

GMIN315 - TP n°1 - synthèse sonore par programmation de bas niveau

Objectif : Le but de ce TP est de mettre en place les briques permettant de jouer des signaux sonores de la forme

$$s(t) = A \times ADSR(t) \times \text{timbre}(\text{fréq}, t) \text{ où}$$

- A est un gain de volume sonore appliqué à l'ensemble du son ; plus A est important, plus le son est fort ;
- $ADSR(t)$ décrit l'enveloppe du volume sonore, c'est un nombre entre 0 et 1 qui évolue au cours du temps ;
- fréq est la fréquence d'oscillation du son, qui peut varier au cours du temps. Plus la fréquence est élevée, plus le son est aigu ;
- timbre définit la forme des oscillations du son, donc sa nature : pur, vocal, instrumental etc.

Préambule :

- sur l'espace pédagogique, récupérer l'archive correspondant au TP1 et la décompresser dans votre espace ;
- vérifier que le projet se compile (`make x86` en 32 bits, `make x64` en 64 bits).
- ouvrir le fichier `main.cpp` dans votre éditeur favori

Exercice 1 : Compléter la fonction `cb_question_1` de manière à jouer un son "pur" $s(t) = A \times \sin(2 \times \pi \times \text{freq} \times t)$ indéfiniment. Modifier cette formule pour altérer le son (soyez créatifs !)

Exercice 2 : Compléter la fonction `main` pour permettre à l'utilisateur de modifier au clavier : l'amplitude ($\times 2$ ou $\div 2$), la fréquence ($\times 1.05946$ ou $\div 1.05946$, rapport entre deux demi-tons), le panorama (volumes relatifs droite/gauche) et l'altération du son.

Exercice 3 : Programmation de l'enveloppe ADSR

Compléter la fonction `enveloppe` qui, étant donné le nombre x de secondes écoulé depuis le début du son, renvoie un nombre $ADSR(x)$ entre 0 et 1 défini par la courbe de la figure 1.

Exercice 4 : compléter la fonction `frequence` qui :

- prend en données un tableau à deux lignes temps/fréquence et un temps x en secondes ;
- renvoie en résultat la fréquence correspondant au temps x par interpolation du tableau.

Par exemple, pour le tableau 1 et pour $x = 1$ l'algorithme renvoie 300 (interpolation linéaire) ou mieux, 282,8 (interpolation logarithmique).

Exercice 5 : compléter la fonction `timbre` de manière à calculer le signal sonore correspondant à un son pur, un son carré puis un son triangulaire.

Bonus pour les rapides : la *synthèse additive* consiste à additionner plusieurs sons purs, une fondamentale et des harmoniques, de telle sorte que les amplitudes c_i (constantes dans le temps) décroissent avec l'ordre de l'harmonique. Implémenter la partie "synthèse additive" de la fonction `timbre`. Pour éviter la saturation en volume, on pourra diviser le signal obtenu par la somme des c_i :

$$s(t) = \frac{\sum_{i=1}^n (c_i \times \sin(2 \times \pi \times i \times \text{freq} \times t))}{\sum_{i=1}^n c_i}$$

et par simplicité, on pourra stocker les coefficients c_i et le nombre d'harmoniques dans des variables globales.

Exercice 6 : Intégrer les trois fonctions précédentes dans la fonction `cb_globale`, de manière à lui faire jouer un son $s(t) = A \times ADSR(t) \times \text{timbre}(\text{fréq}, t)$. Utiliser cette fonction pour imiter des sons du quotidien, par exemple une voiture qui passe des vitesses, une goutte qui tombe dans l'eau, un fantôme, une mélodie connue... Soyez créatifs !

Evaluation : Enregistrez vos sons les plus réussis puis envoyez-les à votre encadrant.

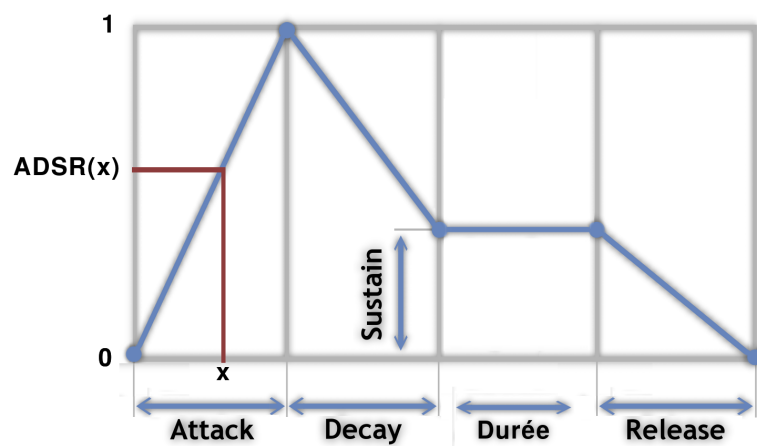


FIGURE 1 – Courbe ADSR (*source de l'illustration : Wikipedia*)

Temps	0	2	3	6
Fréquence	200	400	500	200

TABLE 1 – un exemple de tableau temps-fréquence