

## Partie 1

Pour chaque question sur les lignes de commande à fournir, je me suis permis d'en mettre deux

1. une pour les formats ogv
2. une générale pour tous types de formats

### Question 1)

#### OGV

OGV est une extension de nom de fichier vidéo, provenant des formats Ogg et Vorbis' crée par Xiph.Org Foundation

#### VORBIS

Vorbis est un algorithme pour compresser et décompresser des fichiers audios numériques. Il est ouvert, libre et sans brevet et il supporte le multi-canal.

#### THEORA

Theora est un format de compression vidéo (format de vidéo ouvert et sans brevets, rivalisant avec les formats MPEG4, WMV et RealVideo).

### Question 2)

#### a) Lire la vidéo et le son

1. `gst-launch filesrc location=video.ogv ! oggdemux name=ogg ogg. ! queue ! theoradec !  
ffmpegcolospace ! autovideosink ogg. ! queue ! vorbisdec ! audioconvert ! autoaudiosink`
2. `gst-launch filesrc location=video.ogv ! decodebin name=decod decod. ! queue !  
ffmpegcolospace ! autovideosink decod. ! queue ! audioconvert ! autoaudiosink`

#### b-)lire la vidéo en remplaçant l'audio par un autre fichier audio

1. `gst-launch filesrc location=video.ogv ! oggdemux name=ogg ogg. ! theoradec !  
ffmpegcolospace ! autovideosink ogg. ! filesrc location=audio.mp3 ! mad ! audioconvert !  
autoaudiosink`
2. `gst-launch filesrc location=video.ogv ! decodebin name=decod decod. ! ffmpegcolospace !  
autovideosink decod. ! filesrc location=audio.mp3 ! mad ! audioconvert ! autoaudiosink`

## Question 3) Lire la vidéo, le son, et les sous-titres

1. `gst-launch filesrc location=video.srt ! subparse ! overlay. filesrc location=video.ogv !  
oggdemux name=decode ! queue ! theoraec ! ffmpegcolospace ! subtitleoverlay  
name=overlay ! autovideosink decode ! queue ! vorbisdec ! audioconvert ! autoaudiosink`
2. `gst-launch filesrc location=video.srt ! subparse ! overlay. filesrc location=video.ogv !  
decodebin name=decode ! queue ! ffmpegcolospace ! subtitleoverlay name=overlay !  
autovideosink decode ! queue ! audioconvert ! autoaudiosink`

## Question 4) Fichier contenant la ( vidéo + son + sous titres)

### a-) son au choix

1. `gst-launch filesrc location=video.srt ! subparse ! overlay. filesrc location=video.ogv !  
oggdemux name=decode \  
decode ! queue ! theoraec ! ffmpegcolospace ! subtitleoverlay name=overlay ! theoraenc !  
m. \  
decode ! filesrc location=audio.mp3 ! mad ! audioconvert ! vorbisenc ! m. \  
oggmux name=m ! filesink location=video_soustitres_partie1.ogv`
2. `gst-launch filesrc location=video.srt ! subparse ! overlay. filesrc location=video.ogv !  
decodebin name=decode \  
decode ! queue ! ffmpegcolospace ! subtitleoverlay name=overlay ! theoraenc ! m. \  
decode ! filesrc location=audio.mp3 ! mad ! audioconvert ! vorbisenc ! m. \  
oggmux name=m ! filesink location=video_soustitres_partie1.ogv`

### a-) son d'origine

1. `gst-launch filesrc location=video.srt ! subparse ! overlay. filesrc location=video.ogv !  
oggdemux name=decode \  
decode ! queue ! theoraec ! ffmpegcolospace ! subtitleoverlay name=overlay ! theoraenc !  
m. \  
decode ! queue ! vorbisdec ! audioconvert ! vorbisenc ! m. \  
oggmux name=m ! filesink location=video_soustitres_partie1.ogv`
2. `gst-launch filesrc location=video.srt ! subparse ! overlay. filesrc location=video.ogv !  
decodebin name=decode \  
decode ! queue ! ffmpegcolospace ! subtitleoverlay name=overlay ! theoraenc ! m. \  
decode ! queue ! audioconvert ! vorbisenc ! m. \  
oggmux name=m ! filesink location=video_soustitres_partie1.ogv`

## Partie 2

L'application développée peut lire différents formats de vidéo (ogg, ogv, avi).

Fonctionnalité rajouté : stop, replay

Afin de développer le lecteur audio vidéo, j'ai procédé en 5 étapes distinctes :

1. la première fut l'analyse des besoins et la documentation sur l'ensemble des librairies exigées pour la réalisation du projet
2. la seconde étapes à la spécification
3. la troisième aux développements des modules
4. la quatrième étape à l'implémentation de l'interface graphique
5. la dernière étape pour les tests de l'application

## Librairies utilisées

### gstreamer-0.10

GStreamer est une bibliothèque logicielle de manipulation de sons et d'images (appelée aussi framework multimédia) distribué sous licence libre, écrite en C, initialement développée pour proposer une solution capable de concurrencer QuickTime et DirectShow sur GNU/Linux.

### gstreamer-interfaces-0.10

Cette librairie permet de disposer de l'interface XOverlay, qui permet à l'application contenant la sortie la vidéo du flux de choisir et fixer son propre conteneur afin que l'interface graphique créée par GTK puisse contenir la fenêtre de rendu de Gstreamer.

### gtk+-2.0

GTK+ (The GIMP Toolkit) est un ensemble de bibliothèques logicielles, c'est-à-dire un ensemble de fonctions permettant de réaliser des interfaces graphiques. Cette bibliothèque a été développée originellement pour les besoins du logiciel de traitement d'images GIMP. GTK+ est maintenant utilisé dans de nombreux projets, dont les environnements de bureau GNOME, Xfce, Lxde et ROX. GTK+ est un projet libre (licence GNU LGPL 2.1) et multiplate-forme.

## Description des fichiers

Le code de l'application est contenu dans quatre fichiers distincts dont (1 fichier header et 3 fichiers sources) :

1. **pmp.h** : ce fichier contient les prototypes des différentes fonctions ainsi que la structure de données utilisées dans l'application. La structure StreamData contient essentiellement l'ensemble des éléments nécessaires à la création du pipeline pour la visualisation de la vidéo, d'un booléen pour activer et désactiver les sous-titres, un slider pour traquer l'évolution de la lecture de la vidéo, un entier pour identifier l'ID du slider pour le rafraîchissement du slider sur l'interface graphique ainsi qu'un la durée de la vidéo en nanosecondes.
2. **pmp\_cb.c** : ce fichier contient l'ensemble des fonctions de type CALLBACK, appelé lors du déclenchement d'un événement précis.
3. **pmp\_ui.c** : ce fichier contient l'ensemble des fonctions nécessaires à la création de l'interface graphique de l'application.

4. **main.c** : ce fichier contient le programme principal de l'application.

## Screen

## BUG

Après la fin de la lecture ou en cliquant sur le bouton stop puis play la vidéo ne se relance pas parfois quand les sous-titres sont activés en fonction des systèmes d'exploitations. Je n'ai toujours pas trouvé de solution à ce bug. Lorsque les sous-titres ne sont pas activés on peut arrêter et relancer la vidéo à notre guise. (une solution , je suis preneur aob.diallo@gmail.com)