



**R 5.A.06**

**2023 - 2024**

# **Sensibilisation à la programmation multimédia**

## **TP n° 1 Programmation du son**



**ANNE Jean-François**

# Sensibilisation à la programmation multimédia

## Programmation du son

Le but de ce TP est de se familiariser avec la programmation Python en audio.

### A. Exercices de base :

#### 1°) Exercice 1

Un fichier audio WAV (PCM non compressé) possède les propriétés suivantes : stéréo, 16 bits, 44100 Hz, taille du fichier 97896 octets. Recherchez :

- 1) Quelle est la taille de l'en-tête (en octets) ?
- 2) Quelle est la taille des données (en octets) ?
- 3) Quelle est la durée du son de ce fichier (en ms) ?

#### 2°) Exercice 2

Afficher en python les caractéristiques d'un fichier son fourni.

■ <http://prof-tc.fr/Lycees/file/Terminale%20S/Specifique/Fichiers/P1C2/Piano%20La3.wav>

#### 3°) Exercice 3

Créer un programme qui joue les notes sur une octave

#### 4°) Bruit blanc

Ecrire un programme python qui génère un fichier .wav qui contient 10s de bruit blanc, qui affiche son courbe en fonction du temps et qui affiche son spectre.

#### 5°) Encodeur de Code DTMF (dual-tone multi-frequency)

Ecrire un programme python qui génère et sauvegarde un fichier .wav qui contient les tonalités DTMF d'un numéro de téléphone à 10 chiffres, et qui affiche sa courbe et son spectre.

#### 6°) Echo acoustique

Ecrire un programme python qui lit un fichier .wav, ajoute de l'écho, le joue sur le pc et le sauvegarde dans un fichier .wav.

■ <https://lasonotheque.org/detail-0477-cri-wilhelm.html>

### B. Battements

#### 1°) Exercice 1 "Battements"

Créer un programme python qui génère deux signaux différents : un signal de fréquence  $f = 440$  Hz et  $f_2 = 444$  Hz. Et faire la somme dans une nouvelle variable. On obtient la

### *Sensibilisation à la programmation multimédia*

superposition de deux signaux sinusoïdaux de fréquence  $f = 440$  Hz et  $f_2 = 444$  Hz (on verra que c'est une différence de  $1/6$  de demi-ton, à la limite du perceptible).

- Vérifier que l'on entend des battements
- Déterminez la fréquence des battements :

## **C. MP3 :**

### **1°) Exercice 1 :**

Ecrire un programme python qui lit un fichier .mp3, qui affiche les caractéristiques du fichier MP3 :

**Titre, Artiste, Album, Durée, Taille du fichier, Bitrate, Echantillonnage, Canaux audios, Format audio, Taux de compression.**

Vous pourrez après modifier votre programme pour qu'il modifie quelques tags ID de votre MP3.

- <https://audiohub.fr/musique-libre-de-droits-gratuite>

### **2°) Détermination la perte due à la compression MP3 d'un fichier WAV**

Vous prendrez un fichier wav de la qualité d'un CD audio que vous compresserez en MP3 avec un débit inférieur ou égal à 128k/s. Puis vous afficherez le spectrogramme du fichier wav, du fichier MP3, et de la différence entre le fichier wav et MP3.

Vous pourrez réaffichez le spectrogramme de différence pour plusieurs débits de MP3.

- <https://universal-soundbank.com/>

## **D. Pour aller plus loin : Son Multicanal**

### **1°) Exercice 1 :**

Ecrire un programme python qui crée un fichier .wav, qui contient un son multicanal (7.1) et qui le joue sur le PC.

### **2°) Exercice 2 :**

Créer un programme python qui prend 8 fichiers wav : flute, clarinette, piano, guitare, trompette, violon, harpe et qui les diffuse individuellement sur chacun des 7 canaux du 7.1 et le fichier batterie sur le canal LFE .1

- <https://universal-soundbank.com/>

## **E. Analyse :**

### **1°) Analyse spectrale du chant d'une baleine bleue :**

- Chargez l'enregistrement du chant d'une baleine bleue
- Ecouter le chant de la baleine.
- Relevez la fréquence d'échantillonnage.
- Visualisez les 20 secondes enregistrées et affichez le spectre du chant de baleine.

### *Sensibilisation à la programmation multimédia*

On rappelle que la bande de fréquences audibles par l'oreille humaine s'étale de 20 Hz à 20 kHz. Lorsque la fréquence est faible, le son est grave (de 20 à 200 Hz). On parle de son médium pour une fréquence comprise entre 200 et 1000 Hz et de son aigu lorsque la fréquence est comprise entre 1000 et 15000 Hz.

- Relevez la plage de fréquences et en déduire le type de son produit par le chant d'une baleine.

Il est possible de restreindre l'analyse d'un signal sur une durée d'observation donnée.

- Réalisez l'analyse du signal sur la fenêtre de 0 à 5s permettant de se focaliser sur le premier morceau du chant. Observez les harmoniques principales de ce premier son émis par la baleine.
- Faites glisser la fenêtre pour observer les caractéristiques des 3 autres sons émis par la baleine.

## 2°) Détermination d'un numéro de téléphone par analyse spectrale

Un code DTMF (dual-tone multi-frequency) ou FV (Fréquences Vocales) est une combinaison de fréquences utilisée pour la téléphonie filaire traditionnelle utilisée sur le Réseau téléphonique commuté (RTC) (sauf voix sur IP). Ces codes sont émis lors de l'appui sur une touche du clavier téléphonique, et sont utilisés pour la composition des numéros de téléphones et de fax.

Techniquement, chaque touche d'un téléphone correspond à un couple de deux fréquences audibles qui sont émises simultanément. On s'intéresse ici au code DTMF utilisé aux Etats-Unis qui utilise sept fréquences bien distinctes qui permettent de coder les douze touches du clavier téléphonique représenté sur la figure 3.1. Ces sept fréquences sont visibles sur le bord gauche et sur le bas de la figure 3.1.

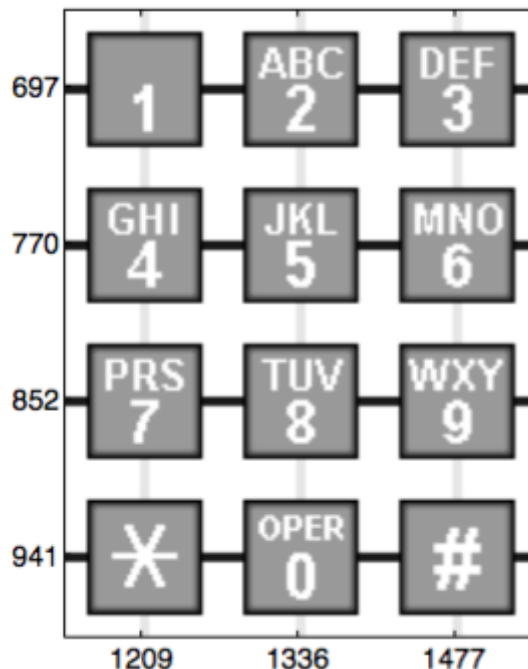


Figure 3.1 - Diapason

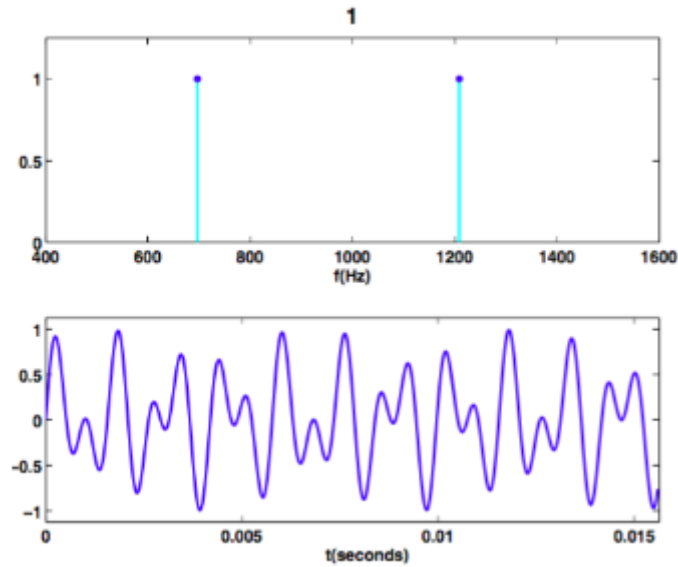
Elles sont également précisées dans le tableau ci-dessous.

697 Hz	770 Hz	852 Hz	941 Hz	1209 Hz	1336 Hz	1477 Hz
--------	--------	--------	--------	---------	---------	---------

Le son généré par l'appui d'une touche résulte de la somme de deux sinusoïdes aux fréquences associées. Ainsi l'appui de la touche 1 produira le signal suivant :

$$y_1 = \frac{1}{2} (\sin(2\pi \times 697 \times t) + \sin(2\pi \times 1209 \times t));$$

La figure 3.2 représente le signal et son spectre d'amplitude lorsqu'on appuie sur la touche "1" du clavier. Les deux fréquences peuvent être clairement identifiées à partir du spectre.



**Figure 3.2 – Evolution temporelle et spectre du signal résultant de l'appui de la touche 1 du clavier**

## **F. Webographie**

- [Documentation sur le module wave](#)
- [WAVEform audio file format](#)
- [PCM \(Pulse Code Modulation\)](#)
- [Site officiel du logiciel libre et gratuit Audacity](#)
- [Lecteur multimédia](#)
- [Editeur hexadécimal](#)
- [Code DTMF](#)
- [Module pygame.mixer \(documentation\)](#)
- [http://nsinfo.yo.fr/cours\\_python/python\\_15\\_son.html](http://nsinfo.yo.fr/cours_python/python_15_son.html)
- [http://nsinfo.yo.fr/cours\\_python/python\\_10\\_images.html](http://nsinfo.yo.fr/cours_python/python_10_images.html)
- <http://pcpagnol.free.fr/CRIADO/acoustique/gamme/note.htm>
- [https://phet.colorado.edu/sims/html/waves-intro/latest/waves-intro\\_fr.html](https://phet.colorado.edu/sims/html/waves-intro/latest/waves-intro_fr.html)
- <https://f2school.com/wp-content/uploads/2019/10/Traitement-du-signal-exercices-03.pdf>
- <https://dept-info.labri.fr/~hanna/ImageSon/>
- <https://www.cimat.mx/~moralesjh/html/ballena1.html>
- [https://github.com/R00ney/matlab\\_ngohara/blob/master/tutorials/resources/bluewhale.au](https://github.com/R00ney/matlab_ngohara/blob/master/tutorials/resources/bluewhale.au)
- <https://fr.mathworks.com/help/signal/ug/extract-regions-of-interest-from-whale-song.html>
- <https://audiohub.fr/musique-libre-de-droits-gratuite>
-