Cahier des charges projet algo

**Objectif :** Réaliser un accordeur pouvant détecter une note ou un accord joué par un instrument. L’accordeur est aussi capable de dire si un accord est mineur, majeur ou non reconnu.

**Principe du programme :**

L’utilisateur a le choix entre jouer une note ou un accord.

-Note : L’utilisateur joue une note qui est captée par le micro, puis décomposée en série de Fourier afin de trouver sa fréquence fondamentale. Puis on compare cette fréquence aux fréquences des notes de la gamme. L’accordeur nous dit quelle est la note la plus proche et si la note jouée est trop haute ou trop basse.

-Accord : On demande à l’utilisateur de jouer les trois notes d’un accord ( à l’état fondamentale). Une fois les 3 notes reconnues (même système que pour l’accordeur) on calcule l’intervalle entre les notes et détermine si c’est un accord mineur/majeur ou non reconnu.

(a) le cahier des charges de votre programme (quelles fonctionnalités sont proposées à l’utilisateur?)

(b) la description du problème posé (quelles sont les problèmes envisagées?)

(c) le principe de l’algorithme (comment les problèmes rencontrées sont ils résolus par le programme?)

(d) la bibliographie (références des documents utilisés) (e) la structuration du programme et des données (explications sur les structures de données utilisées). (f) des suggestions d’améliorations de votre projet, bugs connus

(g) le carnet de route et échéancier décrivant l’évolution de votre travail au fil des semaines (h) le pourcentage d’implication de chaque membre de l’équipe.

* **Calcule de la note Fondamentale**

1. Capturer le Flux Audio :
   1. Utilisation de bibliothèque Java : javax.sound
   2. Utilisation du type DataLine
   3. Capture le flux Audio (DataLine) dans un tableau Byte
2. Transformer le signal continue périodique en matrice numérique
   1. Appliquer un filtre de Gauss : convolution avec le signal capturé (reduction des aberrations (fuite spectrale)
   2. On obtient une matrice numérique filtrée
   3. Calcul de la FTT (methode Cooley-Tukey)
   4. On recherche la valeur Max
   5. On identifie l’amplitude Max et on détermine la fréquence associée
   6. On calcule la Note associée à la fréquence

**EcouteMusique (interface)**

onNewNote(**Son**)

**Son (extends Note)**

Double notes (tableau des Fréquences)

String nom (tableau des notes associées aux fréquences)

Int indexNote (l’index de la note)

Int octave (l’octave de la note)

Int ecart (ecart entre la fréquence capturée et la référence)

Double intensity (l’intensité capturée)

Double frequency (la fréquence capturée)

**Note**

Double notes (tableau des Fréquences)

String nom (tableau des notes associées aux fréquences)

Int indexNote (l’index de la note)

Int octave (l’octave de la note)

Int ecart (ecart entre la fréquence capturée et la référence)

Start extends JFrame

Init – SystemAudio

Ouvre

Menu extends JFrame

ActionPerformed

Bouton « Accordeur

Bouton « Accords »

MenuPrincipal extends Menu

Bouton « Accordeur

Bouton « Accords »

Bouton « Accordeur Bouton « Accords »

Ouvre la fenêtre, écoute et affiche

PanelCaptureSon extends JFrame

PanelCaptureAccord extends JFrame

**Calcule de la note Fondamentale**

CaptureSon **implements** Runnable