudcube.js.

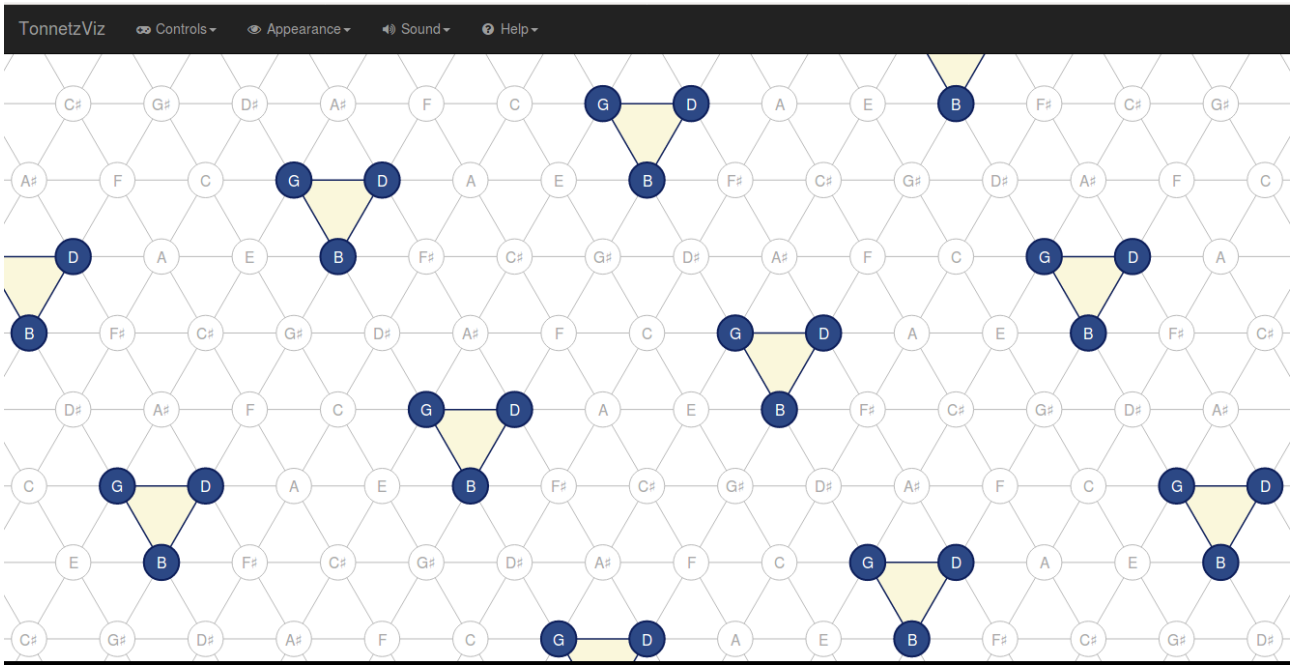
3.Les applications existantes

Deux applications existantes constituent le point de départ de ce projet.

3.1.Présentation de TonnetzViz

« TonnetzViz » est une application JavaScript en ligne permettant de jouer des notes en tapant au clavier et de voir leur représentation dans le Tonnetz en direct.

Ici, la représentation du Tonnetz dans l'application TonnetzViz :



3.2.Présentation de mudcube.js

L'application, « mudcube.js » permet de lire une pièce musicale sous la forme d'une chaîne de caractères (format base64), et de la jouer en générant les signaux MIDI correspondant, comme le ferait un lecteur multimédia classique.

Le format base64 est une ensemble de façon d'encoder de l'information en binaire dans une chaîne de caractères, chaque caractère contient 6 bits d'information.

4.Objectifs du projet

4.1.Premier objectif

L’objectif du projet proposé est de développer un prototype en JavaScript permettant la visualisation de fichiers au format MIDI dans le Tonnetz comme le fait Hexachord (cf. Lexique) .

Le format midi est un format qui permet non seulement de lire une musique mais au aussi de connaître la date à laquelle jouer chaque note. Ce format rend ainsi l’analyse musicale assistée par ordinateur plus facile car il permet à la fois d’analyser le fichier dans son ensemble mais aussi accord par accord.
Une note au format midi est conçu comme on représente une note sur une partition, elle contient des informations sur la durée de cette note, son timbre, l’instrument avec laquelle la jouer, etc ...

Le premier objectif du projet consiste à élargir les fonctionnalités du visualiseur TonnetzViz afin de permettre à l’utilisateur de visualiser le déroulement de fichiers MIDI et non seulement de notes saisies de manière ponctuelle à l’aide du clavier. La lecture de fichiers MIDI se fera à l’aide de la bibliothèque midi.js employée par l’application « mudcube ».

4.2.Second objectif

La suite du projet consiste à ajouter à l’application un parseur de fichier MIDI permettant au visualiseur de se référer à une structure de données chargée en mémoire plutôt qu’à écouter les notes MIDI saisies à la volée au cours de la lecture du fichier MIDI. Cette fonctionnalité permettra notamment une navigation plus facile dans la pièce de musique (retour de x pas en arrière, en avant, etc.). La manipulation de la pièce musicale sous la forme d’une structure de données dans l’application ouvre par ailleurs d’intéressantes possibilités pour l’implémentation à venir de fonctionnalités permettant l’analyse automatique de la pièce musicale sans avoir à la lire à l’aide du lecteur.

Un objectif supplémentaire consiste à développer un côté serveur de cette application pour pouvoir mettre en ligne l’application et y accéder de n’importe où et par tout le monde.

5.Enjeux du projet

Le fait de développer à nouveaux les fonctionnalités du logiciel HexaChord, pour une application, à l’aide de technologies plus modernes, notamment, le langage JavaScript, permet d’une part, de permettre à l’équipe de maintenir son application plus facilement en évitant de ressortir une nouvelle version de l’application sous forme d’un fichier au format (.jar) dès qu’une modification est apportée à l’application.

La simplicité d’accès (seul un navigateur est nécessaire) permettra d’autre part à l’application d’être utilisée plus facilement et de toucher ainsi un public plus large, notamment chez les musicologues et les étudiants en musique qui constituent le publique visé majoritairement.

Enfin, le développement de fonctionnalités côté serveur devrait permettre à terme l’équipe de récupérer un certain nombre de données utilisateurs qui pourront être utilisée dans un cadre d’apprentissage automatique.

6.Objectifs remplis

6.1.Premier objectif

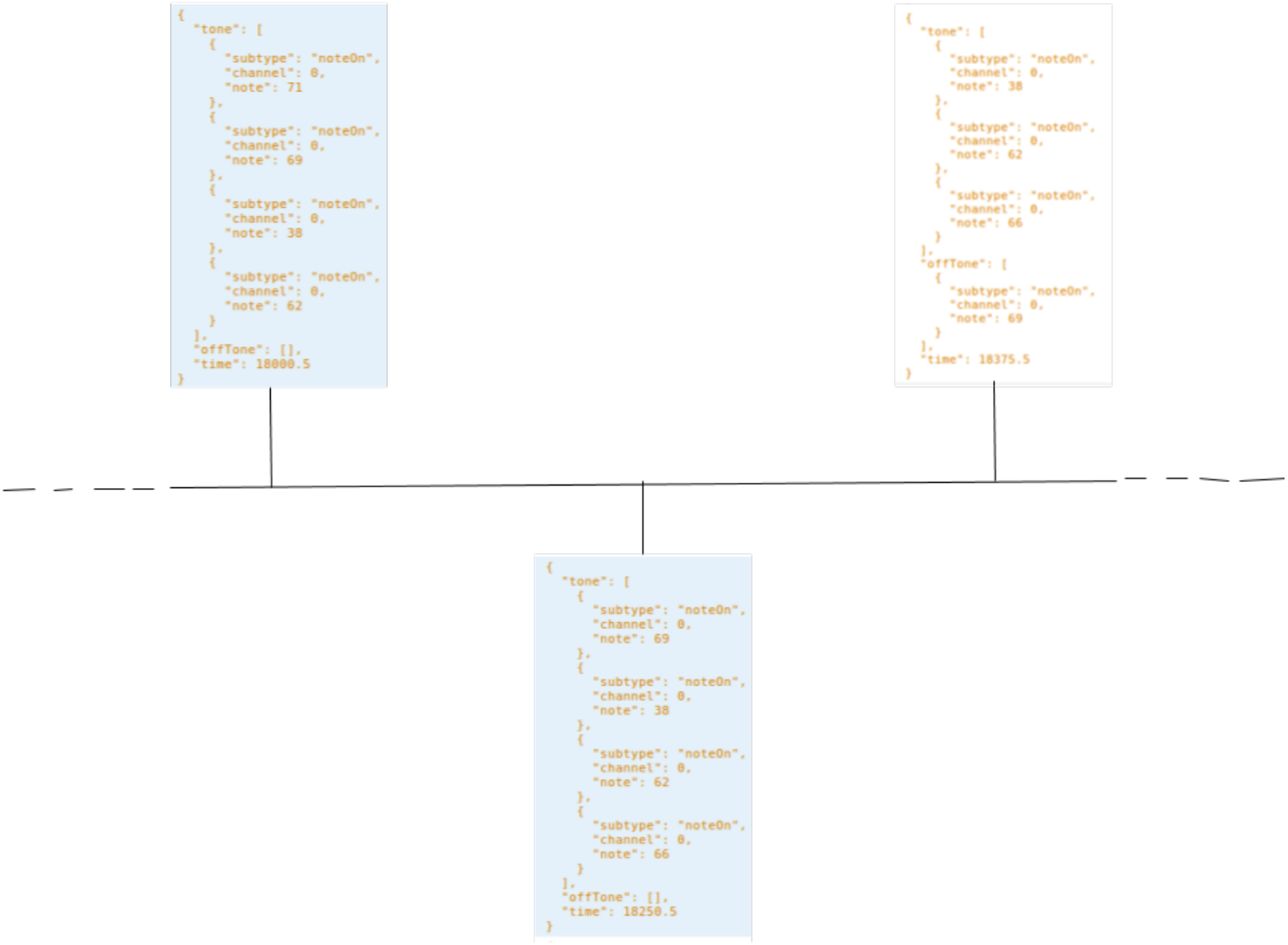
Le premier objectif est rempli : les notes constituant une pièce musicale fournies sous la forme d’une chaîne de caractères (au format base64) sont automatiquement affichées dans l’interface graphique de l’application TonnetzViz.

6.2.Second objectif

Le second objectif principal a également été rempli, l'utilisateur peut sélectionner un fichier midi qu'il souhaite jouer depuis l'application.
L'affichage se fait dorénavant grâce à une structure de donnée correspondant à une fresque chronologique dans laquelle il est possible de naviguer.
A chaque nouvel évènement rencontré, une nouvelle date est ajoutée à cette fresque, avec les notes correspondantes à cette date.
Une fois cette fresque construite, il est possible d'interroger celle-ci pour afficher les notes dans le Tonnetz.

Pour naviguer dans le fichier, une fonction permettant d'avancer ou de revenir en arrière dans le fichier a également été implémentée.
Il suffit d'appuyer sur entrée pour avancer et sur retour pour reculer dans le fichier.
Il est possible de définir la taille du saut dans le temps dans l'interface de l'application.
L'unité utilisée pour ce saut se fait en nombre de d'accord que l'on veut ignorer.
Si on utilise un pas de 1, alors le fait de revenir en arrière rejouera le dernier accord qui a été joué.

Cette fresque ressemble, une fois construite, à la figure suivante :



6.3.Objectif secondaire

Nous avons également commencé à implémenter un serveur avec Nodejs et le module Express. Pour l’instant nous avons seulement adapté l’interface de l’application.

7.Conclusion

Apport pour l’équipe Algomus :

Cette application est pour le moment uniquement disponible en accès local.
Les prochaines étapes seront reliées au développement d’un serveur, ce qui permettrait de rendre accessible cet outil à d’autres musicologues dans le monde. Cette solution permettrait également à l’équipe Algomus de pouvoir mettre à jour l’application une seule fois au lieu de redistribuer l’application à chaque mise à jour.

Apport pour l’étudiant :

Ce projet m’a permis de découvrir le langage JavaScript que j’avais envie d’apprendre à utiliser depuis longtemps. J’ai également appris à utiliser Nodejs.
De plus, ce projet m’a permis d’élargir ma culture musicale, notamment sur des choses peu connues comme le Tonnetz, le format de fichier midi qui apporte beaucoup de possibilités et la musicologie en générale.

8.Remerciements

Je tiens à remercier M. Bigo pour avoir pris de son temps pour me conseiller tout au long de ce projet.

9.Lexique

Accord : ensemble de notes.

Consonance : combinaison de sons dite « agréable » à l'oreille.

Tonnetz : signifie littéralement « réseau tonnal » en allemand et permet de représenter la musique sous forme d'un graphe où chaque nœud est une note et les arêtes représentent les accords.
(cf. annexe)

MIDI : format de fichier musical dans lequel chaque note est représentée par une durée d'activité pendant laquelle la note sera jouée, une note qui permettra de savoir quelle note (fa, do, etc …) doit être jouée ainsi que le délai entre chaque note.

Hexachord : programme développé par l'équipe Algomus permettant de visualiser une musique de format MIDI dans le tonnetz.

10.Annexe

Fig. 1 : http://www.musictheory.net/lessons/10

Fig. 2 : https://fr.wikipedia.org/wiki/Tonnetz#/media/File:Speculum_musicae.png

Hexachord : https://www.youtube.com/watch?v=NQ7LkWCzKxI

Tonnetz : https://i.ytimg.com/vi/NQ7LkWCzKxI/hqdefault.jpg

TonnetzViz : https://cifkao.github.io/tonnetz-viz/

MIDI.js : https://galactic.ink/midi-js/