Projet d'informatique : rapport final

I Limitation et problème du programme

L'utilisation de mémoire est assez conséquente, pour sortir notre fichier de 2.5MB nous avons un pique d'utilisation de la mémoire RAM atteignant 900MB. Cela signifie donc que nous gardons beaucoup d'informations inutiles ou répétitives en mémoire.

II Utilisation de constructions non-déclaratives

Nous ne sommes pas sortis du cadre fonctionnel pour la réalisation du programme.

III Implémentation surprenante

Rien à signaler.

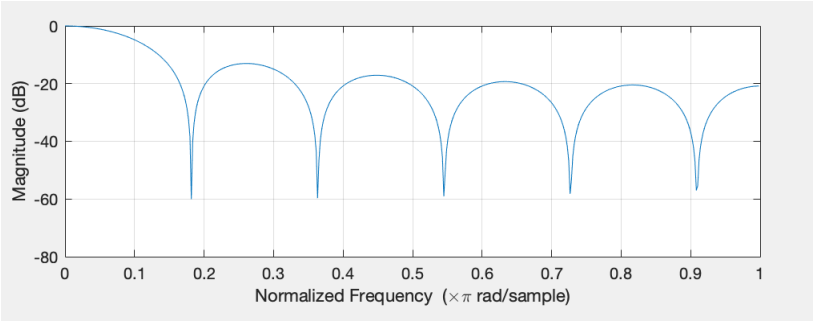
IV Complexité

- Duration : Notre fonction "Duration" appelle "DRatio" qui parcours une fois la liste contenant les notes et calcule le ratio à appliquer à la durée de chaque note grâce à la formule $\frac{\text{Nbre de secondes souhaitée}}{\text{Nbre de secondes totale}}$. Elle fait ensuite appelle à "Stretch" qui reparcours la liste et multiplie toute les durées par ce ratio. En posant n valant la taille de la liste. Nous avons donc : $T_{DRatio}(n) = 1 + T_{DRatio}(n-1)$ ce que nous pouvons simplifier en $T_{DRatio}(n) = 1 + n$ en sachant que $T_{DRatio}(0) = 1$. Nous pouvons appliquer le même raisonnement pour "Strectch". Que nous additionnons car, la fonction "DRatio" n'appelle "Stretch" qu'une seule fois tout à la fin. Nous obtenons donc une complexité en $\Theta(n)$.
- Merge : Notre fonction "Merge" utilise une sous-fonction qui effectue la somme pondérée de l'accumulateur et de la liste suivante (la pondération correspondant à l'intensité de cette musique). Nous savons que sommer deux listes ainsi que multiplier une liste par un facteur sont tout deux de complexité $\Theta(n)$. N valant ici la taille de(s) ⟨Music⟩. Nous devons effectuer ces fonctions m fois (m correspondant aux nombres de ⟨Music⟩ à merger). Nous avons donc une fonction de complexité $\Theta(n)$ pour la longueur des listes et $\Theta(m)$ pour le nombre de listes.
- Loop : Notre fonction "Loop" cherche d'abord à savoir si la durée demandée est plus petite que la durée de la liste de $samples$. Si c'est le cas, nous utilisons "List.take" afin de prendre le nombre d'éléments demandé. Dans ce cas-ci, la complexité correspond à celle de "List.take" qui est $\Theta(n)$, n valant la taille de la liste.
  Si la durée demandée excède la durée (en secondes) de la liste de $samples$, nous utilisons la fonction "Repeat" qui assemble une liste composée de copies de celle-ci de manière récursive terminale en utilisant l'opérateur 'l'. Cette fonction a une complexité $\Theta(m)$, m correspondant au nombre de répétitions. Nous utilisons ensuite "List.take" afin d'obtenir la partie tronquée de la fonction pour atteindre la durée désirée. Or, m dépend de n et vaut "Nbre de secondes souhaitée **div** Nbre de secondes de la liste" qui est beaucoup plus petit que n dans la plus part des cas. Nous pouvons donc simplifier la complexité en $\Theta(n)$.

V Extensions

- Fonctions complexes :
  - echo(delay :⟨duration⟩ decay :⟨factor⟩ repeat :⟨natural⟩ ⟨music⟩)
    Un filtre d'écho qui permet d'enchaîner plusieurs écho. Chaque nouvel écho est un écho du précédent.
  - crossfade(seconds :⟨duration⟩ ⟨music⟩ ⟨music⟩)
    Ce filtre superpose un fondu sortant (fade-out) sur la première musique et un fondu entrant (fade-in) sur la seconde musique.

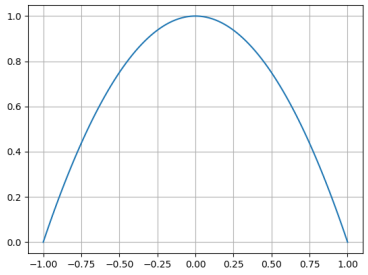
- lowpass(⟨music⟩)
  Ce filtre filtre passe-bas permet de laisser passer les basses fréquences et d'atténuer les hautes fréquences, c'est-à-dire les fréquences supérieures à la fréquence de coupure qui dans notre cas vaut 400Hz. Chaque échantillon de sortie équivaut à une somme pondérée des 11 derniers échantillons d'entrée dont les facteurs ont été calculés grâce à une simulation "MATLAB" pour un filtre d'ordre 10. Pour plus d'informations, voici un des sites consultés [1].



Schématisation du filtre passe-bas

- Ces trois fonctions complexes sont utilisées dans notre exemple à différents moments. Elles peuvent toutes trois être désactivées grâce au $boolean$ "Extension" dans la fonction "Mix".

- Lissage : La fonction "Lissage" permet de lisser une liste de $samples$ qui lui est donnée. Cette fonction applique un lissage correspondant à la fonction : $-x^2 + 1$, de [-1,0] pour le lissage d'entrée. Et de [0,1] pour le lissage de sortie.



Fonction $-x^2 + 1$

  Elle est commandé par un $boolean$ se trouvant dans la fonction Mix afin de pouvoir exécuter le programme normalement lors des tests. Le $boolean$ en question a pour non "Lissage".
- Instruments : L'implémentation des instruments a été commencée. Il est possible de modifier l'instrument voulu, mais ceci n'a pas d'impact pour la suite car ce n'est pas pris en charge dans notre fonction "Mix".
- Créativité : Nous avons choisit de retranscrire des bouts de partition pour piano du morceau "Pirates des Caraïbes". La partition est un peu difficile à lire mais fonctionnelle.

1. http ://herve.boeglen.free.fr/Tsignal/chapitre3/chapitre3.html