



## Rapport de soutenance n°1

*Chef d'équipe* : Mayes Tidjedam  
Camille Poingt  
Alexandre Voinier

Promo 2024

# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Présentation du groupe</b>	<b>4</b>
2.1	Les membres du groupe . . . . .	4
2.1.1	Mayes Tidjedam . . . . .	4
2.1.2	Camille Poingt . . . . .	4
2.1.3	Alexandre Voinier . . . . .	4
<b>3</b>	<b>État de l'art</b>	<b>5</b>
3.1	Les géants de l'audio . . . . .	5
3.2	Les logiciels d'enregistrement vidéo . . . . .	6
3.3	Qu'en avons nous tiré? . . . . .	6
<b>4</b>	<b>Présentation des technologies utilisées</b>	<b>7</b>
4.1	GTK+ et Glade . . . . .	7
4.2	FMOD . . . . .	7
<b>5</b>	<b>Répartition et planning des tâches</b>	<b>8</b>
5.1	Répartition des tâches . . . . .	8
5.2	Planning des tâches . . . . .	8
5.3	Description des tâches . . . . .	9
5.4	Ressentis concernant le projet et son avancement . . . . .	9
<b>6</b>	<b>Récit de notre avancement</b>	<b>10</b>
6.1	La Librairie "FMOD" . . . . .	10
6.2	Interface graphique . . . . .	11
6.3	Spectre . . . . .	14
6.4	Enregistrement du son . . . . .	15
6.5	Lecture de la piste audio . . . . .	15
6.5.1	Initialisation de la structure MusStruct . . . . .	15
6.5.2	Le bouton Pause . . . . .	16
6.5.3	Le bouton Play . . . . .	16
6.6	Importation des fichiers audio . . . . .	17
6.6.1	Quelques vérifications... . . . .	17
6.6.2	Création du son . . . . .	18
6.6.3	Ajouts dans l'interface graphique . . . . .	18
6.7	Création d'un nouveau document . . . . .	19
6.8	Sauvegarde d'un fichier audio . . . . .	19
<b>7</b>	<b>Le site web</b>	<b>19</b>
<b>8</b>	<b>Conclusion</b>	<b>20</b>
<b>9</b>	<b>La suite de notre projet</b>	<b>21</b>

# 1 Introduction

Si pour les années précédentes nous avons eu un sujet imposé : un jeu, un OCR... cette année on nous a donné le choix du sujet avec la seule contrainte que l'algorithmique y ait une place importante.

Notre groupe a donc longuement cherché un sujet intéressant et original pour ce projet. Nous nous sommes accordés autour d'une idée: créer un outil de traitement de son.

Cette première partie de l'année a été marquée par la démission de Théo Letellier. Cette événement a amené à revoir la répartition des tâches ainsi que la planning que nous avons mis en place dans le cahier des charges. Vous allez donc pouvoir observer ce changement sur les deux tableaux présents dans la cinquième section de ce rapport. De plus, nous vous présenterons tout ce que nous avons fait jusque là.

## 2 Présentation du groupe

TMAC a été créé fin janvier 2021 dans le cadre du projet de S4. Il est composé de quatre étudiants de l'EPITA qui ont pour ambition de révolutionner le monde de l'informatique. Vous trouverez ci-dessous une brève présentation de chaque membre du groupe afin d'en apprendre un peu plus sur ce groupe très peu connu du grand public.

### 2.1 Les membres du groupe

#### 2.1.1 Mayes Tidjedam

Chef du projet TMAC studio, je suis aussi un grand fêru de musique.

La création a toujours fait parti de ma vie et c'est pourquoi je me réjouis de pouvoir faire partie intégrante de ce groupe de travail. Je perçois ce projet également comme une occasion de poursuivre le développement de compétences telles que : l'écoute, la communication et le management.

#### 2.1.2 Camille Poingt

J'ai toujours aimé travailler sur un ordinateur sans trop m'intéresser à l'informatique.

Ce n'est qu'en entrant à l'EPITA que j'ai commencé à coder, à m'intéresser à ce domaine. Je ne suis sûrement pas la meilleure, mais je compte faire tout mon possible pour parvenir à mener à bien ce projet.

#### 2.1.3 Alexandre Voinier

Futur Directeur de la sécurité chez Google, je viens me former dans cette école réputé du sud de Paris, EPITA. Je viens de la banlieue parisienne à côté de Disneyland. Avec ce projet j'espère découvrir de nouvelles facettes de l'informatique et ainsi agrandir ma palette de compétences.

## 3 État de l'art

### 3.1 Les géants de l'audio

Il existe beaucoup de vieilles applications qui permettent d'enregistrer et modifier du son. On retient souvent les géants tels que Audacity ou FL studios:



Le premier est destiné à un public amateur n'ayant donc pas forcément de connaissance avec le domaine. L'interface et l'utilisation sont simples. Cependant, les rendus sont rarement à la hauteur de nos attentes. De plus, malgré le fait qu'elle soit intuitive et ergonomique, l'interface d'Audacity est vieille visuellement parlant.

Le second quant à lui est destiné aux professionnels où à ceux qui veulent produire et enregistrer des mélodies et/ou des sons avec un rendu de qualité professionnel. L'énorme problème de cette solution, c'est qu'elle requiert énormément d'investissement pour apprendre à s'en servir correctement. Tout ceci ajouté au fait que cette solution est payante.

### 3.2 Les logiciels d'enregistrement vidéo

Il existe deux grandes familles de logiciels permettant d'enregistrer les flux vidéos d'un ordinateur. La première permet une diffusion en direct tandis que la seconde permet de sauvegarder un fichier vidéo pour le diffuser en différé. Ces logiciels permettent également d'enregistrer et/ou de retransmettre le son provenant du micro et/ou du support.



**OBS Studio**



**XSplit**

Aucun de ces services ne permet d'avoir une visualisation du son précise et aucune ne permet de faire des modifications sur ce dernier.

### 3.3 Qu'en avons nous tiré?

Le monde de l'audio sur pc n'est composé que d'extrêmes. D'un côté nous avons des outils difficiles à prendre en main mais permettant d'avoir un rendu optimal.

De l'autre nous avons un outil facile d'utilisation ne permettant pas toujours d'avoir le meilleur résultat et commence à se faire vieux (Audacity créé en 2000).

Nous souhaitons apporter de la nuance et fournir une solution alternative à l'utilisateur. Allier la facilité d'utilisation de l'un avec la qualité de rendu de l'autre. C'est donc avec un objectif clair que TMAC s'est formé et à commencer à concevoir TMAC Studio

## 4 Présentation des technologies utilisées

Afin de mener à bien notre projet, nous avons eu besoin de certaines librairies.

### 4.1 GTK+ et Glade

Dans un premier temps, nous avons utilisé la bibliothèque gtk+ et le logiciel glade pour construire une interface graphique facile à utiliser. Nous utilisons Glade car c'est un logiciel qui nous permet de facilement créer notre interface et d'effectuer des modifications sans trop de difficultés.



### 4.2 FMOD

Par ailleurs, nous avons trouvé une librairie pour réaliser toutes nos opérations sur le son : la librairie FMOD. Nous n'avons jamais utilisé cette librairie mais elle nous semble pour le moment être la plus adaptée à notre projet.



## 5 Répartition et planning des tâches

### 5.1 Répartition des tâches

	Mayes	Alexandre	Camille
Enregistrement du son			
Affichage du spectre du son			
Jouer la piste audio			
Sauvegarde des fichiers audios			
Importation de fichiers audios			
Interface graphique			
Découpage de la piste audio			
Réglage de la réverbe			
Réglage de la hauteur			
Site Web			

Légende :

	très impliqué(e)
	moyennement impliqué(e)
	très peu impliqué(e)

### 5.2 Planning des tâches

	Première	Deuxième	Troisième
Enregistrement du son	✓		
Affichage du spectre du son	40%	✓	
Jouer la piste audio	✓		
Sauvegarde des fichiers audios	30%	✓	
Importation de fichiers audios	✓		
Interface graphique	33%	66%	✓
Découpage de la piste audio		50%	✓
Réglage de la réverbe		50%	✓
Réglage de la hauteur		50%	✓



### 5.3 Description des tâches

Enregistrement du son: C'est la principale fonctionnalité de TMAC Studio. Le son pourra provenir de l'ordinateur même ou d'une source extérieure. L'objectif est de le conserver pour pouvoir ensuite l'exploiter.

Affichage du spectre du son: Ce dernier permet d'avoir une vision globale du son enregistré. Le volume et/ou la hauteur de ce dernier à chaque instant sont des données facilement déchiffrables sur un spectre de son.

Jouer un son enregistré: Cela permettra à l'utilisateur de savoir où en est son traitement du son sans avoir à sauvegarder puis jouer le fichier audio lui-même.

Sauvegarde des fichiers audios: Le rendu final après traitement devra pouvoir être stocké sur la machine et être joué.

Importation de fichiers audios: L'utilisateur pourra ainsi réutiliser des fichiers audios pré-enregistrés et/ou revenir sur un travail laissé en suspens.

Interface graphique: C'est cette dernière qui permettra à l'utilisateur de se servir confortablement de TMAC studio. Elle devra être sobre et ergonomique tout en ayant une certaine élégance.

Découpage de la piste audio: Cette fonctionnalité permettra de réorganiser la piste audio. Nous pourrions retirer une partie de cette dernière et supprimer les silences.

Réglage de la réverbération: Nous pourrions ajouter ou créer de la réverbération. Cette dernière sera modulable selon plusieurs critères (la taille de la pièce).

Réglage de la hauteur: Cet effet permettra de rendre le son plus grave ou plus aigu.

### 5.4 Ressentis concernant le projet et son avancement

Le groupe comptait à l'origine 4 membres. Nous ne sommes aujourd'hui plus que 3. Malgré ce changement, notre groupe a su rebondir et redoubler d'efficacité. Nous avons pris cet événement comme une expérience à prendre.

L'équipe est satisfaite du travail réalisé tant par sa quantité que sa qualité. C'est riche de cette sensation de satisfaction que nous commençons la deuxième étape de la conception de notre projet.

## 6 Récit de notre avancement

### 6.1 La Librairie "FMOD"

La première mission, une fois le cahier des charges validé, fut d'installer la librairie "FMOD". Il fallait faire en sorte qu'elle soit incluse dans le dossier git et facilement utilisable.

Notre première idée était simplement de créer un dossier avec la librairie et d'indiquer au compilateur (GCC) où se trouvait ce dossier avec les annotations "-L" et "-I".

Aucune erreur lors de la compilation mais lors de l'exécution, notre programme ne trouvait pas les fichiers sources de la librairie. Une deuxième idée fut, grâce au "Makefile", de copier la librairie dans le dossier "/lib". Cela permettrait au compilateur de la retrouver sans peine, cependant nous avons ensuite découvert que nous devons disposer des droits administrateurs pour appliquer cette solution.

Après des recherches plus approfondies sur le sujet et beaucoup de lecture nous avons enfin trouvé la solution. Toute cette recherche a mené à cette ligne de commande:

```
"export LD_LIBRARY_PATH=chemin/vers/notre/librairie:$LD_LIBRARY_PATH"
```

Cette commande va ajouter le chemin spécifié à la liste des chemins à utilisés pour chercher les librairies.

Cela fonctionnait enfin. La seule inconvenance est liée au chemin qui doit être renseigné. En effet celui ci est légèrement différent sur chaque ordinateur, de ce fait cette commande ne peut pas être rajouté dans le "Makefile".

## 6.2 Interface graphique

Nous avons utilisé "GTK" dans nos projets de S3, il nous a donc semblé évident que nous allions l'utiliser de nouveau dans ce projet. De plus le logiciel Glade nous permet de gagner beaucoup de temps dans l'agencement de notre interface.

Les fonctions de "CALLBACK" de "GTK" ne prennent qu'un seul paramètre. Or nous avons beaucoup de variables nécessaires à la conception de notre projet. Nous avons donc décidé de créer une structure qui nous fournira tout ce dont nous avons besoin. Cette dernière contient notamment nos variables et nos autres structures. Il nous suffit alors de donner en paramètre cette structure aux fonctions "CALLBACK" pour avoir accès à tout type de donnée. Voici les structures pour les deux écrans:

```
typedef struct SplashStruct
{
    GtkWidget *window;
}SplashStruct;

typedef struct EditScreen
{
    GtkWidget *window;
    GtkButton *rec_btn;
    GtkButton *stop_btn;
    GtkButton *play_btn;
    GtkButton *pause_btn;
    GtkButton *replay_btn;
    GtkBox *grilleG;
    GtkBox *grille;
}EditScreen;
```

Et maintenant la structure universelle:

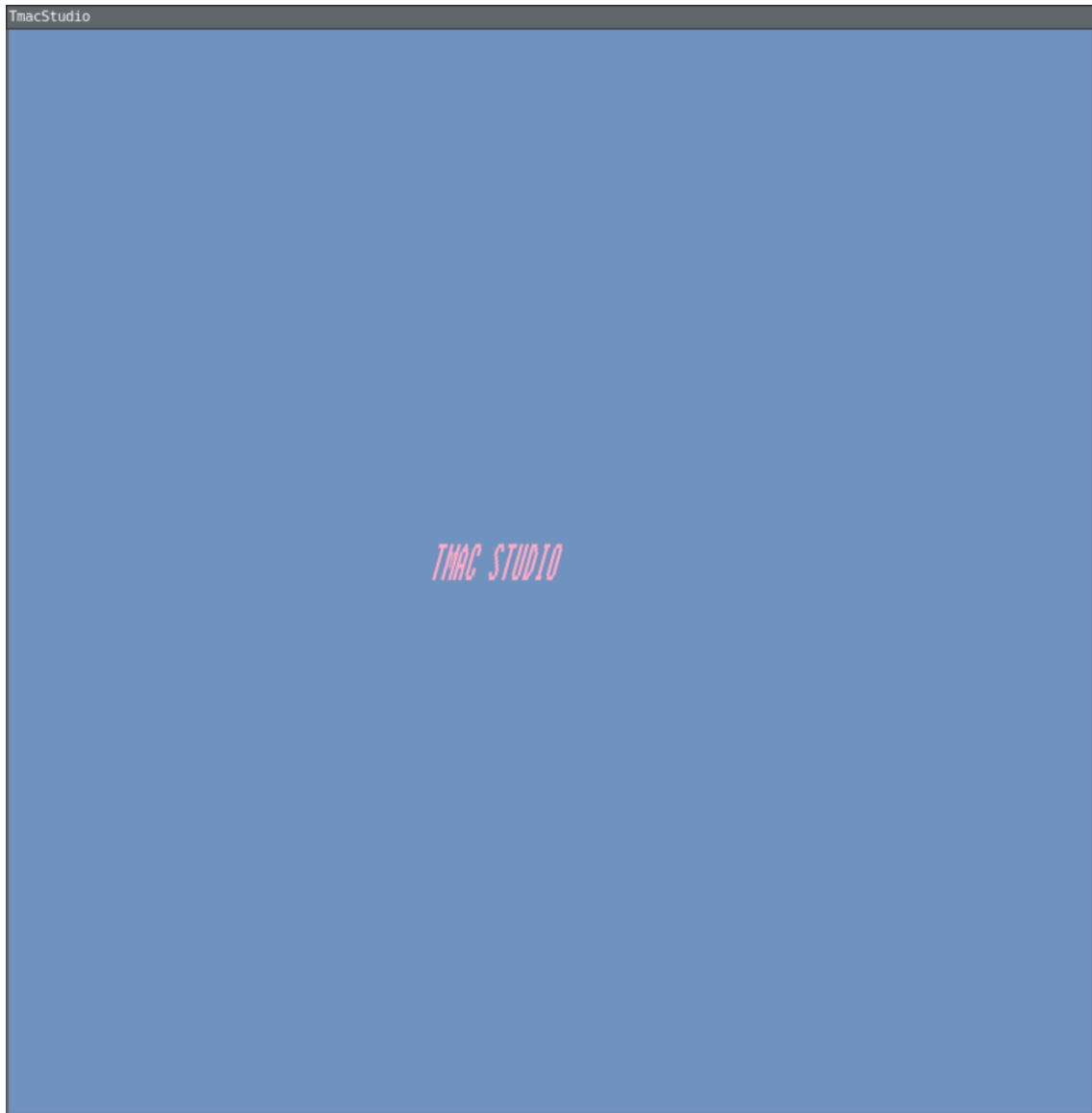
```
typedef struct Ui
{
    // Default parameters
    size_t default_width;
    size_t default_height;

    // Builder
    GtkBuilder *builder;
    guint loading;
    // Screens
    SplashStruct splash;
    EditScreen edit;

    MusStruct mus;
}Ui;
```

Pour l'interface graphique nous nous sommes inspirés de l'apparence d'Audacity. Un design simple, qui permet une bonne prise en main sans besoin de notice.

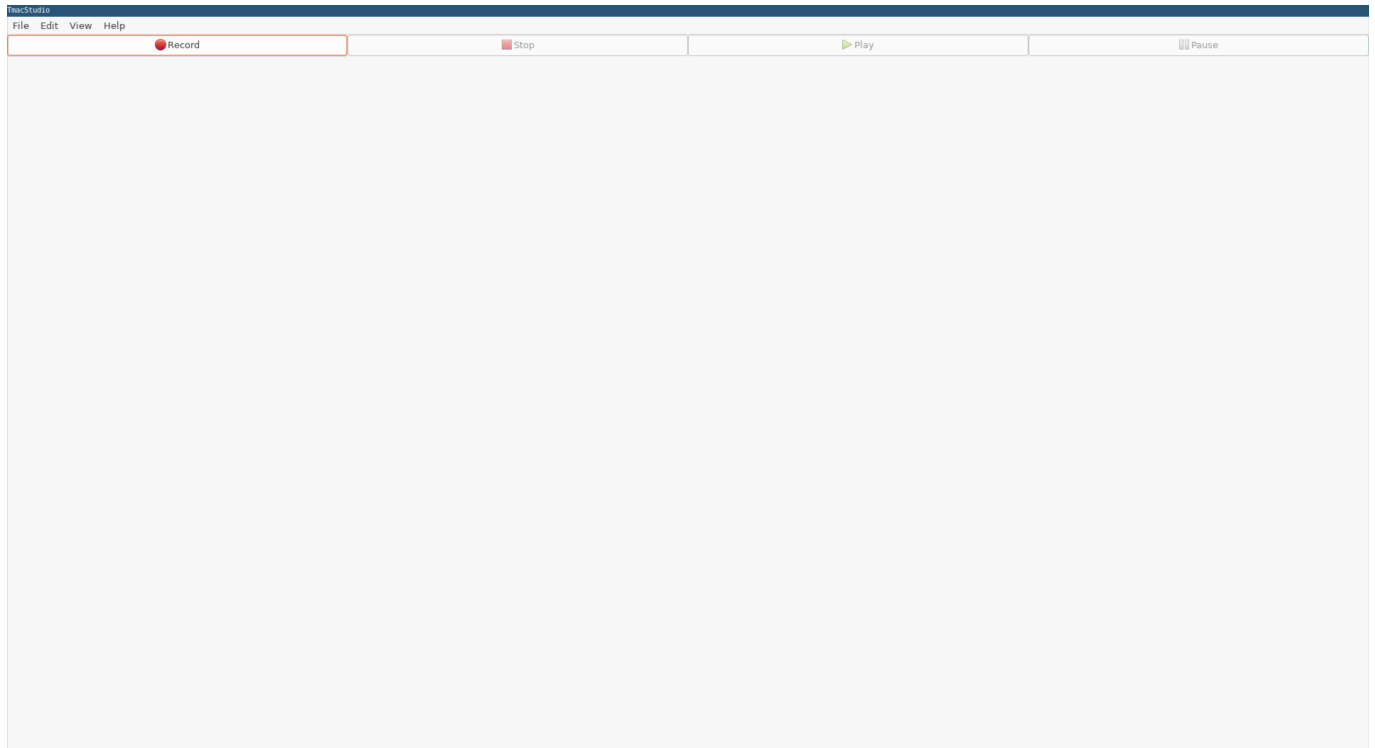
Au lancement du programme une première petite fenêtre apparaît, il s'agit d'un écran de chargement.



Ensuite la page principale apparaît.

La barre en haut de la fenêtre permet de quitter l'application, d'ouvrir un fichier audio mais aussi d'en créer un nouveau. En dessous de cette barre se trouvent quatre boutons : RECORD STOP PLAY et PAUSE. Ils sont les piliers de notre application.

Dans le futur c'est sur cette fenêtre que nous allons pouvoir observer le spectre du son mais aussi pouvoir effectuer des modifications de hauteur ou ajouter de la réverbération.

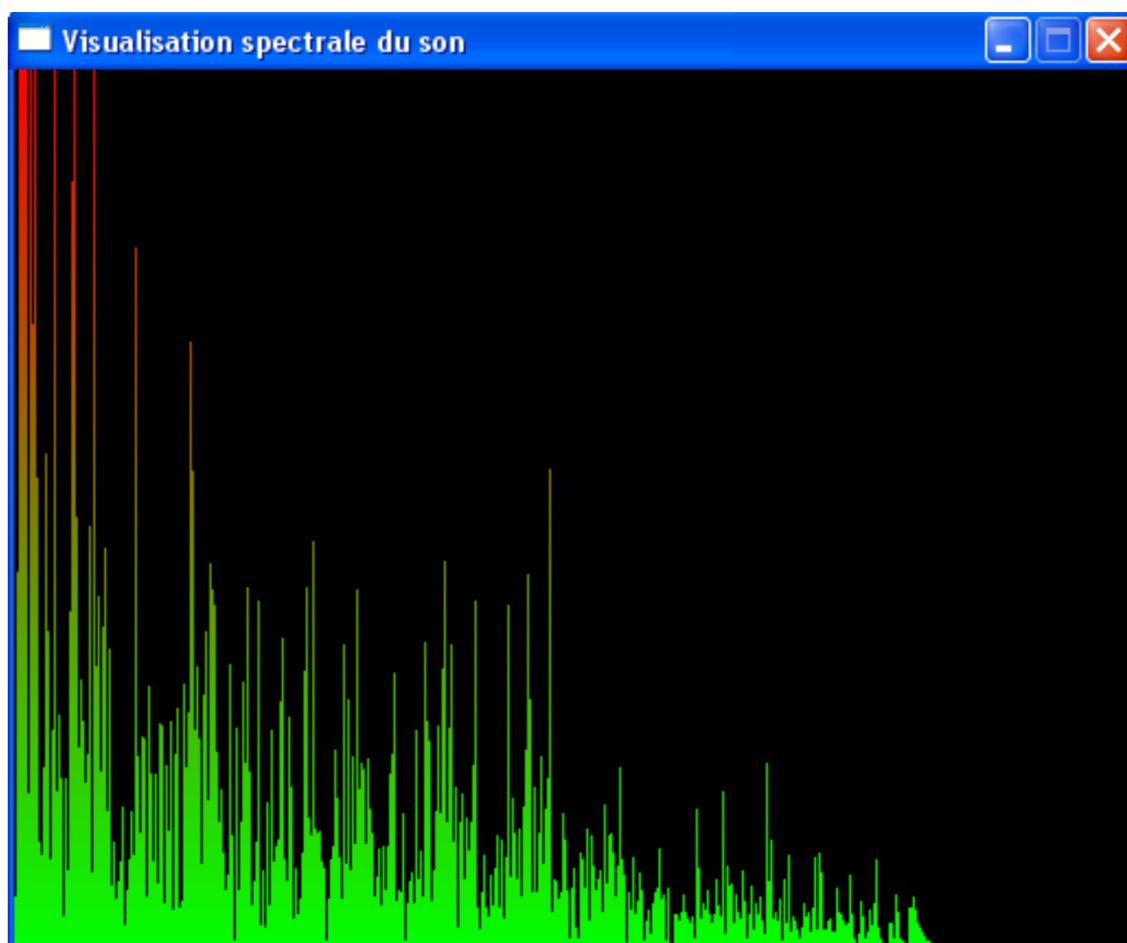


## 6.3 Spectre

Le spectre audio était une tâche programmé pour cette soutenance mais avec le départ de Théo cela n'a pas été possible. Mais nous avons pu faire des recherches sur ce qu'est un spectre et comment nous allons pouvoir l'afficher. Pour finir, nous nous sommes intéressés à la manière dont nous allons pouvoir l'exploiter via "FMOD".

Le spectre d'un son est le tableau ou la représentation graphique des fréquences qui, s'ajoutant les unes aux autres, constituent le son.

Pour l'afficher nous allons utiliser les zones de dessins de "GTK". Avec "FMOD" nous devrions pouvoir récupérer un tableau contenant les valeurs pour toutes les fréquences. Avec ces valeurs nous allons dessiner un graphique en colonnes où chaque colonne représentera une fréquence. Les fréquences ne sont pas les mêmes tout au long du son, nous allons donc mettre à jour cette affichage le plus souvent possible afin qu'il soit le plus représentatif possible. Voici à quoi devrait ressembler le spectre audio dans notre application :



Nous avons un petit peu expérimenté dans une branche de notre répo. Pour l'instant nous réussissons à créer un tableau qui contiendra les fréquences.

La vraie difficulté vient de la documentation qui n'est pas très fournie. Ceci est dû au fait que "FMOD" a plusieurs champs d'utilisation et que le notre est le moins prisé de tous. Nous sommes malgré tout confiants et espérons avoir ce visuel opérationnel pour la seconde soutenance.

## 6.4 Enregistrement du son

L'application permet d'enregistrer le son du micro de votre ordinateur. Ce son peut ensuite être joué, sauvegardé en fichier et par la suite nous pourrions le modifier.

Ainsi, lorsque l'on appuie sur le bouton d'enregistrement, une fonction est appelée afin d'effectuer les vérifications et les actions requises et les actions voulues. On doit d'abord vérifier

## 6.5 Lecture de la piste audio

Comme dans tout logiciel de gestion du son, la lecture de la piste audio est une fonctionnalité essentielle. Dans cette partie, nous vous parlerons des opérations liées à la lecture de la piste audio ainsi que de la structure utilisée pendant celle-ci.

### 6.5.1 Initialisation de la structure MusStruct

Afin d'avoir accès aux différentes variables du son, nous avons créé une structure "MusStruct". Cette structure est incluse dans la structure générale "Ui" qui contient toute les données que nous avons créé et/ou cherché.

La structure "MusStruct" contient pour la moment un pointeur vers le "FMOD\_SYSTEM", un pointeur vers un "FMOD\_SOUND", deux booléens "is\_paused" et "is\_recording" ainsi que un entier non signé "datalength".

```
typedef struct MusStruct
{
    FMOD_SYSTEM *system;
    FMOD_SOUND *musique;
    int is_paused;
    int is_recording;
    unsigned int datalength;
}MusStruct;
```

Cette structure est initialisée dans la fonction principale juste après l'initialisation de l'écran principal. Lors de son initialisation, les variables sont soit mis à 0, soit à NULL selon le type à l'exception du "FMOD\_SYSTEM" qui est créé puis initialisé à l'aide des fonctions de la librairie "FMOD : FMOD\_System\_Create()" et "FMOD\_System\_Init()". Pour la prochaine soutenance nous avons pour projet de changer la variable "FMOD\_SOUND" en liste chaînée de variables "FMOD\_SOUND" pour pouvoir charger plusieurs musiques en mémoire en même temps.

Pour finir, lors de l'initialisation de cette structure, les boutons "play", "pause" et "stop" sont désactivés et le boutons "record" est activé.

### 6.5.2 Le bouton Pause

Lorsque l'on appuie sur le bouton Pause, on va tout d'abord récupérer le "FMOD\_CHANNELGROUP" principal à l'aide de la fonction "FMOD\_System\_GetMasterChannelGroup()" puis on va vérifier l'état du booléen "is\_paused" de la structure "StructMus".

Si celui-ci est à 0, on va le mettre à 1 et activer la pause avec la fonction "FMOD\_ChannelGroup\_SetPaused()". Dans le cas contraire, on change également la valeur du booléen et on enlève la pause avec la même fonction.

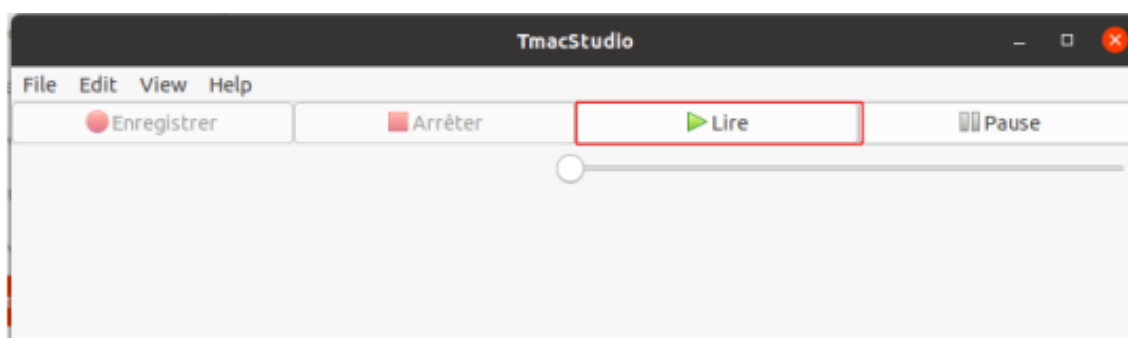


Pour le moment, on utilise des groupes de canaux étant donné que l'on a qu'un son à gérer mais pour la prochaine soutenances, nous essayerons de reproduire ces mêmes fonctions et différenciant les canaux selon les musiques.

### 6.5.3 Le bouton Play

Les actions effectuées lorsque l'on appuie sur le bouton play sont assez simples.

On va vérifier dans un premier temps si le système est en pause car si celui-ci est en pause, on ne pourra rien jouer du tout. Si c'est le cas, on désactive la pause de la même manière que dans la fonction Pause().



Enfin on va utiliser la fonction "FMOD\_System\_PlaySound()" qui va jouer le "FMOD\_SOUND" passé en paramètre depuis le début. Ainsi à tout moment, en appuyant sur ce bouton, on pourra faire rejouer le morceau depuis le début.



## 6.6 Importation des fichiers audio

A présent nous allons vous parler des étapes nécessaires pour importer un fichier audio dans notre code.



### 6.6.1 Quelques vérifications...

Dans un premier temps il est nécessaire de vérifier si il n'y a pas déjà un son dans la variable "FMOD\_SOUND" de la "MusStruct". Si cette vérification n'a pas lieu et qu'il y a déjà un son chargé, l'ancien son sera toujours là et se superposera avec le nouveau son chargé.

Ainsi si il y a déjà un son dans cette variable, il est nécessaire de libérer ce son avec la fonction "FMOD\_Sound\_Release()". Cette fonction stoppe toutes les actions en cours liées au son passé en paramètre.

Par ailleurs, il est également nécessaire de vérifier si le système utilisé est en pause. En effet s'il l'est, le nouveau son chargé sera lui aussi en pause. Par conséquent si le son est en pause alors on enlève la pause de la même manière qu'énoncé précédemment.

Ces vérifications seront aussi effectuées lorsque l'on veut enregistrer un nouveau son.

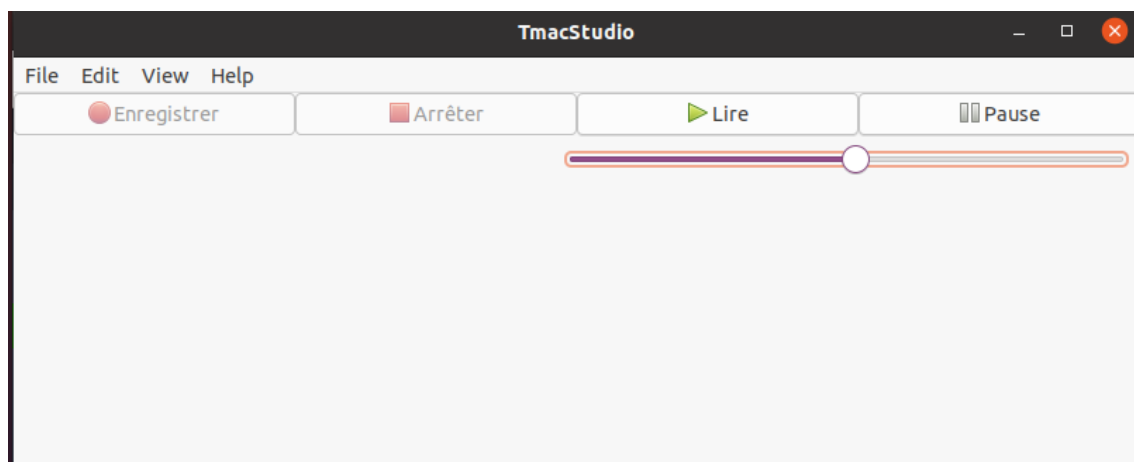
### 6.6.2 Création du son

Une fois toutes ces vérifications effectuées, on peut créer le nouveau son avec la fonction "FMOD\_System\_CreateSound()". Pour pouvoir ouvrir une musique déjà existante, il suffit de récupérer le chemin de ce fichier son avec une fenêtre de choix "GTK" et de le passer en paramètre de la fonction citée précédemment.

Ensuite on vérifie si le son a bien pu être créé et seulement si c'est le cas, on va stocker le son nouvellement créé dans la variable "FMOD\_SOUND" de la "MusStruct" et on actualise l'activation des boutons de l'interface. Par exemple, les boutons record et stop doivent être désactivés car ils ne sont plus utiles pour le moment.

### 6.6.3 Ajouts dans l'interface graphique

Une fois le son créé, une zone de dessin ainsi qu'un slider s'insère dans l'écran juste en dessous des boutons "record", "stop", "play" et "pause".



Cette zone de dessin n'est pas utilisée pour le moment mais le sera pour la deuxième soutenance lorsque nous ajouterons l'aspect graphique du son.

Par ailleurs, le slider permet de régler le volume du son. Celui-ci est initialisé à 0 au début et peut être augmenté en glissant le slider vers la droite.

Ainsi à chaque fois que la valeur du slider change, celle-ci est récupérée et utilisée dans la fonction "FMOD\_ChannelGroup\_SetVolume()" en tant que nombre flottant. Pour l'instant un seul slider peut être ajouté dans une même fenêtre puisque nous ne gérons qu'un son.

## 6.7 Création d'un nouveau document

Il est possible de créer un nouveau document à l'aide du bouton "Nouveau" situé dans la menu File en haut à gauche de l'interface. Ce bouton entraîne tout simplement une ré-initialisation de la structure Mustruct, la suppression du slider et de la zone de dessin ajouté ainsi que la gestion des boutons comme à l'ouverture du logiciel.

Ainsi une fois ce bouton utilisé, on retrouve l'interface comme on l'a découverte à l'ouverture.

## 6.8 Sauvegarde d'un fichier audio

La sauvegarde de fichier audio figure parmi les tâches qui devait être finies pour la première soutenance et que nous n'avons pas eu le temps de terminer.

Cependant nous avons quand même fait beaucoup de recherches et pour le moment, notre fonction "save" permet bien de créer un fichier de type ".WAV" et d'y mettre les informations du son à enregistrer comme par exemple sa fréquence ou encore son type. Nous pouvons même choisir le nom que nous voulons donner à ce fichier à l'aide d'une fenêtre de choix de gtk.

Nous avons essayé de récupérer les données du son au fur et à mesure que nous l'enregistrons mais cette idée entraînait une boucle infinie qui faisait planter notre application... Nous pensons qu'à la place de cette boucle, il faudrait plutôt mettre un appel régulier à une fonction externe avec un temps d'attente entre chaque appel pour que notre programme accepte mieux cette boucle.

## 7 Le site web

Le site web est proche de sa version finale. Il manque uniquement un bouton permettant de télécharger de l'application. Nous voulions un site à l'image de notre application: sobre, épuré, intuitif et élégant.

La page a été codé en "HTML5", les animation et autres visuels, quant à eux, ont été réalisés en CSS.

Il a pour but de présenter l'application et l'équipe en quelque mots.

## 8 Conclusion

En conclusion, nous sommes satisfait des tâches accomplies pour cette soutenance. En effet malgré le départ de Théo, nous avons pu produire un contenu dont nous sommes fiers. Si certaines tâches telles que l'enregistrement du son, l'importation ou encore le lecture de la piste audio ont pu être finie, d'autres sont en cours de réalisation mais nous ne désespérons pas ! Nous prévoyons beaucoup d'améliorations pour la prochaine soutenance telles que la gestion de plusieurs sons à la fois ou encore l'affichage du spectre du son et la sauvegarde de fichiers audio.

## 9 La suite de notre projet

Le travail accompli jusqu'ici nous a permis de prendre en main la bibliothèque fmod et de mettre en place notre zone de travail. Produire une base visuelle était aussi nécessaire. Avec tous ces éléments, nous sommes prêts à nous plonger dans des tâches plus détaillées.

La suite sera rythmée par la conception de l'aspect graphique de l'application et l'implémentation d'outils variés permettant d'exploiter le son.