
INTERPRÉTEUR MUSICAL

Introduction

Votre logiciel va interpréter une chaîne de caractère (fichier texte) et la traduire en une musique (fichier WAV). La chaîne de caractères en entrées est habituellement construite par un autre logiciel (vous n'avez pas à vous occuper de ce logiciel).

Votre logiciel va recevoir un nom de fichier en entrées sur la ligne de commande, une valeur numérique indiquant le tempo de la musique et le nom du fichier en sorties. Le code pour lire les arguments de la ligne de commande et lire le fichier est déjà écrit. La chaîne de caractères lus et le tempo sont donnés en arguments à une fonction 'traduction'. Cette fonction reçoit une chaîne de caractère, le tempo et les transforme en une séquence de nombre représentant le son à placer dans le fichier WAV. C'est dans cette fonction que vous devez écrire votre code.

La transformation que va effectuer votre logiciel va être expliquée en trois étapes. Dans un premier temps, comment transformer les caractères en une suite de note musicale. Deuxièmement, nous allons transformer les notes en leurs fréquences et finalement générer l'onde sonore à partir des fréquences. Le code pour lire la chaîne de caractères et pour construire un fichier WAV est déjà fourni.

Génération des notes

La première étape consiste à construire une suite de note. Pour cela, nous allons utiliser une mécanique d'interprétation impérative. C'est-à-dire que les caractères de la chaîne vont représenter des instructions indiquant des modifications (action) à apporter à un état (une description). L'état que vous allez utiliser contient l'information suivante :

- Une note, généralement composée d'un numéro de note entre 0 et 11 et un numéro d'octave représenté par un chiffre entre 2 et 9. Initialement note = 9 et octave = 4.
- Une durée : une valeur Double, initialement à 1.
- Un volume : une valeur (je conseille un Double) entre 0 et 10, initialement à 5.
- Une pile contenant des états.

Chaque caractère va possiblement modifier des valeurs dans cet état. Voici la liste des caractères et leurs effets :

- '0' : Volume = 0.
- '+': Augmente le volume de 1, maximum 10.
- '-': Diminue le volume de 1, minimum 0.
- '1': Durée = 1.

- '*' : Multiplie la durée par 2.
- '.' : Multiplie la durée par 1.5.
- '/' : Divise la durée par 2.
- '[' : Sauvegarde l'état courant sur la pile d'états.
- ']' : Remplace l'état courant par l'état au sommet de la pile d'états. Le sommet de la pile d'états est dépilé.
- 'a'..'g' : Ces caractères minuscules remplacent le numéro de note selon la table plus bas.
- '2'..'9' : Le nombre remplace l'octave de la note.
- '&' : Diminue le numéro de la note de 1, si le numéro de note était à 0, alors il passe à 11 et l'octave est diminué de 1. Si la note est à 0 et l'octave à 2, alors cela n'a pas d'effet.
- '#' : Augmente le numéro de la note de 1, si le numéro de note était à 11, alors il passe à 0 et l'octave est augmenté de 1. Si la note est à 11 et l'octave à 0, alors cela n'a pas d'effet.
- '!' : Indique qu'il faut jouer la note courante. Donc, ajouter la note courante à la suite de note à jouer.
- Tout autre caractère est accepté, mais ils n'ont aucun effet.

Cela nous donne une liste de notes. Il faut maintenant trouver la fréquence de chaque note.

Lettre	Note
a	9
b	11
c	0
d	2
e	4
f	5
g	7

Génération des fréquences

À chaque note est attribuée une fréquence d'oscillation pour l'onde sonore la représentant. Les valeurs représentant la note (numéro n et octave c) sont utilisées pour trouver cette fréquence.

$$q = 12c + n - 57$$

$$a = 2^{(1/12)}$$

$$f = 440a^q$$

Finalement, il reste à construire les ondes à partir des fréquences, durée et volume de chaque note.

Génération de l'onde sonore

La dernière étape consiste à générer une onde pour chaque note de la mélodie. Ces ondes seront concaténées pour construire l'onde à placer dans le fichier WAV. Un fichier WAV contient une représentation digitale d'une

onde, donc un échantillonnage : évaluer la fonction de l'onde à intervalle régulier et placer les valeurs dans une liste.

Il nous faut donc déterminer l'intervalle de temps entre chaque échantillon. L'industrie utilise une fréquence de 44 100 Hz pour graver les CD. Cela indique qu'il y aura 44 100 échantillons par seconde. Il vous faut donc 1 échantillon toute les (1/44 100) secondes (nous appelons cet intervalle la période λ). L'onde commence au temps $t = 0$. Ensuite t est incrémenté de la période λ pour chaque échantillonnage successif $t = t + \lambda$. Il reste à évaluer la fonction d'onde au temps t avec un volume v et une fréquence f :

$$r_t = v \sin(2\pi f t)$$

Il reste à déterminer le nombre d'échantillons pour une note, cela dépend de la durée d de la note. Votre fonction a reçu le tempo b de la musique. Cela indique combien de notes de durée 1 seront jouées par minute. Donc, une note aura une durée, en seconde, de $s = 60d/b$. Puisqu'il y a un échantillon à toutes les λ secondes, votre note devra être échantillonnée (s/λ) fois. Vous devrez construire une liste d'échantillon pour chaque note et ensuite les concaténer, ce qui vous donnera une seule liste de valeur Double entre -1 et 1.

Les fichiers WAV, tel que le programme va les construire, demande des valeurs entières comme échantillon. Il reste donc une dernière opération, vous devez multiplier chaque valeur de la liste par 32767 afin d'obtenir une liste d'entier entre -32767 et 32767. Votre fonction doit retourner cette liste.

Code

Un fichier de départ est disponible sur le site du cours. Vous devez écrire le code pour la fonction 'chaineVersSon'. Vous pouvez (devez) ajouter des fonctions intermédiaires pour rendre votre code lisible. Il y a déjà des librairies importées dans l'entête du code. Vous ne pouvez pas importer d'autres librairies que celle déjà disponible.

Commentez votre code.

Remise

Vous devez remettre votre fichier (1 seul) sur Moodle avant le 28 février 23 :55. Vous pouvez travailler en équipe de deux (2).

Évaluation

- Fonctionnalité : 16 pts. Le TP compile et les tests donnent les bons résultats. Quelques tests seront disponibles sur le site du cours.
- Lisibilité : 4 pts. Le code doit être commenté. Pour chaque fonction, donner son rôle et le rôle de ces paramètres.