

Réalisation d'une interface graphique pour un logiciel de traitement vidéo

Adrien Palmieri
Alexandre Hublau
Jean-Christophe Chevrier
Alexandre Da Silva Carmo



Sommaire

1. Présentation du logiciel FFMPEG
2. Problématique : facilité d'utilisation
3. Présentation du projet
4. Comparaison des langages envisagés
5. Présentation des solutions existantes
6. Premières fonctionnalités à implémenter
7. Conclusion



Présentation de FFMPEG

- ensemble de logiciels libres
 - traiter des flux vidéos/audio (contenu multimédia)
 - Le pack est composé de :
 - ffmpeg (éditer des vidéos, des flux audio, convertir)
 - ffplay (lecture)
 - ffserver (un serveur http permet de faire de la diffusion en continu)
-
- supporte de nombreux formats de fichiers multimédia





Problématique : facilité d'utilisation

- outil en ligne de commandes
- documentation et logiciel en anglais
- énormément de fonctionnalités
- commandes complexes
- opérations longues : temps perdu en cas d'erreur

```
ffmpeg -i fichierSource.avi -vframes 1000 -vf "transpose=1" fichierDestination.mp4
```



Présentation du projet

Objectif : Créer une interface graphique pour FFmpeg qui permet à un utilisateur lambda d'utiliser le logiciel.

Problématique : trouver le rapport optimal fonctionnalités / interface

- Le programme doit remplir un maximum de fonctions
- L'interface doit rester simple et intuitive pour l'utilisateur
- Le programme peut faire plusieurs appels simultanément pour optimiser le temps de traitement

2 solutions techniques envisagées :

- utiliser le langage Java et effectuer des appels système à FFmpeg
- utiliser le langage C++ et implémenter les librairies de FFmpeg

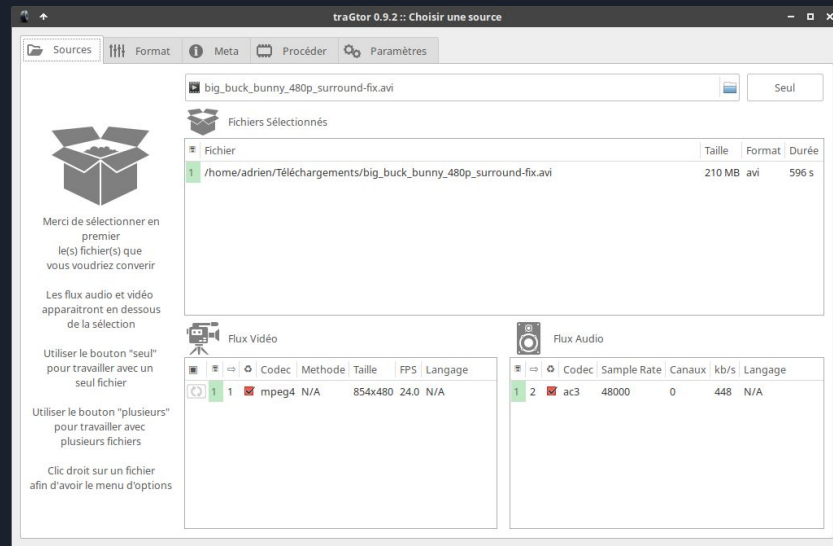
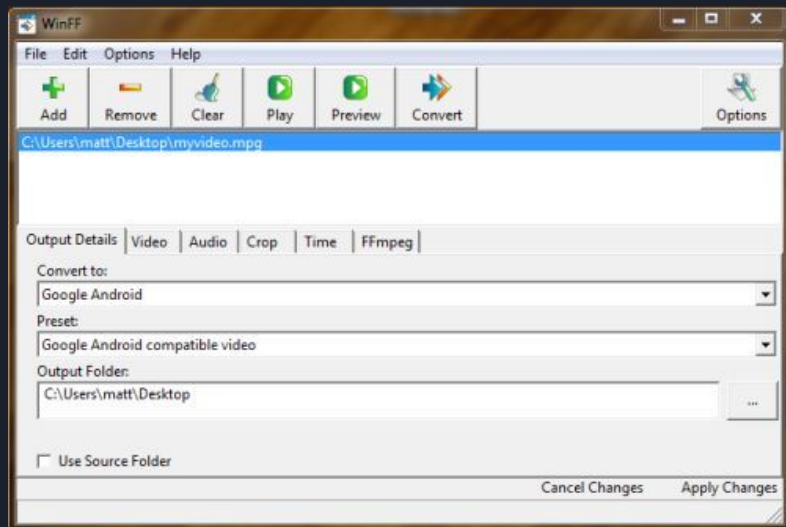


Langage envisagés

	Java	C++
Avantage	<ul style="list-style-type: none">- Connaissance en java et IHM (API awt et swing)- Des exemples existent déjà (ffmpeg-cli-wrapper)	<ul style="list-style-type: none">- FFmpeg dispose de bibliothèques en C++ ce qui simplifie l'utilisation- Possibilité de traiter les erreurs générées plus facilement
Inconvénient	<ul style="list-style-type: none">- Obligation de faire des appels systèmes pour exécuter les commandes	<ul style="list-style-type: none">- Langage non maîtrisé- IHM non maîtrisé

Présentation des solutions existantes

Windows : WinFF



Linux : traGtor



Premières fonctionnalités à implémenter

Pour un meilleur rapport fonctionnalités interface, nous avons choisi d'implémenter les fonctionnalités suivantes :

- Conversion (video <-> video, video <-> son, video <-> image)
- Lecture
- Modification de la résolution
- Rogner
- Pivoter



Conclusion

- avoir une IHM intuitive
- minimiser le temps de traitement
- avoir un maximum de fonctionnalités
- développer dans un environnement connu
- flexibilité :
 - choisir l'influence de la conversion sur le système
 - lancer des opérations en tâche de fond