

**Projet :   
Py-Reconote**

ISN Terminales S

Balthazar CRAVATTE

Mattéo RATET

Gilles Tuffet

**Sommaire**

**I - Introduction et présentation du projet**

**II - Analyse du besoin, recherche d’idées**

**& structuration du projet**

**A –** Recherche d’idées préliminaires

**B** – Analyse du besoin et définition des objectifs

**C** – Positionnement par rapport aux solutions existantes

**III – Structure du projet & Collaboration**

**A –** Moyens mis en œuvre

**B** – Schéma d’organisation

**C –** Quelques captures montrant le projet final

**IV – Réalisation personnelle**

**V – Intégration & Validation**

**A** – Intégration du projet

**B** – Tests, Corrections et Validation

**VI – Bilan & Perspectives**

**VII – Diffusion du Projet**

**VIII - Index**

**I -** Introduction et présentation   
du projet

D’après Victor Hugo “La musique, c’est du bruit qui pense”. C’est à dire que la musique est une suite d’onde mécanique audible qui évoque aux humains des émotions, des souvenirs, des histoires.

Aujourd’hui, de par son caractère universel, la musique est un élément culturel majeur. De nombreux imprudents se lancent dans l’apprentissage de cet art et découvrent vite son potentiel infini. C’est pourquoi, l’apprentissage de la théorie et la formation de l’oreille musicale est chronophage.

Cependant, l’explosion de l’internet et du numérique a changer notre manière d’apprendre et d'appréhender la musique. De nombreux adolescent ne voulant, ou n’ayant, pas les moyens de se payer des cours, s'achètent une guitare et apprennent de manière autodidacte la musique.

Mais Les outils musicaux numériques ne s’arrêtent pas au répertoire de tablature, telle que *Ultimate-Guitare* et compagnie. En effet, des outils performant, comme *Meludia,* permet de s'entraîner à forger son oreille musicale. Un rêve pour tout musicien que d'acquérir l’oreille absolue, cette faculté de pouvoir identifier n’importe quelle note juste en l’entendant. Un rêve pratiquement inatteignable, car c’est une faculté innée. Si on ne la possède pas dès la naissance il est impossible de l’avoir durant sa vie.

Or, avoir ce don en poche peut être un atout majeur pour toute pratique instrumentale ou de composition. C’est alors que l’outils numérique peut venir aider à la quête du graal.

En effet, puisque l’onde sonore est une onde mécanique progressive, elle peut être résumé par deux paramètres : la pression acoustique en fonction du temps. En analysant un fichier sonore numérique et en extrayant des informations physiques de l’onde, nous pourrons en déduire sa note (c’est-à-dire sa fréquence).

Ainsi, l’outil du numérique pourrait permettre d’ouvrir au plus grande nombre des facultés hors du commun. Nous allons donc nous ne demander, de quelle manière l’outils numérique peut-simplifier l’analyse musicale ?

**II -** Analyse du besoin, recherche d’idées   
& structuration du projet

**A –** Recherche d’idées préliminaires

La première chose à faire fut de s’accorder tous les trois sur un projet à mener. Deux idées principales émergèrent :

* Un moteur physique pour simuler des phénomènes mécaniques ou thermodynamiques
* Et un projet axé sur le son et la musique.

Au bout de quelques semaines nous nous sommes tournés vers le son, en espérant avoir également l’occasion d’étudier la physique des ondes ainsi que les transformés de Fourier dont nous avions déjà entendu parlé.

Balthazar étant musicien et ayant souvent recours à des logiciels de MAO (musique assisté par ordinateur), il souhaitait créer un programme qui lui serait utile et qu’il pourrait maîtriser facilement.

Nous nous sommes donc orientés vers de la génération d’ondes sonores. A partir de la là le projet est resté assez vague pendant quelques temps. L’objectif premier semblait selon Balthazar être la création de l’oscilloscope qui générerait les signaux périodiques auxquels on appliquerait des filtres pour modifier le son.

Cependant, à ce moment-là, le projet et les objectifs étaient mal compris, mal définis et changeaient régulièrement et nous ne savions pas par où commencer Nous stagnions. De plus Matteo et Gilles n’avaient pas d’ordinateur portable et nous ne pouvions pas véritablement mettre à profit les heures d’ISN car nous ne pouvions rien installer sur les ordinateurs de lycée.

**B –** Analyse du besoin et définition des objectifs

Quelques semaines avant la deadline, nous avons finalement réussi à définir nos attentes et nos objectifs.

Mettre au point un programme en langage Python 3.6 capable d’une part d’analyser une séquence de notes à partir d’un fichier audio brut et de retourner leurs noms (en isolant leurs fréquences à l’aide de transformés de Fourier, une solution mathématique existante donc, mais qui nécessite une certaine compréhension du sujet), les intervalles les séparant ainsi que la gamme à laquelle elles appartiennent, et d’autre part de générer et d’encoder un fichier sonore correspondant à une gamme sélectionnée.

**C –** Positionnement par rapport aux solutions existantes

Des alternatives existent pour lire et analyser les fréquences d’un fichier son. En effet, Notre projet peut s’apparenter à des applications telle que Shazam.

Néanmoins, *Py-reconote* répond de manière original à cette problématique. Si nous prenons l’exemple de Shazam, une application qui permet de reconnaître les morceaux. Son principe est simple :  assigner aux 11 000 morceaux qu’il a dans sa base de données une empreinte unique. Lorsque le programme est lancé, il est capable de reconnaître cette empreinte et affiche donc en sortie le morceau correspondant.

La grande différence entre le *Py-reconote* et Shazam, est que notre programme reconnaît non pas un morceau mais une suite de notes et qu’il ne se base pas sur l'assignation d’empreintes pour analyser la fréquence.

En effet, le *Py-reconote* utiliser les transformations de Fourier pour calculer les fréquences caractéristiques des notes du fichier en question.

Le *Py-reconote* n’est donc pas un reconnaisseur de morceaux mais un analyseur musical et scientifique. Cet outil permet donc de reconnaître musicalement une note par le calcul de sa fréquence. Ce calcul se basant sur sa la forme spatiale de l’onde.

Le projet *Py-Reconote* répond donc à une problématique de simplification de la théorie musicale. En effet, Le but principal de cet outil numérique est de pouvoir d’après l’analyse d’une fichier sonore y extraire les propriétés musicales et physiques. Les fichiers traités par *Py-reconote* sont des suites de notes croissantes.

**III –** Structure du projet & Collaboration

**A –** Présentation des outils collaboratifs et organisation du travail de groupe.

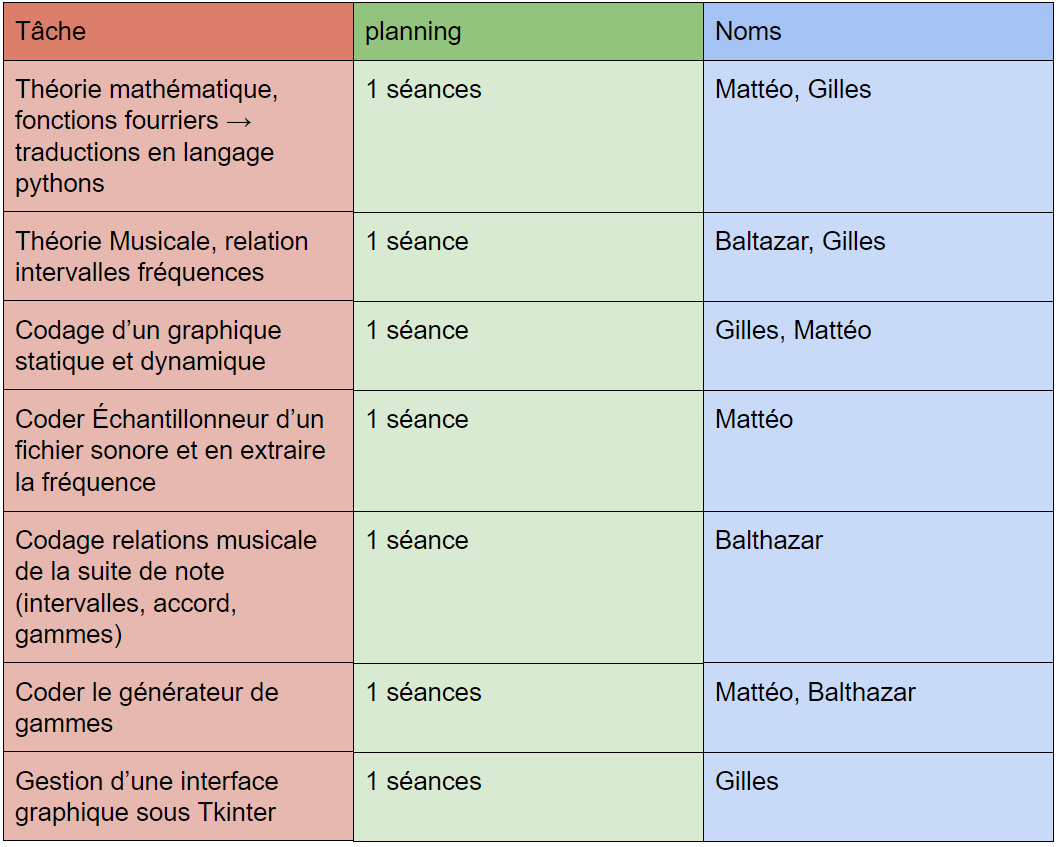
Étant habitués à faire des travaux collectifs ensemble. Nous avons des méthodes et habitudes de travailles bien rodées.

Nous avons principalement utilisé la plateforme Curse Voice. C’est un logiciel de messagerie instantané et de chat vocal qui a la particularité de marcher sous la forme de serveur, ou chacun peut se connecter et se déconnecter à sa guise sur la conversation. Cette plateforme permet une flexibilité d’utilisation et une réduction des contraintes techniques liées à la synchronisation de nos emplois du temps respectifs. Nous avons codé la plupart du temps tous ensemble, de manière à améliorer notre efficacité.

La deuxième plateforme collaborative que nous avons utilisé est Google Docs. C’est un traitement de texte collaboratif en ligne. Nous nous en servons très régulièrement dans les autres matières dès que nous devons faire un projet en groupe. L’avantage de cet outil, c’est que nous y sommes très familiarisés avec.

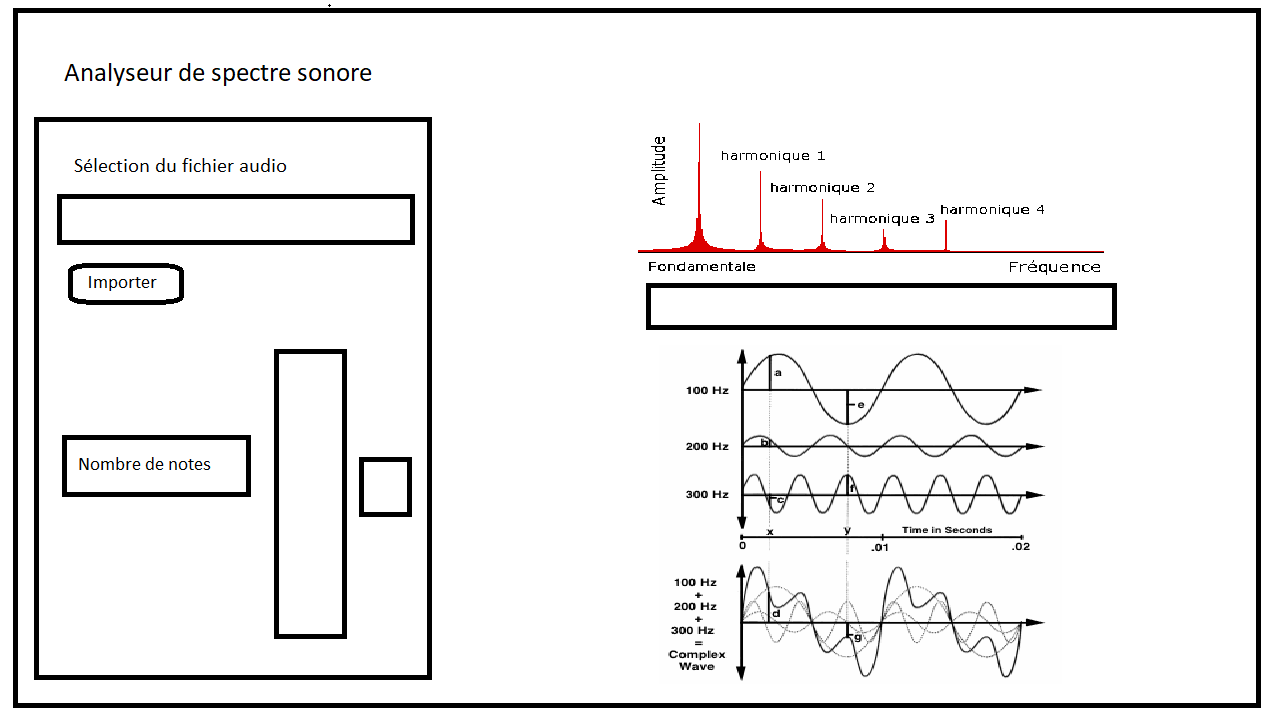
Et enfin, la plateforme en ligne Github. C’est un service web d'hébergement et de gestion de développement de logiciels. L’avantage de Github sur L’open source est qu’il amène chaque contributeur à télécharger les sources du projet et à proposer ensuite ses modifications à l’équipe du projet, GitHub repose sur le principe du fork (embranchement).  Chaque personne qui change le code devient contributeur à part entière. Nous, nous en somme servit pour s'échanger les différentes versions du code. Cependant le fait que ce n’est pas une plateforme collaborative instantané nous a posé problème à plusieurs reprises. En effet, deux personnes travaillant au même moment, peuvent ne pas travailler sur la même version du programme. Ainsi des erreurs persistaient. Nous pensons que pour les prochains projet informatique nous nous tournerons vers d’autres alternatives.

Répartition des tâches :



**B –** Schéma d’organisation

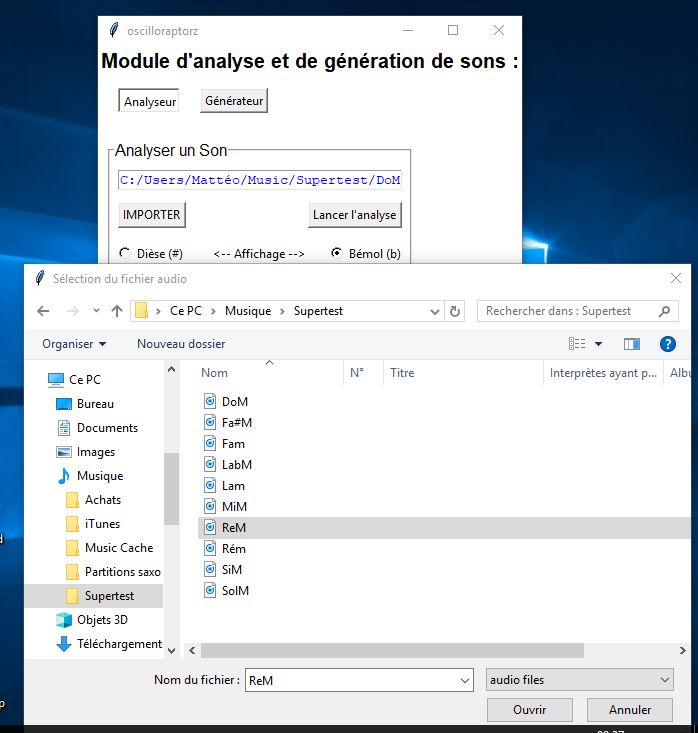
Voilà un premier croquis d’organisation (fait sur Paint) que l’on avait fait avant de commencer à coder :



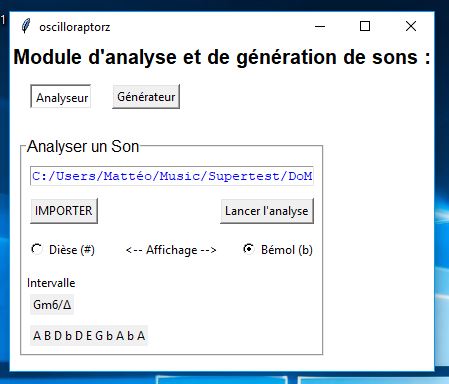
On peut déjà voir la partie de gauche dédiée aux paramètres. On remarque que les sorties textuelles ainsi que l’onglet « Générateur » sont absents.

Les seules sorties envisagées au début étaient des graphiques : le spectre sonore (le cadre noir en dessous représentait les notes en format texte avec leurs fréquences associées que l’on voulait afficher sous chaque pic), ainsi que la décomposition en fonctions sinus du signal périodique original. Bien sûr, nous envisagions au départ de rendre ce deuxième graphique dynamique, ce qui s’est avéré quasi impossible sans interférer avec Tkinter.

**C –** Quelques captures montrant le projet final

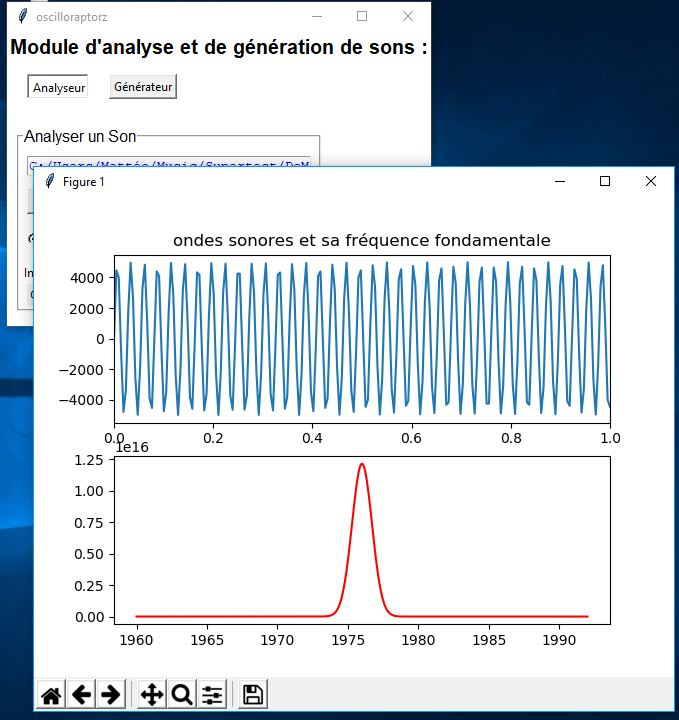


Sur cette capture on peut voir l’importation d’un fichier sonore test après avoir cliqué sur le bouton « importer » (qui ouvre l’uploader de fichier). Le fichier sélectionné, on peut voir apparaître son chemin d’accès dans le champ texte de la fenêtre en arrière-plan.

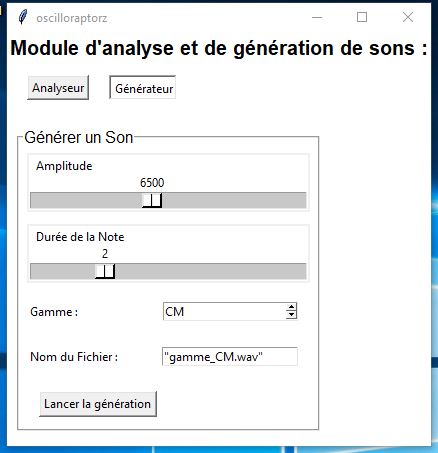


Sur cette deuxième capture on peut voir le programme, en mode « Analyse » après importation et analyse d’un fichier (dont on peut voir le chemin d’accès apparu dans le premier champ).

L’affichage est en mode « bémol » (il est en « Dièse » par défaut), et les notes obtenues en sortie sont effectivement en bémol.



Sur cette troisième capture du module d’analyse, on peut voir les deux graphiques. Le premier (en bleu, représente) l’intensité du son en fonction du temps pour une seconde. Le deuxième donne la fréquence fondamentale de la première note analysée).



Cette dernière capture montre le programme en mode générateur et toutes les entrées prises en comptes : l’amplitude du son (de 0 à 15000) la durée de la note (de 1 à 5 secondes) sont définies à l’aide de scalebox.  
La gamme (ici Do Majeur) choisie à l’aide d’une spinbox.

Enfin le nom du fichier que l’on entre au clavier.

**V –** Validation & Intégration

**A** – Intégration du Projet

Au sein du projet Py-Reconote, j’ai travaillé principalement sur l’analyse musicale du programme. Dans un premier j’ai du convertir en note de musique des fréquences. En musique, il y a une périodicité dans les notes. En effet une note qui a une fréquence 2n (où n est un entier relatif) supérieure ou est inferieure est la même note mais a une hauteur différente. On appellera cette intervalle une octave. En occident, actuellement, nous divisons une Octave en 12 demi-ton (1ton = 2 demi-tons) selon la gamme tempéré. C’est à dire que les notes sont tout un peu faux mais qu’il n’y a pas d’intervalle dissonant, comme ça a été le cas au moyen âge.

Pour reconnaître n’importe qu’elle note à partir des fréquences, on va transposer toute les fréquences sur une octave d’étude. C’est le principe de la fonction *octave* où on lui donne en entré une fréquence, elle va la transposer à l’octave d’étude 3 (comprise entre**)**.

Elle ressort la fréquence (variable = freq) transposé ainsi que le numéro (variable = n) de l’octave selon l’octave 3. Si la note est inferieure à 255, on doit la transposer à des octaves supérieures, donc on la multiplie la fréquence par des puissance de deux 2. Dans le cas ou la fréquence est supérieure on la divise par des puissances de deux. La boucle s’arrête quand la fréquence est dans l’intervalle.

Une fois la fréquence est transposée dans l’octave d’étude, on l’injecte dans la fonction *NoteFreqJuste*. Cette fonction faisant appelle à la fonction octave permet d’identifier une note à partir d’une fréquence à travers des conditions. Cette fonction retourne en sortie la note, en notation Anglophone (A = La, B = Si , C=Do, D = Ré, E = Mi, F = Fa, G = Sol) et une valeur juste de la fréquence de la note dans la gamme tempéré.

À partir de cette fonction, est créer deux liste. Une liste Contenant la notation des notes, qui servira pour l’affichage et une autre avec les fréquences justes, utilisées pour l’identification de la nature musicale de la note. On entend par identification de la nature musicale de la suite note, savoir si il y a deux, quelle est l’intervalle. Si il y a trois ou quatre note connaître l’accord. Et pour terminer si il y a 6 note ou plus savoir quelle est le mode de l’accord. Pour cette partie du code il suffit de faire appelle à la fonction *Sortie*, qui distribue selon la taille de la suite, l’appelle de fonction d’identification. On rentre dans la fonction suite deux liste : La Liste *ListeNote*, la liste d’affichage, contenant l’appellation anglo-saxone des notes de musique, ainsi que la liste *NoteFreqJuste*, notre liste d’étude. Cette fonction (*sorti*) va faire appelle à d’autre fonction d’identification. Telle que *AccordDeffinisseurTaille4*. Les fonctions d’identification se base sur l’étude des intervalles au sein de la suite de note. En effet, dans le cas de *GammeDeffinisseur*, une gamme majeure (mode Ionien) de Do par exemple est caractérisée par une tierce majeur, une quarte juste et une neuvième majeur (qui revient à une seconde majeur à l’octave inférieure). Les algorithmes de définition se font donc appelle à deux fonctions différentes pour reconnaître les intervalles.

J’ai donc créé deux fonctions différentes *IntervalleBasique* et *IntervalleComplexe*. La première fonction identifie les intervalles qui ont des rapports juste (quinte, quarte et tierce). On rentre dans cette fonction la liste d’étude, l’intervalle choisie, ainsi qu’un True/False pour la position dans la suite. L’algorithme calcule le rapport entre la tonique et l’ensemble des fréquences de la liste. Si un des rapports est compris dans un intervalle précis, alors la fonction retourne « True ». *IntervalleComplexe,*  travail sur l’identification de fonction dont les rapports sont faux. Une sixte majeure par exemple équivaut à la tierce de la quarte de la tonique. La fonction *IntervalleComplexe*  fait donc appelle à la fonction *IntervalleBasique.* À partir de ces études d’intervalle il donc possible de reconnaître n’importe quelle fonction musicale.

**B** – tests, Correction et Validation

La réalisation finale est une interface, où l’on peut télécharger un fichier son en .wav, qui ressort des graphique ainsi que des la composition musicale de la suite de note. La sortie musicale s’affiche sous forme de texte, dans une notation spécifique. En effet la compréhension des sorties demandent quelques bases en théorie de la musique. Ainsi, l’outils numérique Py-Reconote, s’intégré bien pour facilité la théorique musicale pour les musicien avec quelque base. Les testes que j’ai effectués sur ma partie du programme est basé sur la reconnaissance de liste de fréquence plus ou moins complexe de longueur différentes. Ainsi la programme à passé la plus part des testes sur des liste de note juste. Cependant il ne fonctionne pas très bien lors d’identification de fichier enregistré acoustiquement. En effet les fréquences reçues peuvent être trop fausses pour être traités.

**VI –** Bilans & Perspectives

Malgré l’aspect technique, le projet reste peu abouti. Notamment sur l’aspect de la reconnaissance des natures musicales des suites notes. En effet, ce programme ne sait pas analyse les renversement, c’est à dire lors que dans un accord quelconque les notes ne sont pas rangé dans l’ordre normale (tonique, tierce, quinte), cette pratique étant très courant dans la compostions de musiques actuelle. De plus l’algorithme ne sait pas analyser une gamme décroissante. En résumé pour améliorer cette algorithme il faudrait dans un premier temps l’optimiser (réduire le nombre de commande), approfondir la bibliothèque. Il serait aussi envisageable d’écrire les notes sous la forme de portées grâce au module music21.

Alors que l’on est dans une année décisive pour notre orientation, ce projet m’a permis de découvrir le monde de l’information et du numérique. J’envisage de faire des études d’ingénieure, et notamment des écoles où l’apprentissage du code est au centre de cursus. J’ai pu aussi voir, lors des portes ouvertes, que les écoles d’ingénieures utilisent de plus en plus « le travaille par projet ». Ces travaux par projet se rapprochent de la démarche que suivit lors de l’enseignement de la spécialité ISN. J’ai pu me confronté une première fois à cette pédagogie. J’ai donc pu voir que j’ai beaucoup de chose à faire pour être plus productif. En effet, j’ai tendance à ne pas savoir me poser de limites à mon imagination, j’ai donc une tendance a me perdre dans mon flow d’idée. Ça m’a posé problème lors de création du projet, car je n’arrivais pas à me poser sur une idée fixe et m’y tenir. Je cherchais toujours à dépasser aller plus loin dans l’idée originelle. Ce qui m’a bloqué plusieurs fois dans l’avancé du projet.

**VII –** Diffusion du projet

<https://github.com/lasource2018/oscilloraptor>

Nous avons choisie la version 4.0 de Creative Common  comme licence de diffusion pour plusieurs raison. Tout d’abord, possible d’utiliser cette licence dans le cadre de projets *Open data*. Il permet de mieux contrôler juridiquement l'accessibilité des différentes parties d’un projet. Par exemple de choisir d’attribuer une même licence Creative Commons pour une publication et les jeux de données liés, ou une licence pour la publication et  une autre pour les données. De plus Créative Common ne nécessite pas d’adaptation de juridique nationale pour être applicable. L’aspect modulable de Creative Common nous largement séduite comparé aux autres licences. On pense que la licence répondra précisément à nos attentes

**VIII –** Index (code)



