

# Rapport intermédiaire Projet long

## Pour le 18 Février 2022

Chang Patrick, Du Vincent

February 16, 2022

### 1 Introduction

La musique fait partie de notre quotidien, elle nous fait plaisir, nous donne envie de danser et nous garantit une joie de vivre. Nous avons eu l'idée de l'accompagner avec une chose tout aussi présente, la lumière. En effet, notre projet consiste en une ampoule intelligente capable de changer de couleur en fonction des fréquences de la musique, capable de reconnaître une voix humaine et de changer de couleur vis-à-vis de celle-ci et enfin, capable de reconnaître le type de musique jouer. Tous ceux-ci en temps réel.

Notre équipe étant composée de 2 personnes :

- **CHANG Patrick**, étudiant en M1 IMPAIRS à l'Université de Paris
- **DU Vincent**, étudiant en M1 IMPAIRS à l'Université de Paris

### 2 Nos premiers choix

Afin de réaliser ce projet, nous avons tout d'abord concrétiser un cahier des charges reprenant ce qu'on devrait obtenir au final. Comment réussir à y parvenir. Et une première liste de classes qui nous servira de base, comme une classe **Vue** et une classe **Calcul**.

Nous avons décidé d'utiliser **Gitlab** un logiciel de gestion de données pour mieux se partager nos travaux et mieux s'organiser. Pour organiser le travail, nous avons fait une liste de tâches à réaliser par ordre de priorité.

Au début du projet, nous avons pensé à une organisation bien précise des fichiers en utilisant **Gradle**, un outil permettant la compilation et la construction de projet.

Tout d'abord un dossier **src/main/java** contenant tous les fichiers codés en **Java**. A l'intérieur de ce dossier, puisque nous n'avons pas encore d'ampoule disponible, nous avons un fichier **Vue.java** contenant toute la partie graphique du projet.. Celle-ci nous permettra de simuler une ampoule. **DBFiller.java** lie

la base de données locale et le projet. Et **Recorder.java** se charge d'enregistrer le son en temps réel. Un dossier **src/test/java** contient des tests pour quelques fonctions.

### 3 Une ampoule qui réagit à la musique

Les enregistrements audio sont généralement faits en fonction du temps. La transformée de Fourier (FFT) est une opération qui permet de représenter des signaux apériodiques en complexes c'est-à-dire avec une partie réelle et une partie imaginaire. Ces complexes peuvent eux-mêmes se traduire en fréquences (montrer calcul). Cette FFT permet de passer un signal du domaine temporel au domaine fréquentiel. Grâce à elle, nous pouvons nous donner une plage de fréquences à traduire en couleurs Par exemple les fréquences allant de :

- de 0hz à 100hz en bleu
- de 100hz à 200hz en cyan
- de 200hz à 300hz en jaune
- de 300hz à 400hz en orange
- de 400hz à 500hz en rouge

Pour ne pas capter des petits bruits indésirables tels que des froissements de papiers, nous ne gardons que les fréquences dépassant une certaine magnitude. La magnitude d'un nombre complexe est sa distance avec le point d'origine dans son plan complexe. Ici, la magnitude est aussi l'amplitude de la fréquence. Avec elle on calcule notamment les décibels enregistrés. En limitant la magnitude, nous gardons uniquement les fréquences dépassant un certain seuil de décibel.

$$magnitude = \sqrt{partie\_reelle^2 + partie\_imaginaire^2}$$

### 4 Base de données

Nous utilisons **PostgreSQL** comme système de gestion de base de données. Le fichier DataBase.sql une fois exécutée crée automatiquement une nouvelle base de données et un nouvel utilisateur sur votre ordinateur. Il contient 3 tables, une table **musics(music\_id, name)**, une table **classification(musicid, genre)** avec le genre de chaque musique et, une table **musics\_fingerPr(music\_id, fingerPr)** contenant les empreintes digitales de chaque chanson (cf. plus bas : Retrouver le genre d'une musique).

Les musiques sont classées en différents genres, allant du **soft**, **médium** jusqu'au **hard**. Nous avons réalisé une enquête pour décider où classer chaque musique, et garder les tendances sortantes.

Chaque musique est composée de plusieurs fingerprint (un peu comme les empreintes de doigts d'un humain). Pour générer ces empreintes nous lisons

chaque fichier audio contenu dans le dossier `res/music` et appliquons l'algorithme FFT sur des petits fragments de musique. Ces empreintes forment l'identité de la musique. Ensuite, nous séparons les fréquences par 5 intervalles allant de 40 Hz à 300 Hz. Cette plage de fréquences inclut la plupart des tonalités d'instruments et voix que nous retrouvons dans les musiques les plus écoutées. Pour chaque intervalle, nous ne gardons que la fréquence ayant la plus grande magnitude, c'est-à-dire le plus de décibels. Donc chaque intervalle possède une seule fréquence, ce qui nous donne **5 fréquences** par fragments de musique. Nous les stockons dans la base de données comme un type `Double`.

## 5 Retrouver le genre d'une musique

Pour retrouver le genre d'une musique que nous enregistrons en temps réel, nous appliquons l'algorithme FFT à un fragment de celle-ci puis nous cherchons dans la base de données toutes musiques partageant au moins 90% d'empreintes avec elle ; grâce à une requête `SQL`. En temps réel, s'il y a une correspondance pour un fragment nous affichons son genre sur l'interface graphique.

## 6 Les problèmes

Nous avons rencontré plusieurs difficultés jusqu'à maintenant.

1. un problème de micro, lors des tests les résultats étaient incompréhensibles.
2. Trouver des idées de méthode pour stocker une musique dans une base de données, sous quel format ?
3. Comment retrouver le genre d'une musique grâce à une base de données ?
4. Comprendre le résultat de sortie d'une FFT.
5. Retirer les bruits de grésillements, et détecter la voix.

## 7 Continuité

Nous n'avons pas encore terminé la détection de voix, celle-ci marche lorsqu'il n'y a pas de musique. Pour tester nos programmes, nous écoutons le son obtenu après l'application d'une FFT et en essayant de retirer les fréquences autres qu'une voix. Nous réalisons une normalisation du son pour retirer les bruits de grésillements, et homogénéiser les sons. Il nous faut aussi choisir de meilleurs couleurs avec une animation plus agréable.