OBJECTIF DU PROJET

Ce projet consiste à créer et programmer une interface graphique sous MATLAB capable de synthétiser de la musique à partir d'un ensemble de notes et de paramètres réglables, autrement dit un synthétiseur !

Le projet est divisé en deux grandes parties : une partie « Travaux préliminaires » qui vous aidera à construire les fonctions de base du synthétiseur, et la deuxième partie qui les intégrera dans votre interface graphique.

Vous pouvez dès maintenant vous organiser en équipes de 2 à 3 élèves, sachant que l'exigence (particulièrement de la deuxième partie) sera pondérée selon le nombre d'élèves ingénieurs par groupe (3 cerveaux devant être plus productifs que 2 !).

Première partie : travaux préliminaires

A. Un générateur de signal basé sur les séries de Fourier

Dans de nombreuses applications de traitement du signal, il est nécessaire de générer une variété de formes d'onde périodiques avec une fréquence fondamentale programmable. Une façon d'y parvenir est de stocker les coefficients de la série de Fourier des formes d'onde souhaitées. Soient $\{a_k\}$ les coefficients de la série de Fourier de la forme d'onde souhaitée. Supposons que nous voulions produire un signal discret, ie un vecteur $x[n]$, échantillonné à f_s = 8000 Hz, dont la fréquence fondamentale, f_0, est comprise entre 100 Hz et 1000 Hz. Une solution serait de calculer :

$$x[n] = \sum_{k=-K}^{K} a_k e^{ik\left(\frac{2\pi f_0}{f_s}\right)n}$$

en prenant une valeur suffisamment grande de K pour être suffisamment représentatif (troncature de la série de Fourier).

Soient a_k les coefficients de la série de Fourier d'une onde carrée périodique (créneau) de période $T = 1$, définie sur $]-\frac{T}{2};\frac{T}{2}[$ par :

$$x(t) = \begin{cases} 1, & si\ |t| \leq 1/4 \\ -1, & sinon \end{cases}$$

Soient b_k les coefficients de la série de Fourier d'une onde triangulaire périodique de période $T = 1$, définie sur $]-\frac{T}{2};\frac{T}{2}[$ par :

$$x(t) = \begin{cases} 1 - 4t, & si\ t \geq 0 \\ 1 + 4t, & sinon \end{cases}$$

1) Déterminer par le calcul la valeur des coefficients a_k et b_k en fonction de k

2) Choisissons K = 19, f_s = 8000, f_0 = 210, et faisons en sorte de calculer $x[n]$ pour n allant de 0 à 2 f_s (autrement dit deux périodes, ici 2s, de signal). Dans Matlab, cela revient à partir d'un vecteur `n=0:(2*fs)` . Programmer une fonction dans un fichier *vecteurFourier.m* permettant d'obtenir le vecteur x à partir des différents paramètres. On choisira la syntaxe `vecteurFourier(K,fs,f0,n,forme)` qui permettra d'obtenir le vecteur x par `x=vecteurFourier(K,fs,f0,n,forme)`. `forme` prendra la valeur 1 pour une onde carrée et 2 pour une onde triangle. Pour jouer le vecteur ainsi obtenu, on utilisera l'instruction `soundsc(real(x),fs)`. Décrire l'effet obtenu dans les deux cas.

3) Utiliser la fonction précédente en changeant la valeur de `f0` à 420 (Hz) et en jouant le son correspondant. Que se passe-t-il ?

4) Résoudre le problème en changeant également la valeur de `K` jusqu'à obtenir un son satisfaisant, puis proposer une règle permettant de choisir `K` en fonction de `f0`. Créer une nouveau fichier fonction *vecteurFourier2.m* de syntaxe `vecteurFourier2(fs,f0,n,forme)` prenant en compte cette valeur adaptée.

B. De la note à la musique

Pour générer de la musique avec Matlab, il faut séquencer un ensemble de notes. Dans cette section, vous allez écrire un court programme qui utilise le générateur de signaux de la première partie pour jouer la comptine anglo-saxonne "Mary had a little lamb".

1) Pour jouer de la musique, nous devons d'abord générer les fréquences de notre gamme. La gamme de douze tons commençant à 440 Hz (correspondant au la du diapason) est donnée par `gamme440=440*2.^((0:12)/12)` et les fréquences en « la majeur » sont données par le vecteur `la_majeur=gamme440([1 3 5 6 8 10 12 13])`. Lister chacune de ces 8 fréquences et les écouter à l'aide de la première partie.

2) Pour les notes de "Mary had a little lamb", l'indice (relatif à la gamme majeure de la question précédente) est donné par :
`notes_mary=[3 2 1 2 3 3 3 2 2 2 3 5 5 3 2 1 2 3 3 3 2 2 3 2 1]`
Les fréquences correspondantes sont données par `freq_mary=la_majeur(notes_mary)`
A l'aide d'une boucle et de la première partie, écrire un script permettant de générer des tons de 0.5s pour chaque note et de les concaténer pour créer un vecteur y que l'on écoutera en en utilisant `soundsc`.

3) En réalité, les morceaux musicaux jouent également sur la durée des notes. Créer un fichier fonction synth.m permettant de créer le vecteur y du morceau avec la syntaxe y=synth(notes,durees) où freq est le vecteur de notes et durees le vecteur de même taille correspondant aux durées de chaque note (de sorte que le résultat du 3 serait équivalent à un vecteur avec toutes les valeurs à 0.5). Autrement dit, la ième note doit avoir un indice `notes(i)` et une durée `durees(i)`.

4) Enregistrer le morceau « Mary had a little lamb » au format wave en utilisant la fonction `audiowrite` (cf aide de Matlab) qui sera joint en rendu sous le nom *MHALL.wav*.

5) En plus des fonctions et du fichier wave, rendre un script *testsynth.m* permettant de reproduire la séquence génération/enregistrement pour ce morceau ainsi qu'un court rapport incluant vos observations. L'ensemble peut prendre la forme d'un livescript.

Deuxième partie : conception de l'IHM

Cette deuxième partie correspond à la conception de l'interface homme-machine, autant dans l'analyse des fonctionnalités souhaitées, dans la conception graphique, que dans la programmation des différentes fonctionnalités.

CAHIER DES CHARGES

A l'issue de la première partie, vous devez désormais avoir programmé des fonctions capables de générer des séquences sonores de notes et durées variables, de les jouer et d'exporter le signal ainsi généré. L'IHM créée devra donc a minima permettre :

- De renseigner une séquence de notes sous forme d'indices, de durées et de formes d'ondes au choix
- De visualiser le signal ainsi généré
- De jouer le signal sonore
- D'exporter le signal sous forme d'un fichier WAVE
- De sauvegarder et de charger une séquence précédemment créée
- De modifier certains paramètres (par exemple la fréquence d'échantillonnage)

Les fonctionnalités listées ci-dessus sont incontournables.

Par ailleurs, l'implémentation des fonctionnalités suivantes n'est pas exigée mais sera valorisée :

- Rajout d'une modulation en amplitude du signal généré selon un modèle ADSR, cf : https://fr.wikipedia.org/wiki/Enveloppe_sonore
- Rajout de formes d'onde (en plus du carré et du triangle), jusqu'à l'import éventuel de signaux sonores
- Rajout d'une autre piste (ie signal en parallèle, basé sur une autre fréquence de base)

Lors de la conception, une attention devra être apportée à l'utilisation de composants graphiques variés et favorisant l'ergonomie de l'interface.

NOTATION

La notation s'attardera particulièrement sur les points suivants :

- RESPECT DU CAHIER DES CHARGES (un non respect du cahier des charges sera sanctionné, et inversement l'ajout de fonctionnalités supplémentaires sera valorisé)
- RECOURS A DES COMPOSANTS VARIES
- ERGONOMIE (aisance d'utilisation et intuitivité)
- QUALITE DE PROGRAMMATION (dont intelligibilité des variables et structures et commentaires dans le code). **Un code non fonctionnel donnera lieu à un 0**

Pour ce dernier point, il est impératif de s'assurer de votre application se lance sur une version 2022 de MATLAB, par exemple en testant votre code sur une autre machine que la votre.

Attention également à vérifier que votre dossier compressé n'est pas corrompu.

La mutualisation de code inter binôme sera sévèrement sanctionné. De plus, le binôme s'engage en rendant son projet à ne pas avoir eu recours au plagiat.

MODALITES DE REMISE

Le travail sera effectué de préférence **en binôme** . Aucun monôme autorisé. L'organisation en trinôme est autorisée mais l'exigence sera alors accrue, dont l'extension du cahier des charges à au moins une des fonctionnalités optionnelles.

Le projet sera à rendre sur la plateforme CAMPUS via le compte d'un seul des membres du groupe, à une date limite qui sera communiquée ultérieurement.

Le rendu consistera en un dossier compressé comportant les deux ou trois noms de famille des membres du groupe et incluant :

1) *Un rapport synthétique :* celui-ci devra notamment expliciter la phase de conception, avec un schéma anticipé de votre interface (types de composants et placement), les liens envisagés entre les différents composants et tout autre élément jugé adéquat. Le but est de rendre compte de votre réflexion préalable avant la construction et le codage proprement dits. Des documents manuscrits scannées sont acceptés.
   Vous êtes aussi invité(e)s à y faire figurer les difficultés et limites éventuelles rencontrées ainsi que les choix qui en ont résulté, autrement dit à raisonner en tant qu'ingénieur(e) !

2) *Le projet lui-même*, comportant l'ensemble des fichiers nécessaires au bon fonctionnement de l'interface ainsi que ceux produits lors de la première partie (que vous aurez normalement utilisés dans la deuxième !).

Aucun travail en retard ni oubli de fichier ne sera accepté, donc n'attendez pas le dernier moment et vérifiez votre livrable !