DM3. Structure d'un fichier MIDI

- Pour les PTSI 1 Avant toute chose, créer un répertoire DM3 dans votre répertoire "Informatique". Sur le site ENT/Moodle/cours"infoPTSI", télécharger jingle.mid que vous copierez dans votre répertoire DM3.
- Pour les PTSI 2 Avant toute chose, créer un répertoire DM3 dans votre répertoire "Informatique". Sur le site https://ptsilamartin.github.io/info/DM.html, télécharger jingle.mid que vous copierez dans votre répertoire DM3.

Rappel : On prendra bien soin, dans tout le TP, de documenter les fonctions écrites et de les tester. Les tests seront présentés en commentaires dans le script.

Mercredi 14 février à 22h : date limite d'envoi du fichier.

Pour les PTSI1

Déposez votre script sur le site ENT/Moodle/cours"infoPTSI"/DM3

Pour les PTSI2

Déposez votre script sur le site http://envoi.lamartin.fr/.

- Concevoir un algorithme répondant à un problème précisément posé.
- Écrire des instructions conditionnelles avec alternatives, éventuellement imbriquées.
- Choisir un type de données en fonction d'un problème à résoudre.
- Traduire un algorithme dans un langage de programmation.
- Lire et écrire dans un fichier.

1 Présentation

1.a Généralités



A l'origine, le MIDI (Musical Instrument Digital Interface) est une interface, une norme matérielle (câbles et prises) et logicielle (protocole d'échange de données) de communication entre instruments de musique électroniques. Avec un clavier Midi branché sur un ordinateur, il est possible d'enregistrer des accords, des rythmes, des mélodies, et même des orchestrations complètes grâce à des logiciels nommés arrangeurs ou séquenceurs.

Contrairement à un enregistrement classique à l'aide d'un microphone, ce ne sont pas des sons qui sont enregistrés, mais les notes jouées sur le clavier. La norme Midi attribue un numéro à chacune de ces notes. Le résultat de l'enregistrement est un fichier Midi composé uniquement de nombres : les numéros des notes et des indications sur l'interprétation ainsi que les numéros des sons à utiliser.

Les fichiers MIDI peuvent être lus sur PC via le lecteur Windows Media Player, sur Mac il est nécessaire d'installer l'application VLC version 3.0.4.

1.b Élements de structure à identifier dans ce DM

Il existe au moins deux plages dans un fichier MIDI:

- plage d'entête (commençant par "MThd")
- plages de données (commençant par "MTrk")

La plage de d'entête

La première plage de données renseigne sur la structure du fichier et la plus petite unité de temps considérée :

- 4 octets nombre magique x4D546864: "MThd"
- 4 octets espace des spécifications du fichier : 0, 0, 0, 6 pour les 6 octets suivants
- 2 octets type SMF (Standard Midi File) :
 - . 0, 0 : une seule plage de données où se mélangent les événements de plusieurs canaux ;
 - . 0, 1 : plusieurs plages qui se suivent dans le fichier, mais jouées simultanément ;
 - . 0, 2 : plusieurs plages, jouées l'une après l'autre (peu utilisé).
- 2 octets nombre de plages de données ("MTrk"), de 1 à 65.535
- 2 octets nombre de divisions de la noire (résolution temporelle du fichier)

La plage de données

Les plages sont composées de délais, d'événements musicaux, de contrôles et de métadonnées dont il sera question dans la section suivante.

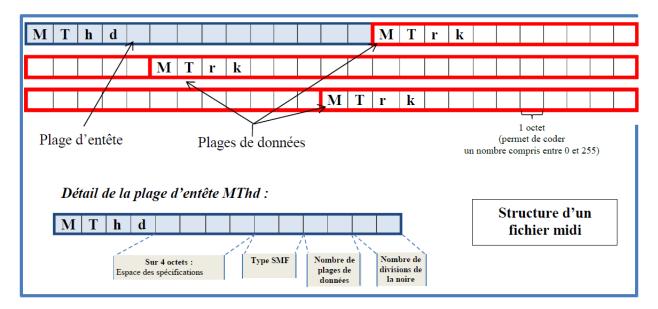


FIGURE 1 – Trame d'un fichier MIDI

Remarque: on utilise le code ASCII pour les lettres (par exemple 'a' est codé en décimal par 97). Pour passer du code ASCII au code décimal et inversement, se référer au TP4 (cryptographie). Pour répondre à votre curiosité, vous pouvez retrouver l'ensemble des informations sur cette norme en vous rendant sur le site: https://www.jchr.be/linux/midi-format.htm

2 Travail à effectuer

2.a Lecture - Écriture d'un fichier MIDI et mise en forme des données

Contrairement aux fichiers habituellement utilisés, notamment dans le TP6, les fichiers midis sont des fichiers binaires. Nous allons lire ce fichier octet par octet, pour cela il suffit de **rajouter le caractère** 'b' au moment de l'ouverture du fichier.

En dehors de l'ajout de ce caractère, vous utiliserez la procédure habituelle vue en cours pour écrire/lire un fichier.

On obtient alors des objets de type « bytes » de la forme $b'\x01\x02\x03'$ qui contiennent les octets du fichier lu. On propose de convertir directement ces objets en listes d'entiers de la forme [1,2,3], que vous avez davantage l'habitude de manipuler. Pour ceci on utilise les fonctions suivantes :

PYTHON

```
listeEntier = list(ligneBytes) #conversion de bytes en liste d'entier ligneBytes = bytes(listeEntier) #conversion d'une liste d'entier en bytes
```

Lecture du fichier

Question 1. Écrire un programme qui lit le fichier jingle.mid et affiche dans le shell les lignes du fichier une à une.

Remarque : sur Pyzo, il faut changer le répertoire courant pour que lors de l'exécution le script aille chercher le fichier dans le dossier où se trouve les fichiers utiles. On utilise les instructions ci-dessous :

PYTHON

```
import os #importation de la bibliothèque os os.chdir(r'D : \chemin') #changement de répertoire courant #chemin est le chemin complet vers le dossier contenant le fichier midi
```

Une autre méthode, plus simple (toujours avec Pyzo), par un clic droit sur l'onglet de votre programme, sélectionner "Run File as script" pour la première exécution. Tous les fichiers utiles doivent être dans le même dossier.

Mise en forme des données

Question 2. Créer une liste les_lignes_entier qui contient les listes d'entiers correspondant à chaque ligne lue, en utilisant la fonction list présentée ci-dessus.

Les sauts de lignes n'ont rien de significatif ici (le caractère '\n' est un caractère comme un autre en MIDI : ne pas l'enlever); on souhaite donc regrouper l'ensemble des données en une seule liste, en mettant les lignes obtenues « bout à bout ».

Question 3. Créer une liste donnees regroupant toutes les lignes. Vérifiez que cette liste contient bien 23783 éléments.

Écriture d'un fichier MIDI

La suite du TP propose de modifier des éléments dans le fichier MIDI, puis d'écouter l'effet de la modification. Il faut pour cela être capable d'écrire un nouveau fichier MIDI à partir des données modifiées.

Question 4. Écrire une fonction ecrire_midi(nom du fichier:str,liste:list) qui écrit un fichier midi à partir d'une liste d'entiers. Cette fonction prend pour arguments le nom du fichier à générer (chaîne de caractère) et la liste de données à écrire (liste d'entiers). Cette fonction ne renvoie rien.

Aide : cette fonction reprend les étapes ci-dessus en sens inverse : conversion en bytes, puis écriture en binaire. Pas besoin de redécouper les lignes.

Question 5. Tester cette fonction en générant le fichier jingle2.mid qui contient exactement les mêmes données que celles du fichier initial. Vérifier, en écoutant, que les sons n'ont pas été modifiés.

2.b Étude de l'entête du fichier

A partir de la liste d'entiers donnees obtenue à la question 3 :

- Question 6. Afficher les 14 octets définissant la plage d'entête.
- Question 7. Répondre dans le script sous forme de commentaire :
 - Quelle est la signification des quatre premiers termes (vous justifierez votre réponse par une ligne de code en utilisant les fonctions vues dans le TP4)?
 - Quel est type SMF utilisé ici?
 - Quel est le nombre de plages?
 - Quel est le nombre de division de la noire?

2.c Modification du tempo

Question 8. Placer le nombre de divisions de la noire à 255. Écrire un nouveau fichier jingle_tempo.mid. Vérifier l'effet obtenu.

La modification pourra être conservée pour la suite du DM.

2.d Identification des plages MTrk

Chaque plage de données commence par le mot MTrk.

Question 9. Convertir le mot b'MTrk en entiers. Noter L_MTRK la liste des quatre entiers correspondants.

Question 10. Créer une liste les_indices qui contient l'indice de début de chaque piste MTrk.

Question 11. Valider le nombre de plages obtenues au regard de la question 7.

2.e Modification d'un instrument

Découverte des instruments utilisés dans le fichier jingle mid

Les données de chaque piste « MTrk » contiennent des événements qui peuvent être par exemple : la production d'une note, le choix d'un instrument, la modification d'un timbre, la modification du volume . . .

La modification d'un instrument se code sur 2 octets :

- le premier octet « Cc » (en hexadécimal) indique un changement d'instrument sur le canal c.
- le second octet indique le code de l'instrument choisi (cf annexe).

Exemple: « C4 06 » : changement d'instrument sur le canal 4, code instrument : 06, correspondant au clavecin.

Sachant qu'un fichier MIDI contient au plus 16 canaux numérotés de 0 à 15, un changement d'instrument sera donc codé, en **hexadécimal**, de C0 à CF. En notation **décimale** un changement d'instrument est codé par 192 pour le canal 0; 193 pour le canal 1; 194 pour le canal 2; etc; 207 pour le canal 15.

Note : Un canal MIDI permet le transport des données pour commander un instrument équipé d'un port MIDI.

- Question 12. Rechercher, dans la liste données, tous les changements d'instrument : créer deux listes :
 - une liste les_instruments contenant tous les codes instruments utilisés;
 - une liste les_canaux qui contient le numéro du canal utilisé (de 0 à 15).

Question 13. Indiquer, sous forme de commentaire, la liste des instruments utilisés, et le canal correspondant.

Modification d'un instrument

Question 14. Remplacer l'instrument « violon » par un « jeu de cloche ». Écrire le résultat dans le fichier jingle_cloche.mid.

Question 15. Écouter l'effet obtenu!

Facultatif: atténuation de la mélodie

Question 16. En vous aidant de la norme MIDI (voir par exemple le site https://www.jchr.be/linux/midi-format.htm), modifier le « control change » associé au volume du canal « channel volume », pour diminuer fortement le volume de la mélodie (canal de l'ex violon). Écrire le fichier jingle_acc.mid.

ANNEXE

Extrait de la norme General Midi : codage des instruments

Codage décimal	Pianos et claviers	Codage décimal	Cordes et timbales
	Piano à queue		Violon
	Piano de concert		Alto
	Piano électrique		Violoncelle
	Piano bastringue		Contrebasse
	Piano électronique - 1		Cordes - effet trémolo
	Piano électronique - 2		Pizzicato de cordes
	Clavecin		Harpe
	Clavicorde		Timbales
Codage décimal	Instruments chromatiques	Codage décimal	Ensembles et chœurs
	Célesta		Ensemble de cordes – 1
	Glockenspiel		Ensemble de cordes – 2
	Boîte à musique		Cordes de synthèse – 1
	Vibraphone		Cordes de synthèse – 2
	Marimba		Chœurs – Aaah
	Xylophone		Chœurs – Oooh
	Jeu de cloches		Chœurs de synthèse
15	Dulcimer	55	Tutti d'orchestre symphonique
Codage décimal	Orgues	Codage décimal	Cuivres
16	Orgue Hammond	56	Trompette
17	Orgue Hammond - effet percussion	57	Trombone
18	Orgue électronique	58	Tuba
19	Grand orgue	59	Trompette en sourdine
20	Harmonium	60	Cor d'harmonie
21	Accordéon	61	Pupitre des cuivres
22	Harmonica	62	Cuivres de synthèse - 1
23	Bandonéon	63	Cuivres de synthèse - 2
Codage décimal	Guitares	Codage décimal	Anches
24	Guitare acoustique nylon	64	Saxophone soprano
25	Guitare acoustique métal	65	Saxophone alto
26	Guitare électrique jazz	66	Saxophone ténor
27	Guitare électrique	67	Saxophone baryton
28	Guitare électrique - son amorti	68	Hautbois
29	Guitare électrique - son saturé	69	Cor anglais
30	Guitare électrique avec distorsion	70	Basson
31	Harmoniques de guitare	71	Clarinette
Codage décimal	Basses	Codage décimal	Vents
	Basse acoustique		Piccolo
	Basse électrique	-	Flûte traversière
	Basse électrique avec médiator		Flûte à bec
	Basse frettless		Flûte de pan
	Slap basse – 1		Cruche (Jazz)
	Slap basse – 2		Flûte shakuhachi
	Basse de synthèse - 1		Sifflet
	Basse de synthèse - 2		Ocarina
39	Dasse de synthese - Z	/9	Ocarina