

2016-2017

**Réaliser par :**

* Bouchti NORDINE
* LAMINE Marouane
* phily TOM
* rostan ESTELLE
* Traoré AICHATOU

Encadrant : Fabrice huet

KARAOKE INVERSEE

Rapport Projet D’année

M1 MIAGE |DE NICE SOPHIA ANTIPOLIS

Introduction

Le projet d’année en Master 1 MIAGE est important. Le choix du sujet l’est aussi. Outre le fait que le projet constitue une note importante pour notre moyenne du semestre, ce projet peut être un plus pour notre avenir professionnel. Le choix du sujet a été rapide, comme une évidence pour nous. Nous avons choisi de réaliser une application de karaoké inversée à l’aide de l’API **WEB AUDIO**. De notre point de vue ce projet présente deux caractéristiques fondamentales : d’une part la nouveauté de l’API pour nous et d’autre part en tant que futurs Miagistes, élargir nos champs de compétences avec cette API.

Nous allons voir à travers ce rapport dans une première partie la présentation du projet ainsi que ses principaux objectifs et dans une seconde partie, Quelles méthodes avons-nous utilisées afin de mettre en œuvre le projet ? Le travail accompli et enfin dans une dernière partie nous verrons la gestion du projet, les résultats obtenus, ainsi que les évolutions possibles de l’application.

Table des matières

[I. Présentation du sujet 3](#_Toc483347614)

[1. Contexte 3](#_Toc483347615)

[2. Taches planifiées 3](#_Toc483347616)

[3. Taches existant 3](#_Toc483347617)

[II. Travail accompli 4](#_Toc483347618)

[1. Choix techniques 4](#_Toc483347619)

[2. Taches effectuées **Erreur ! Signet non défini.**](#_Toc483347620)

[3. Résultat livré au client 5](#_Toc483347621)

[III. Gestion de projet 5](#_Toc483347622)

[1. Outils utilisés 5](#_Toc483347623)

[2. Organisation des taches dans le groupe 6](#_Toc483347624)

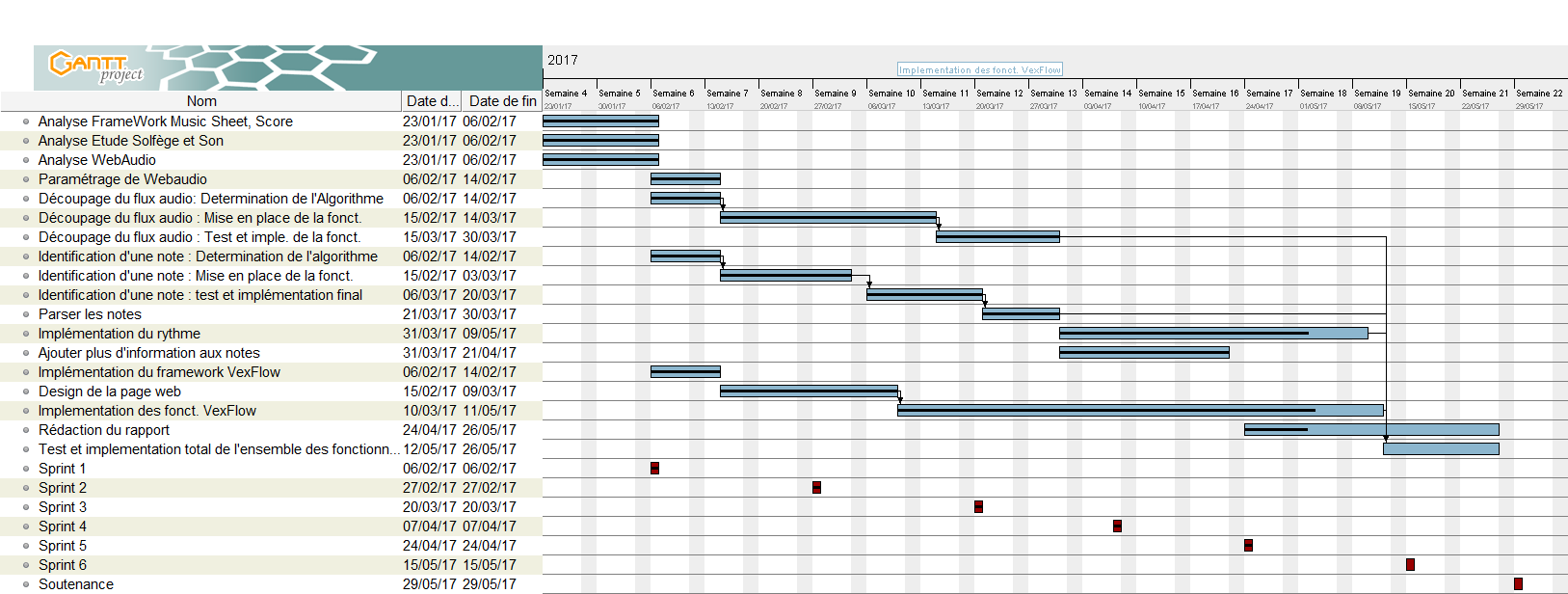
[IV. Conclusion 6](#_Toc483347625)

# Présentation du sujet

## Contexte

Dans le cadre de notre projet d’année nous nous intéressons à développer une application de karaoké inversée qui à partir d'une note de musique jouée ou un fichier audio, analyse les fréquences et affiche les notes correspondantes sur une partition à l'écran. Le fonctionnement est donc globalement celui d'un accordeur. Pour que le projet soit très XXIème siècle, Nous avons choisi de le réaliser avec l’API WEB AUDIO, HTML, JAVA SCRIPT, CSS, BOOSTRAP.

## Taches planifiées



## Taches existant

# Travail accompli

## Choix techniques

Nous avons réalisé l’application karaoké inversé avec les technologies suivants :

* Traitement du son audio

**Capture de l’audio en HTML5**

Pendant de nombreuses années, nous étions obligés de compter sur des plugins de navigateur tels que Flash ou Silverlight pour capter le son. La montée de l’HTML5 a apporté une transition remarquable en puissance de l’accès au matériel du périphérique. La géolocalisation (GPS), L'API d'orientation et l'API Web Audio (matériel audio), sont des exemples de montée. Ces fonctionnalités sont extrêmement puissantes, révélant des API JavaScript de haut niveau qui s'ajoutent aux fonctionnalités matérielles principales du système.

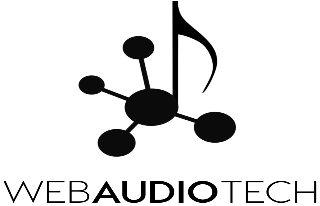
Notre recherche dans la nouvelle API est focalisé sur navigator.getUserMedia(), qui permet aux applications web d’accéder à camera et microphone de l’utilisateur.

* *getUserMedia()*

Le rythme de développement pour trouver une API de capture plus adaptée a accéléré grâce à l’effort WebRTC (Web Real Time Communication). Cette spécification est supervisée par le groupe de W3C WebRTC. GetUserMedia () est lié à WebRTC car elle est la passerelle pour cet ensemble d'API. Il fournit les moyens d'accéder au flux local de l'utilisateur notamment le microphone et la caméra.

**Support des navigateurs :**

getUserMedia() a été pris en charge depuis Chrome 21, Opera 18, et Firefox 17.



La Web Audio API propose un système puissant et flexible pour contrôler les données audio sur internet. Elle permet notamment de sélectionner des sources audio (microphone, flux media), d'y ajouter des effets, de créer des visualisations, d'appliquer des effets de spatialisation (comme la balance), etc.

* Structuration de l’interface de l’application



* Mise en forme de l’interface de l’application

* Affichage des notes de musique identifiées

## Projets d’affichage d’une partition en HTML5 / JavaScript

* VexFlow (<http://www.vexflow.com/>)
* alphaTab ([http://www.alphatab.net](http://www.alphatab.net/))
* abcjs (<https://github.com/paulrosen/abcjs>)
* abcweb (<http://wim.vree.org/js/>)

## Formats de notation musicale :

* MusicXML ([https://www.musicxml.com](https://www.musicxml.com/))
* ABC ([http://abcnotation.com](http://abcnotation.com/))

## Comparatif des projets :

|  | **[VexFlow] (**[**http://www.vexflow.com/**](http://www.vexflow.com/)**)** | **[alphaTab] (**[**http://www.alphatab.net**](http://www.alphatab.net/)**)** | **[abcjs] (**[**https://github.com/paulrosen/abcjs**](https://github.com/paulrosen/abcjs)**)** | **[abcweb] (**[**http://wim.vree.org/js/**](http://wim.vree.org/js/)**)** |
| --- | --- | --- | --- | --- |
| Formats d'entrée | VexTab | AlphaTex markup | ABC | ABC ou MusicXML |
| Documentation / communauté | [VexFlow wiki de 19 pages] (<https://github.com/0xfe/vexflow/wiki>) et [tutoriel] (<https://github.com/0xfe/vexflow/wiki/The-VexFlow-Tutorial>) | [Documentation en ligne sommaire] (<http://www.alphatab.net/documentation/>) et [API] (<http://api.alphatab.net/alphatab/index.html>) | [API] (<https://github.com/paulrosen/abcjs/blob/master/api.md>) | [documentation] (<http://wim.vree.org/js/readme.html>) |

## Synthèse

D'une part, on souhaite travailler avec un format standard pour la notation de musique. D'autre part, on souhaite pouvoir générer ce format à l'aide d'un framework. Quel est le framework le plus adapté à notre projet ?

Les formats MusicXML et ABC sont des formats standards mais il n'existe pas de librairie en JavaScript pour les générer. Vexflow propose un format ouvert, même si non standard. On peut le générer à l'aide de JavaScript. C'est pour cela que nous avons choisit d'utiliser Vexflow comme framework pour notre projet.



VexFlow est une API de rendu de notation de musique en ligne open source. Il est entièrement écrit en JavaScript, et fonctionne correctement dans le navigateur. VexFlow prend en charge HTML5 Canvas et SVG.

* Algorithme pitch détection

Dans cette partie, on a décidé quel algorithme qu’on va utiliser dans notre application, c’est l’algorithme pitch détection, Le but de ce détecteur est de trouver la fréquence fondamentale d'un signal.

C’est un algorithme conçu pour estimer le pitch ou la fréquence fondamentale d'un signal quasi- périodique ou oscillant, généralement un enregistrement numérique de la parole ou une note musicale ou une tonalité. Cela peut se faire dans le domaine temporel ou le domaine fréquentiel ou les deux domaines. Nous l’avons utilisé pour estimer un pitch de signaux dans une gamme de fréquences, et d’analysé des fréquences des sons audio.

**Pourquoi pitch detect ? :**

Les techniques d'estimation par la fréquence fondamentale peuvent être séparées en deux catégories :

* Des estimateurs basés sur le domaine de temps dont on observe la forme du flux de son entrant.
* Des estimateurs basés sur le domaine de fréquence dont on observe le Spectrum de la fréquence.

On trouve aussi des méthodes qui combinent les deux techniques.

On outre, ça nous permet d'obtenir une fréquence fondamentale en traitant le spectre de la parole directement.

**Les informations extraites par l’algorithme :**

***La note :*** on récupère en notation (abc/sol) les symboles permettant de représenter un fragment de musique par une convention d'écriture de la hauteur et de la durée d'un son.

***Le cent :*** il permet d'indiquer avec précisions les petits écarts entre la hauteur d'une note particulière et une référence donnée. Voir : <https://en.wikipedia.org/wiki/Cent_(music)>

***La durée :*** d'un son musical est le laps de temps pendant lequel on peut identifier ses vibrations (les sons pouvant évidemment durer plus ou moins longtemps).

**Les avantages et les inconvénients :**

La complexité d’évaluation du Fondamental est une tâche difficile pour de nombreuses raisons telles que la non-stationnarité du signal vocal, une certaines irrégularités dans l’excitation glottique ou encore une interaction avec le premier formant, la décision du voisement, la distinction entre les segments non voisée et les segments voisée à énergie réduite, pour cela, l’algorithme pitch détection a des avantages et des inconvénients tels que :

* Avantages :
* L’estimation de la fréquence fondamentale avec des calculs très simples.
* Peu couteuse en temps de calcul car il nécessite peu d’opérations arithmétiques de multiplications et d’additions.
* Inconvénients :
* Des erreurs qui affects pratiquement.
* Manque de précision.

**Description de la technique :**

Dans l’algorithmes d’extraction de la F0, trois phases essentielles durant le traitement s’impliquent : le prétraitement, le traitement et le pos traitement.

La phase de prétraitement est réservée à la préparation du signal issue d’un microphone. Elle consiste à choisir la durée des trames d’analyse et du recouvrement afin de moins compromettre la condition de stationnarité exigée par l’algorithmes de traitement.

La phase de traitement est réservée à l’extraction de la fréquence fondamentale et dépend donc de l’algorithme utilisé.

La phase de post traitement a pour but de diminuer les erreurs qui peuvent être de plusieurs types.

**En bref :**

Les algorithmes de détection de hauteur peuvent être divisés en méthodes qui fonctionnent dans le domaine temporel, le domaine fréquentiel ou les deux. Un groupe de méthodes de détection de hauteur utilise la détection et la synchronisation d'une fonction de domaine temporel. D'autres méthodes de domaine temporel utilisent des fonctions d'autocorrélation ou des normes de différence pour détecter la similitude entre la forme d'onde et une version temporelle de la même.

* Identification des notes de musique :

Pour pouvoir détecter les notes de musique nous nous sommes basés sur un tableau représentant la variation des différentes fréquences de chaque note.

Nous sommes partis sur un algorithme d’estimation dont le but est de faire correspondre la note en fonction de la gamme tempérée représenté dans le tableau.

Il est implémenté de la manière suivante :

-Nous calculons une fréquence minimum et une fréquence maximum à l’aide de la formule suivante : [(freqBase-0.1) \*2^(octave+1) ; (freqBase+0.1\*2^(octave+1)].

La fréquence de base est égale à la fréquence de la note à l’octave -1.

Nous récupérons l’octave en fonction de la fréquence récupérée.

-Nous vérifions ensuite si la fréquence récupérée à partir de l’algorithme pitch detect appartient à l’intervalle fréquence minimum et fréquence maximum.

Cet intervalle dépend d’une variable = 0.1 choisie par défaut afin de filtrer au mieux les fréquences les plus éloignées d'une note.

- la fréquence récupérée correspond à la note présente dans cet intervalle si elle y appartient.

-Après avoir identifié une note nous lui attribuons le format Américain ABC utilisé par vexflow.

## Problèmes techniques rencontrés

## Résolutions des problèmes

## Résultat livré au client

# Gestion de projet

## Outils utilisés

* Gestion des versions



* Définition des taches



## Organisation des taches dans le groupe

|  |  |
| --- | --- |
| Taches | Contributeurs |
| Analyse Web Audio | Traoré & Marouane |
| Analyse son & solfèges | Phily & Bouchti |
| Analyse Framework score HTML5 | Rostan |
| Paramétrage de web Audio | Traoré |
| Découpage du flux audio | Marouane & Bouchti |
| Identification d’une note | Traoré & Phily |
| Implémentation du Framework Vexflox | Rostan |
| Design de la page | Traoré |

3. Différence entre planning initial et final

# Conclusion