

OIM

Rapport de Projet 2

Ligne de commande

Question 1

Le format OGV est un format proposé par le projet ogg. Il s'agit d'un format libre, donc sans restriction dû aux brevets. Il disposent de deux codec : Vorbis et Theora. Vorbis étant l'encodage du son et Theora l'encodage de la video. A noté qu'il existe une multitude d'extension de fichier du format OGG ou OGV : .ogv, .ogg, .oga. Le .ogv que nous utilisons par la suite est l'une des possibilité pour les fichiers vidéos.

Question 2

La commande suivante permet de lire le fichier video.ogv avec son et vidéo.

```
1 | gst-launch filesrc location=video.ogv ! oggdemux name=demux demux. ! queue ! theoradec ! ↵  
    autovideosink demux. ! queue ! vorbisdec ! autoaudiosink
```

Cette seconde commande permet de lire le flux vidéo du fichier video.ogv avec le flux audio du fichier cranberries.mp3 fourni. Pour cela la sortie audio du demuxer (oggdemux ici) n'est pas utilisé, à la place nous utilisons la sortie d'un autre filesrc qui fournira la musique.

```
1 | gst-launch filesrc location=video.ogv ! oggdemux name=demux demux. ! queue ! theoradec ! ↵  
    autovideosink filesrc location=cranberries.mp3 ! queue ! mad ! autoaudiosink
```

Question 3

La commande suivante permet de lire le fichier vidéo video.ogv avec sa piste de sous-titres video.srt. Pour cela, il suffit de redirigé la sortie vidéo ainsi que le flux de sortie vers le plugin subtitleoverlay qui fusionnera ces deux flux pour les envoyer vers l'autovideosink.

```
1 | gst-launch subtitleoverlay name=overlay ! autovideosink filesrc location=video.ogv ! oggdemux  
    name=demux demux. ! queue ! theoradec ! overlay. demux. ! queue ! vorbisdec ! autoaudiosink  
    filesrc location=video.srt ! subparse subtitle-encoding=UTF-8 ! overlay.
```

Nous pouvons voir une variante avec le bin "decodebin" qui permet de gérer tous les formats. Les commandes précédentes n'étaient valables que pour des formats ogg.

```
gst-launch subtitleoverlay name=overlay ! autovideosink filesrc location=video.ogv ! decodebin
name=demux demux. ! queue ! overlay. demux. ! queue ! autoaudiosink filesrc ←
location=video.srt ! subparse subtitle-encoding=UTF-8 ! overlay.
```

Question 4

Cette commande permet de fusionner les deux fonctionnalités des question 3 et 4 et d'enregistrer la sortie dans un fichier. Pour cela, il faut effectuer les mêmes traitements que précédemment, tout en rajoutant un muxer. La sortie video et audio sera envoyé au muxer qui fusionnera les deux flux puis enverra le flux fusionné vers le plugin filesink qui permet la sauvegarde dans un fichier.

```
1 | gst-launch subtitleoverlay name=overlay ! theoraenc ! oggmux name=muxer ! filesink ←
location=video_soustitres_partiel.ogv filesrc location=video.ogv ! oggdemux name=demux ←
demux. ! queue ! theoradec ! overlay. filesrc location=cranberries.mp3 ! queue ! mad ! ←
audioconvert ! vorbisenc ! muxer. filesrc location=video.srt ! subparse ←
subtitle-encoding=UTF-8 ! overlay.
```

Programmation C

Organisation des fichiers

Les sources du programme sont stockées dans le répertoire src de cette façon :

main.c : fichier contenant le main.

ui.c : fichier contenant les fonctions permettant la création de l'interface. Une seule fonction est utilisable de l'extérieur : get_ui.

av.c : l'ensemble des fonctions gérant l'audio et la vidéo avec gstreamer.

Génération des sous-titres

La génération des sous-titres à partir d'une vidéo est assez délicat. Pour cela nous devons d'abord séparer le son de la vidéo. A partir de la piste audio, nous devons réussir à différencier les paroles des autres sons. Pour cela, il est possible d'effectuer un filtre qui ne récupérera que les sons d'une certaine fréquence : celle de la voix humaine (entre 80 et 1500 Khz).

Il s'agit ensuite de pouvoir récupérer les mots. Pour cela il faudrait avoir accès à un dictionnaire de correspondances entre les sons et les mots. Nous pourrions alors séparer les mots grace aux silences entre eux et faire une recherche dans le dictionnaire. Nous aurions alors l'ensemble des mots prononcé dans la piste audio.