

Université de Cergy-Pontoise

RAPPORT

Pour le projet du module Python

sur le sujet

Métadonnées de fichiers MP3/FLAC et playlist XSPF



Enseignant: **M.Bourdon**

Rédigé par

ABERKANE Mehdi , GHEZIL Achref & JAAFAR Amir

L3-Informatique 2021-2022

Table des matières

<u>Introduction au projet</u>	<u>3</u>
<u>A)But & objectif du projet</u>	<u>3</u>
<u>B)Fonctionnalités attendues du projet.....</u>	<u>3</u>
<u>B.1)En mode console (CLI):.....</u>	<u>3</u>
<u>B.2)En mode graphique (GUI) :.....</u>	<u>3</u>
<u>B.3)Fonctionnalités Envisagées :.....</u>	<u>3</u>
<u>Spécifications du projet</u>	<u>4</u>
<u>A)Les principaux choix techniques</u>	<u>4</u>
<u>Réalisation</u>	<u>5</u>
<u>A)Planning détaillé des différentes tâches du projet.</u>	<u>5</u>
<u>B)Avancement du projet</u>	<u>5</u>
<u>B.1)En mode console (terminale) «CLI»:.....</u>	<u>5</u>
<u>B.2)En mode graphique «GUI» :.....</u>	<u>8</u>
<u>C)Architecture de notre application :.....</u>	<u>9</u>
<u>Conclusion</u>	<u>10</u>
<u>A)Récapitulatif de la réalisation :.....</u>	<u>10</u>
<u>B)Conclusion & Améliorations.....</u>	<u>10</u>
<u>Annexe.....</u>	<u>11</u>
<u>A)Mock-up déroulement du GUI :.....</u>	<u>11</u>

Introduction au projet

A) But & objectif du projet

Le projet de l'option « Python » permet de mettre en œuvre les principaux éléments du contenu du module dans le cadre de la conception d'une petite application en programmation objet ou fonctionnelle (procédurale).

B) Fonctionnalités attendues du projet

B.1) En mode console (CLI):

- L'extraction des méta-données d'un fichier MP3 ou FLAC donné. (avec la commande -f [Chemin_fichier_audio])
- L'exploration complète d'un répertoire 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 . (avec la commande -d [Chemin_dossier])
- La constitution d'une playlist par défaut contenant l'ensemble des fichiers MP3 et FLAC du répertoire donné. (avec la commande -d [Chemin_Dossier] -o [Nom_playlist])
- Améliorations de bases :
 - Offrir la possibilité d'écouter le morceau de musique.(avec la commande -l [Chemin_fichier_audio])

B.2) En mode graphique (GUI) :

- L'affichage des méta-données d'un fichier MP3 ou FLAC donnée (possibilité de sauvegarde).
- Génération d'une playlist par défaut contenant l'ensemble des fichiers MP3 et FLAC du répertoire exploré. (possibilité de sauvegarde et ouverture d'une playlist enregistrée précédemment).
- Génération d'une playlist personnalisé contenant les fichiers MP3 et FLAC sélectionné dans le répertoire exploré. (possibilité de sauvegarde et ouverture d'une playlist enregistrée précédemment).
- Améliorations de bases :
 - Extraire également l'image (*thumbnail*) de couverture (*cover*) de l'album correspondant au morceau et l'afficher à l'utilisateur.
 - Offrir la possibilité d'écouter le morceau de musique.

B.3) Fonctionnalités Envisagées :

Au sujet de l'amélioration consistant à jouer le son, côté console comme côté graphique il y a affichage d'une interface graphique contrôlable depuis :

- Le clavier, dans la situation où vous l'ouvrez depuis la console (programme CLI) donc pas d'obligation d'utiliser la souris.
- L'interface graphique directement avec ses différents boutons (Utilisation GUI)

Et pleins d'actions différentes (Pause/Reprendre , Recommencer , Augmenter/Baisser le volume)

Spécifications du projet

A) Les principaux choix techniques

- Bibliothèque utilisée pour l'extraction des métadonnées: La bibliothèque que nous avons utilisé pour l'extraction des métadonnées est nommée ***TinyTag***. Nous l'avons choisi car elle est très simple à utiliser et également parce que c'est la première que nous avons trouvé qui proposait l'extraction de la couverture d'un morceau.
- Bibliothèque utilisée pour le parcours récursifs d'un dossier (CLI): La bibliothèque que nous avons utilisé pour le parcours récursifs d'un dossier est nommée et la récupération des chemins est ***Os*** plus précisément ses méthodes :
 - ***Os.walk()*** : pour la récupération sous forme de triplet (le chemin du dossier et de ses sous dossiers , le nom des sous-dossiers et le nom des fichiers)
 - ***Os.path.join()*** : pour la concaténation des chemin de dossiers avec les noms des sous dossiers et fichiers.
- Bibliothèque utilisée pour le parcours récursifs d'un dossier (GUI): La bibliothèque que nous avons utilisé pour le parcours récursifs d'un dossier est nommée ***Tkinter*** plus précisément sa méthode :
 - ***filedialog*** : permettant l'ouverture d'une boîte de dialogue basée sur l'arborescence de fichier de votre ordinateur. Mais nous ne l'avons pas utilisé tel quel nous sommes passés par ses méthodes :
 - ***filedialog.askopenfilename()*** : pour la sélection d'un unique fichier dans l'arborescence de dossier créée et qui retourne son chemin.
 - ***filedialog.askdirectory()*** : pour la sélection d'un unique dossier dans l'arborescence de dossier créée et qui retourne son chemin.
 - ***filedialog.askopenfilenames()*** : pour la sélection d'un ou de plusieurs fichiers dans l'arborescence de dossier créée et qui retourne les chemins sous forme d'une liste.
- Bibliothèque utilisée pour le filtre de types Mimes : La bibliothèque que nous avons utilisé pour la vérification du type mimes des fichiers récupérés lors du parcours de dossier est ***mimetypes*** plus précisément sa méthode :
 - ***mimetypes.guess_type()*** : qui renvoie sous la forme d'un tuple (le type mime du fichiers et son encodage) dans notre cas il ne fallait garder que '***audio/mpeg***' et '***audio/x-flac***'
- Bibliothèque utilisée pour la lecture d'un morceau et pour l'interface graphique : La bibliothèque que nous avons utilisé pour les interfaces graphiques est ***pygame*** plus précisément ***pygame.display*** pour toutes les interfaces graphique ***pygame.mixer.music***: pour la lecture d'un fichier audio (possibilité de lecture , mise en pause , replay , modification du volume,...) et ***pygame.image*** pour l'affichage de la couverture d'un morceau car elle contient plusieurs sous méthodes pour l'affichage et placement d'une image donnée.

Réalisation

A) Planning détaillé des différentes tâches du projet.

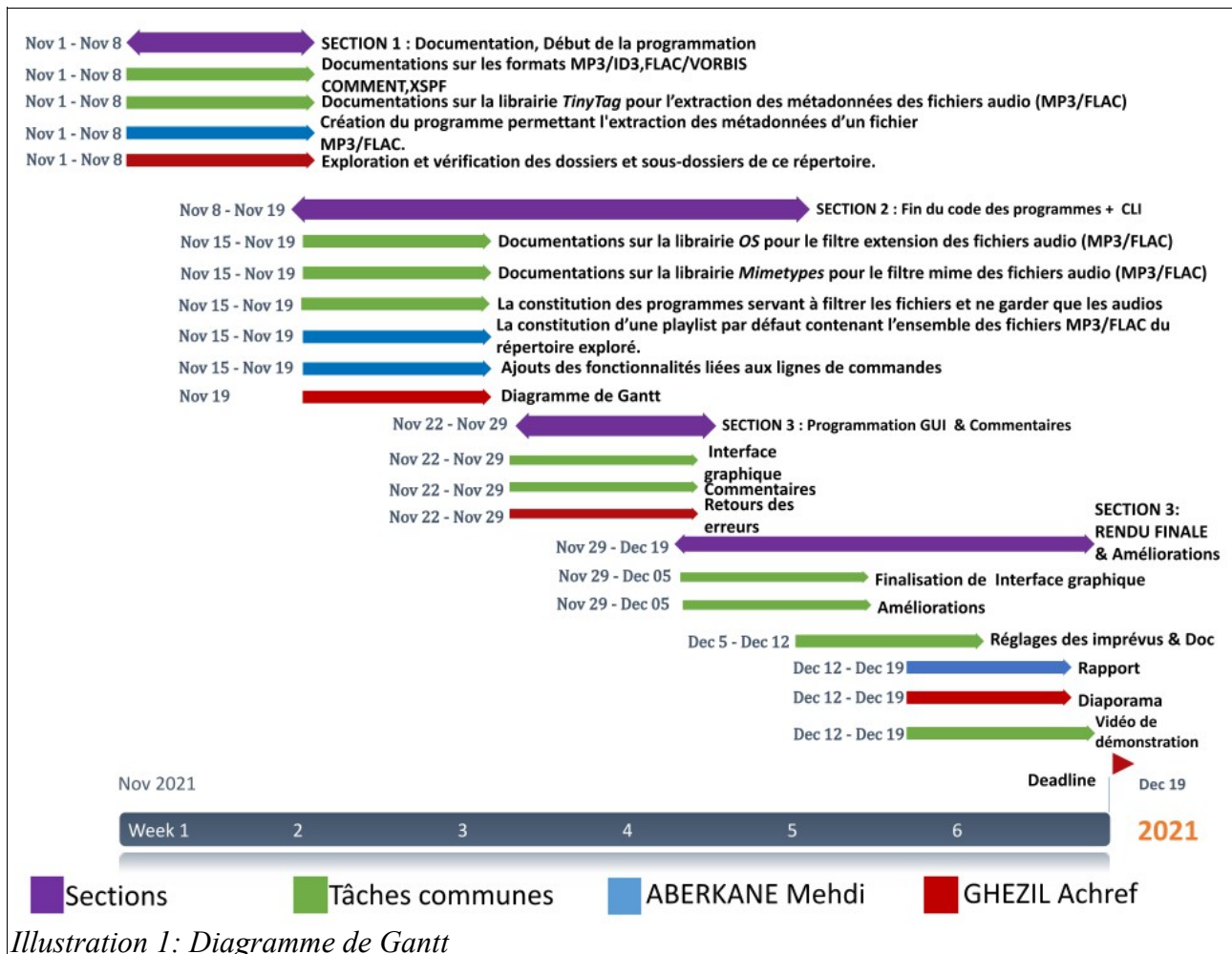


Illustration 1: Diagramme de Gantt

B) Avancement du projet

B.1) En mode console (terminale) «CLI»:

Niveau Avancement côté Terminale nous avons réalisé la totalité des points demandés:

1. Si aucun paramètre passé en argument :(Le nom du fichier compte comme argument 0)

- 'Aucun paramètre passe. Pour plus d'aide tapez -h ou --help'

Commande : `py CLI.py`

```
PS C:\Users\charle\Desktop\W_PYTHON\metadata_mp3-master> py CLI.py
pygame 2.1.0 (SDL 2.0.16, Python 3.9.7)
Hello from the pygame community. https://www.pygame.org/contribute.html
-----
Aucun paramètre passe. Pour plus d'aide tapez -h ou --help
PS C:\Users\charle\Desktop\W_PYTHON\metadata_mp3-master> |
```

2. S'il y a 1 paramètre en argument:

1. Si cette argument est '-h' ou '-help': Affichage des commandes existantes

Commandes: `py CLI.py -h` || `py CLI.py -help`

```
PS C:\Users\charle\Desktop\W_PYTHON\metadata_mp3-master> py CLI.py -h
pygame 2.1.0 (SDL 2.0.16, Python 3.9.7)
Hello from the pygame community. https://www.pygame.org/contribute.html

-----

Les actions possibles sont :
-d [chemin_dossier] pour parcourir un dossier et ses sous dossiers et n'afficher que les fichiers audio (mp3/flac)
-f [chemin_fichier_audio] pour obtenir les métadonnées du fichier '.mp3' ou '.flac' donnée en parametre
-d [chemin_dossier] -o [nom_fichier_playlist_a_créer] pour crée une playlist grâce au fichier et sous dossier du dossier donnée en parametre
-l [chemin_fichier_audio] pour lire une musique
PS C:\Users\charle\Desktop\W_PYTHON\metadata_mp3-master> py CLI.py --help
pygame 2.1.0 (SDL 2.0.16, Python 3.9.7)
Hello from the pygame community. https://www.pygame.org/contribute.html

-----

Les actions possibles sont :
-d [chemin_dossier] pour parcourir un dossier et ses sous dossiers et n'afficher que les fichiers audio (mp3/flac)
-f [chemin_fichier_audio] pour obtenir les métadonnées du fichier '.mp3' ou '.flac' donnée en parametre
-d [chemin_dossier] -o [nom_fichier_playlist_a_créer] pour crée une playlist grâce au fichier et sous dossier du dossier donnée en parametre
-l [chemin_fichier_audio] pour lire une musique
PS C:\Users\charle\Desktop\W_PYTHON\metadata_mp3-master> |
```

2. Si cette argument est seulement '-f' ou '-d' ou '-l': Affichage du message d'erreur suivant *'La commande n'existe pas. Pour plus d'aide tapez -h '*

Commandes: `py CLI.py -f` || `py CLI.py -d` || `py CLI.py -l`

```
PS C:\Users\charle\Desktop\W_PYTHON\metadata_mp3-master> py CLI.py -f
pygame 2.1.0 (SDL 2.0.16, Python 3.9.7)
Hello from the pygame community. https://www.pygame.org/contribute.html

-----

La commande n'est pas complete. Pour plus d'aide tapez -h
PS C:\Users\charle\Desktop\W_PYTHON\metadata_mp3-master> |
```

3. S'il y a 2 paramètres en argument :

1. Si la commande à pour `argv[1]='-f'` et `argv[2]=le chemin d'un fichier audio(MP3/FLAC)`: Affichage des métadonnées dans la console.

Commande: `py CLI.py -f [Chemin_fichier_audio]`

```
PS C:\Users\charle\Desktop\W_PYTHON\metadata_mp3-master> py CLI.py -f Musique/file_mp3.mp3
pygame 2.1.0 (SDL 2.0.16, Python 3.9.7)
Hello from the pygame community. https://www.pygame.org/contribute.html

-----

Vous avez demandé l'affichage des métadonnées dans l'invite de commande, les voilà :

Artist: Ed Sheeran
Album: Ed Sheeran - Bad Habits
Title: Ed Sheeran - Bad Habits
Durée(secs): 230.124
Musique N° 5
Compositeur: Ed Sheeran
Genre: Pop

PS C:\Users\charle\Desktop\W_PYTHON\metadata_mp3-master> |
```

2. Si la commande à pour **argv[1]='-d'** et comme **argv[2]=le chemin d'un dossier**: Affichage des chemins relatif des fichiers audio (MP3/FLAC) dans la console.

Commande : **py CLI.py -d [Chemin_dossier_contenant_des_fichiers_audios]**

```
PS C:\Users\charle\Desktop\W_PYTHON\metadata_mp3-master> py CLI.py -d Musique
pygame 2.1.0 (SDL 2.0.16, Python 3.9.7)
Hello from the pygame community. https://www.pygame.org/contribute.html

-----
Ci-dessous vous avez les fichiers audios du dossier et des sous dossiers de Musique
Musique\damso.mp3
Musique\file_flac.flac
Musique\file_mp3.mp3
Musique\Horizontal.mp3
Musique\Maladresse.mp3
Musique\Mitsubishi.mp3
Musique\mp3.mp3
Musique\test.flac
Musique\Vanille.mp3
PS C:\Users\charle\Desktop\W_PYTHON\metadata_mp3-master>
```

3. Si la commande à pour **argv[1]='-l'** et comme **argv[2]= chemin d'un fichier audio (MP3/FLAC)**. Il y a lecture du morceaux graphiquement donc avec des boutons, mais ses même commandes existe depuis le clavier. Nous avons pensé à cela dans le cas ou si l'utilisateur est en console il puisse l'utiliser sans souris.

Commande : **py CLI.py -l [Chemin_fichier_audio]**

```
PS C:\Users\charle\Desktop\W_PYTHON\metadata_mp3-master> py CLI.py -l Musique/Mitsubishi.mp3
pygame 2.1.0 (SDL 2.0.16, Python 3.9.7)
Hello from the pygame community. https://www.pygame.org/contribute.html

-----
Commandes clavier de notre interface :
- Taper sur H pour afficher les commandes existantes
- Taper sur ESCAPE pour quitter l'interface graphique
- Taper sur P pour:
  - mettre en pause la musique quand elle est lance
  - reprendre la musique quand elle est en pause
- Taper sur R pour recommencer la musique
- Taper sur FLECHE DU HAUT pour augmenter le son
- Taper sur FLECHE DU BAS pour baisser le son
```



4. S'il y a 4 paramètres en argument :

1. Si la commande à pour arguments : **argv[1]='-d'** , **argv[2]= le chemin d'un dossier** , **argv[3]='-o'** et enfin **argv[4]= le nom de la playlist et fichier XSPF**.

Le nom de l'auteur de la playlist vous est ensuite demandé dans la console , suite à cela il y a création et affichage d'une playlist par défaut au format XSPF, avec les fichiers audios (MP3/FLAC) du dossier donné en paramètre. (VOIR IMAGE CI-DESSOUS)

Commande : **py CLI.py -d [Chemin_dossier_contenant_fichiers_audios] -o [Nom_Play]**

```

PS C:\Users\charle\Desktop\W_PYTHON\metadata_mp3-master> py CLI.py -d Musique/SOUS_MUSIQUE -o TOP_FR
pygame 2.1.0 (SDL 2.0.16, Python 3.9.7)
Hello from the pygame community. https://www.pygame.org/contribute.html

-----
Entrer le nom de l'auteur de la playlist : Med
LE FICHIER XSPF EXISTE DEJA. REMISE A ZERO DU FICHIER TOP_FR.xspf :
SUCCES
ECRITURE DE LA PLAYLIST TOP_FR DE Med :
ECRITURE TERMINER.
AFFICHAGE DE LA PLAYLIST TOP_FR :
<?xml version="1.0" encoding="UTF-8"?>
<playlist version="1" xmlns="http://xspf.org/ns/0/">
  <title> TOP_FR </title>
  <creator>Med</creator>
  <info> .... </info>
  <trackList>
    <track>
      <creator> Martyn Brabbins: BBC Scottish Symphony Orchestra </creator>
      <title> MacCunn: The Lay of the Last Minstrel </title>
      <duration> 87 </duration>
      <album> MacCunn: Land of the Mountain and the Flood </album>
      <location>file:Musique/SOUS_MUSIQUE\mp3</location>
    </track>
    <track>
      <creator> Howard Shelley </creator>
      <title> Clementi: Piano Sonata in D major, Op 25 No 6 </title>
      <duration> 166 </duration>
      <album> Clementi: The Complete Piano Sonatas, Vol. 4 </album>
      <location>file:Musique/SOUS_MUSIQUE\test.flac</location>
    </track>
  </trackList>
</playlist>
PS C:\Users\charle\Desktop\W_PYTHON\metadata_mp3-master>

```

B.2) En mode graphique «GUI» :

1. L'affichage des méta-données d'un fichier MP3 ou FLAC donnée (possibilité de sauvegarde).

Pour afficher les métadonnées depuis notre interface graphique vous trouverez sur notre seconde fenêtre le bouton nommé **«Extraction des métadonnées»** (seconde fenêtre= fenêtre qui s'ouvre lorsqu'on clique sur le bouton **« LANCER »**) (**VOIR IMAGE ANNEXE**)

2. Génération d'une playlist par défaut contenant l'ensemble des fichiers MP3 et FLAC du répertoire exploré. (possibilité de sauvegarde et ouverture d'une playlist enregistrée précédemment).

Pour générer une playlist par défaut depuis notre interface graphique vous trouverez sur notre seconde fenêtre le second bouton nommé **«Playlist par répertoire»**.

Une troisième fenêtre va s'ouvrir en vous demandant de sélectionner le dossier que vous souhaitez utiliser pour créer la playlist , si vous n'en sélectionne aucune la fenêtre s'ouvrira jusqu'à temps que cela arrive.

Après l'avoir sélectionné, 2 autres fenêtre vont venir s'ouvrir:

1. La première demandant le nom de la playlist.
2. La seconde demandant le nom de l'auteur de celle-ci.

Quand tout cela est fait la playlist est généré automatiquement dans le répertoire courant. (**VOIR IMAGE ANNEXE**)

3. Génération d'une playlist personnalisé contenant les fichiers MP3 et FLAC sélectionné dans le répertoire exploré. (possibilité de sauvegarde et ouverture d'une playlist enregistrée précédemment).

Pour générer une playlist personnalisé depuis notre interface graphique vous trouverez sur notre seconde fenêtre le second bouton nommé **«Playlist personnalisée»**.

Une troisième fenêtre va s'ouvrir en vous demandant de sélectionner le ou les morceaux que vous souhaitez utiliser pour créer la playlist , si vous n'en sélectionne aucune la fenêtre s'ouvrira jusqu'à temps que cela arrive.

Après l'avoir sélectionné, 2 autres fenêtre vont venir s'ouvrir :

1. La première demandant le nom de la playlist.

2. La seconde demandant le nom de l'auteur de celle-ci.

Quand tout cela est fait la playlist est généré automatiquement dans le répertoire courant.

(*VOIR IMAGE ANNEXE*)

4. Améliorations de bases :

1. Offrir la possibilité d'écouter le morceau de musique.

Pour pouvoir écouter un morceau de musique depuis notre interface graphique vous trouverez sur notre seconde fenêtre le second bouton nommé «*Jouer un morceau*».

Une troisième fenêtre va alors s'ouvrir en vous demandant de sélectionner le morceaux que vous souhaitez écouter (si vous n'en sélectionne aucune la fenêtre s'ouvrira jusqu'à temps que cela arrive) et après l'avoir sélectionné vous aurez une interface graphique composé de la couverture et la musique du morceau choisit. Depuis laquelle vous pourrez faire de nombreuses actions depuis des boutons (Mettre pause / Reprendre, Augmenter/Baisser le volume, Recommencer le morceaux, Quitter l'interface graphique).

(*VOIR IMAGE ANNEXE*)

2. Extraire également l'image (*thumbnail*) de couverture (*cover*) de l'album correspondant au morceau et l'afficher à l'utilisateur.

Pour l'extraction de la couverture d'un morceau sélectionné côté graphique , on la retrouve à 2 endroits différents :

1. Dans la fenêtre après avoir cliquez sur le bouton «*Jouer le son*» , on y le morceau qui est joué et sa couverture.
2. Dans la fenêtre après avoir cliquez sur «*Extraction des métadonnées*» , on y retrouve l'image et les autres métadonnées du morceau donné.

(*VOIR IMAGE ANNEXE*)

C) Architecture de notre application :

Fichier '.py'	Fonction(s)
meatadata.py	<i>metadata(filepath,Affichemeta) RecupMetadonneeListe(filepath) Affiche_Metadonnee(filepath)</i>
parcours_directory.py	<i>parcour_directory(path)</i>
Creatplaylist.py	<i>VerifExtension(chemin) VerifMime(chemin), RecupSong(liste) XsplPlaylist(TitrePlay, AuteurPlay , liste_abspath_son)</i>
dialog.py	<i>GetFile() getFilePL() getDir() getFiles()</i>
CLI.py	<i>main()</i>
playsound.py	<i>playsond(path_son)</i>
GUI.py	<i>draw_text() main()</i>
menuGUI.py	<i>main_menu()</i>
metadataUI.py	<i>metadataScreen(screen)</i>
readPlayListGUI.py	<i>draw_text(text, font, color, surface, x, y) PlayListScreen(screen)</i>

Conclusion

A) Récapitulatif de la réalisation :

Tâche	Réalisée
Extraction des Métadonnées d'un fichier audio (MP3/FLAC) donné. (CLI & GUI)	✓
Parcours et récupération des chemins des fichiers audio d'un dossier donné. (CLI & GUI)	✓
Création d'une playlist par défaut. (CLI & GUI)	✓
Création d'une playlist personnalisée. (GUI)	✓
<u>Amélioration des fonctionnalités :</u> Lecture d'un morceau sélectionné. (CLI & GUI)	✓
<u>Amélioration des fonctionnalités:</u> Extraction de la couverture d'un fichier audio. (MP3/FLAC) (GUI)	✓
<u>Pour aller encore plus loin :</u> Modification et sauvegarde des tags d'un morceau. (GUI & CLI)	✗
<u>Pour aller encore plus loin :</u> Utilisation d'un API WEB pour récupérer les informations complètes d'un album. (GUI & CLI)	✗

B) Conclusion & Améliorations

Pour conclure, nous trouvons que notre application à donc atteint une qualités plus que satisfaisante car elle est capable de réaliser toutes les fonctionnalités principales demandées dans le sujet et également toutes les fonctionnalités que nous avons envisagées.

Bien évidemment même si elle est satisfaisante elle peut encore être améliorée, comme par exemple en :

- Ajoutant les fonctionnalités pour allez encore plus loin :
 - Ajoutant la possibilité de modifier les métadonnées d'un fichier MP3 sélectionné.
 - Ajoutant une API , pour avoir un tas d'autres chansons.
- Ajoutant la possibilité de sélectionner le répertoire dans lequel la playlist ou les métadonnées vont être sauvegardées (Autre part que dans le répertoire courant).
- Ajouter lorsqu'on lance un morceau , la visualisation de la barre pour avancer et reculer dans le morceau.
- Ajout d'un style alternatif.
- Et pleins d'autres

A) Mock-up déroulement du GUI :

