

LP WAM - DU II

POO

Projet de fin d'année

1 Objectifs

Le projet de « POO Python » permet de mettre en œuvre les principaux éléments du contenu du cours dans le cadre de la conception d'une petite application en programmation objet. Le travail est à réaliser en binôme.

2 Contexte du projet

On s'intéresse dans ce projet aux fichiers de musique avec perte de type **MP3 (Moving Picture Experts Group Audio Layer 3)** et sans perte de type **FLAC (Free Lossless Audio Codec)**. Ces fichiers musicaux possèdent des métadonnées : des informations complémentaires stockées dans le fichier MP3 ou FLAC mais qui ne font pas partie du flux audio.

Concernant ces métadonnées, pour les MP3 elles sont organisées selon le format « **ID3** » (cf. <https://id3.org/>) et pour les FLAC selon le format « **Vorbis comment** » (cf. <https://wiki.xiph.org/index.php/VorbisComment>). Elles rassemblent les informations de type durée du morceau, nom de l'artiste, nom de l'album, nom du morceau, etc. Les formats MP3 et FLAC sont donc des conteneurs avec une partie pour les méta-données et une autre partie qui constitue le flux audio proprement dit.

Par ailleurs, ces fichiers MP3 et FLAC peuvent être organisés en playlist indépendamment de leur emplacement sur le disque dur de l'utilisateur. Ces playlists sont structurées et respectent des formats : dans le cadre de ce projet, nous retiendrons le format **XSPF (XML Shareable Playlist Format)** : <https://xspf.org/> : il s'agit d'un format XML de playlist.

Dans un premier temps, les principales actions de votre logiciel seront :

- l'extraction des méta-données d'un fichier MP3 ou FLAC ;
- l'exploration complète d'un répertoire (d'un dossier et de l'ensemble de ses sous-dossiers) contenant des fichiers musicaux : il faudra prévoir un filtre pour ne retenir que les fichiers MP3 et/ou FLAC de cette arborescence, avec vérification de chaque extension et type MIME des fichiers trouvés pour ne retenir que les fichiers MP3 et FLAC ;
- la constitution d'une playlist par défaut contenant l'ensemble des fichiers MP3 et FLAC du répertoire exploré.

En mode console (terminal) « CLI » : les paramètres attendus sur la ligne de commande sont : le nom du fichier MP3 ou FLAC à analyser pour en extraire les métadonnées ou le nom du dossier à explorer pour générer une playlist par défaut : vous utiliserez un paramètre supplémentaire permettant de spécifier le type d'entrée (« **-f** » (**file**) pour un fichier, « **-d** » (**directory**) dans le cas d'un répertoire : ces 2 options étant exclusives l'une de l'autre). Le programme doit afficher directement dans la console le résultat de son analyse. Si aucun paramètre n'est indiqué le programme affiche un message d'erreur fonctionnelle ; avec l'option « **-h** » (**help**), le programme affiche l'aide et les options possibles. Une option supplémentaire « **-o** » (**output**) permet d'indiquer le fichier de sortie pour sauvegarder le résultat d'une extraction de la playlist dans un fichier (format XSPF) à spécifier par l'utilisateur.

En mode graphique « GUI » : l'exploration d'une arborescence quelconque de fichiers permettra de lister tous les fichiers et leurs emplacements afin de générer la playlist par défaut ou une playlist personnalisée, choix des morceaux et choix du nom de la playlist (au format XSPF), on pourra donc réouvrir une playlist enregistrée précédemment. Pour chaque fichier MP3 ou FLAC, on pourra visualiser les principales métadonnées.

Quelques scénarii d'exécution (exemples fictifs de lancement de vos 2 programmes) :

```
python3 cli.py
python3 cli.py -h
python3 cli.py -d .
python3 cli.py -f music.mp3
python3 cli.py -d ./music/ -o playlist.xspf
python3 gui.py
```

Explications : les 5 premières commandes concernent le mode console (aussi appelé mode terminal ou fenêtre de commande) et la dernière commande permet de lancer l'interface graphique :

- la première ligne doit indiquer qu'il manque des paramètres et doit proposer de taper « `-h` » (ou « `--help` ») pour obtenir de l'aide ;
- la deuxième ligne affiche les modes d'utilisation de votre logiciel en mode console (i.e. les options possibles et leur rôle) ;
- la troisième ligne liste et analyse tous les fichiers à partir du dossier spécifié (« `-d` » ou directory) [ici à partir du dossier courant (« `.` »)] en parcourant l'ensemble de l'arborescence des sous-dossiers, en mode console ;
- la quatrième ligne prend en entrée le fichier « `music.mp3` » (« `-f` » ou file) et affiche à l'écran les métadonnées de ce fichier, en mode console (par défaut, on considère le fichier dans le répertoire courant mais le chemin peut être spécifié)
- la cinquième ligne prend en entrée le sous-dossier « `./music/` » du répertoire courant et sauvegarde la playlist générée dans le fichier xspf spécifié, en mode console ;
- la dernière ligne correspond au lancement de l'interface graphique.

Pour les fichiers de playlist générés, vous utiliserez le validateur en ligne « <https://validator.xspf.org/> » pour vérifier que le fichier généré est valide.

Des améliorations des fonctionnalités de base sont attendues :

- en mode graphique, extraire également l'image (thumbnail) de couverture (cover) de l'album correspondant au morceau et l'afficher à l'utilisateur ;
- en mode console et graphique, offrir la possibilité d'écouter le morceau de musique.

Et pour aller encore plus loin (et seulement si toutes les fonctionnalités précédentes sont fonctionnelles), vous pourrez intégrer à votre projet :

1. la possibilité de modifier (et sauvegarder) les TAGS d'une chanson
2. la possibilité d'aller chercher les informations complètes d'un album via une API-Web

3 Déroulement et attendus

3.1 Planning

- identifier les principales sous-tâches du projet à réaliser, leur niveau de priorité, la répartition des rôles au sein du binôme ainsi que le planning correspondant pour chaque tâche (période et durée). Vous créerez un diagramme de GANTT¹ de votre projet. Dans tous les cas, votre diagramme de Gantt est à envoyer par mail au format png (capture ou export) avant le **vendredi 7 avril 2023, 10h00 : 1 point**. Ce fichier respectera les règles de nommage (§ 3.4) ;
- un premier point d'avancement sera fait le **vendredi 7 avril à distance, en fin de journée : 1 point**
- un second point d'avancement aura lieu le **mardi 9 mai à distance, en fin de journée : 2 points** (à titre indicatif, un niveau de réalisation d'environ 75% est attendu lors de ce point d'avancement) ;
- une vidéo faisant office de démonstration (au plus 5 min) : **5 points** (envoi en semaine 20, **mercredi 31 mai 18h00 au plus tard**). Cette démonstration permettra de visualiser les deux fonctionnements en mode console et graphique (il est utile de prévoir un scénario...). Ce fichier respectera les règles de nommage (§ 3.4) ;
- tous les autres attendus (voir § 3.3) en semaine 20, **mercredi 31 mai 12h00 au plus tard** ;
- une soutenance de 20 min par groupe (10 minutes de présentation et 10 minutes de question) est prévue le **vendredi 2 juin 2023, à partir de 12h30 et jusqu'à 15h30 au plus tard**. Un ordre de passage avec la salle vous sera communiqué plus tard.

3.2 Soutenance (4 points)

- la diapositive de titre présentera le binôme, le contexte, le sujet. (i.e. : la page de garde sera compacte) ;
- les autres diapositives (10 au plus) devront présenter les spécificités de réalisation de l'équipe projet, donc aucune information "évidente" (ex. détail du sujet, progression personnelle, ...) ne devra être mentionnée ;
- la diapositive de conclusion mettra en évidence le niveau d'achèvement du projet (points traités et non traités du cahier des charges et extensions s'il y en a) ;
- vous devrez prévoir une version PDF de votre diaporama au cas où.

Les informations suivantes devront être présentes : la répartition des tâches, les principaux éléments de conception.

NB : à éviter ABSOLUMENT : les diagrammes de classes UML illisibles (trop chargés, ...), les programmes (code Python), les captures d'écran (puisque'il y a aussi une vidéo de démonstration), la liste des outils utilisés, ...

Important :

- vous devrez avoir une clé USB contenant votre présentation et votre ordinateur portable qui sera allumé et prêt AVANT d'entrer dans la salle (et un adaptateur vidéo vers de l'USB C).
- vous veillerez à une répartition équitable de votre temps de parole au sein du binôme aussi bien pour la présentation que pour la démonstration.

3.3 Attendus (7 points)

- complétude et qualité du projet : **3 points** (fichier « readme.txt », code Python documenté...). Les livrables sont **à déposer en MP sur le discord** avant les dates indiquées

1. <https://www.ganttproject.biz/download/free>

ET à partager via un lien wetransfer...

- fichier « `readme.txt` » contenant les noms / prénoms des membres du projets ainsi que les informations spécifiques utiles (**1 point**);
- rapport de projet (minimum 5 pages, maximum 10 pages) : **3 points** (le fond et la forme seront évalués). Les 2 fichiers suivants sont à rendre :
 1. le document de traitement de texte (format éditable ODT ou DOCX);
 2. la version PDF de votre rapport.
- l'ensemble des fichiers sources du projet (.py);
- et la documentation (si possible au format Doxygen, à défaut au format Pydoc).

3.4 Règles de nommage

Les fichiers doivent respecter le nommage suivant `DU_II-Projet_P00-NOM1_NOM2-xx.zip`, où **xx**, par exemple, sera remplacé par **Gantt** en réponse à la question 1 du §3.1!

Lors du rendu final, **tous les fichiers et sous-dossiers** à remettre doivent être placés dans **un répertoire unique portant les 2 noms du binôme** (sous la forme `NOM1_NOM2`) à compresser en un seul fichier au **format zip** qui sera déposé comme indiqué au §3.3 et doivent respecter la hiérarchie ci-dessous :

```
NOM1_NOM2
|__ doc_
|    |_ demonstration
|    |_ diaporama
|    |_ rapport
|
|__ src_
|    |_ cli
|    |_ gui
|
|_ README.txt
```

Le non respect de ces règles entraînera des points de pénalité!