

École de technologie supérieure	Trimestre	Hiver 2016
Département de génie logiciel et des TI	Enseignant	Stéphane Coulombe
	Préalable	LOG120
	Crédits	4

GTI310 Structure de données multimédias

Premier Travail Pratique

Sujet : Introduction au multimédia

Description

Ce laboratoire a pour but de vous initier au multimédia. Il consiste en plusieurs tâches exploratoires. Premièrement, vous pourrez explorer les aspects de compression et de qualité visuelle des images avec le logiciel ImageMagick. Vous développerez ensuite un module de lecture d'images pour une application de présentation d'images. Finalement, vous effectuerez des changements de caractéristiques de fichiers audio et vous en évalueriez l'impact sur la qualité à l'aide du logiciel Audacity.

Logiciel ImageMagick

ImageMagick est un logiciel comprenant un ensemble d'utilitaires en ligne de commande permettant notamment de lire les caractéristiques et de convertir des images et cela pour un très grand nombre de formats. De plus, ImageMagick a l'avantage d'être multiplate-forme (Linux, Windows, Mac Os X, etc.). Pour de plus ample information sur ImageMagick, veuillez visiter le site www.imagemagick.org. Le logiciel est disponible sur le site Moodle.

Installation d'ImageMagick

Ouvrez un terminal, téléchargez et installez ImageMagick en suivant les instructions indiquées sur cette page : <http://www.imagemagick.org> (voir l'onglet Download). Vous devez opter pour l'approche alternative, celle suivant immédiatement la phrase : « *Alternatively, you can download the ImageMagick Mac OS X distribution we provide :* » Vous devez garder votre terminal ouvert en tout temps. Si vous fermez votre terminal, vos chemins (paths) seront perdus et le terminal sera incapable d'exécuter les commandes d'ImageMagick. Vous devez alors rentrer de nouveau vos chemins comme suit :

- export MAGICK_HOME="\$HOME/ImageMagick-6.9.3"
 - export MAGICK_HOME="\$HOME/EMPLACEMENTVOSMEDIAS"
- export PATH="\$MAGICK_HOME/bin:\$PATH"
- export DYLD_LIBRARY_PATH="\$MAGICK_HOME/lib/"

À faire :

Téléchargez le fichier media-TP1.zip depuis le site web du cours et décompressez ce fichier. Dans votre terminal, placez-vous dans le dossier media-TP1/Partie1_ImageMagick.

1. Exécutez la commande suivante : *identify -verbose 'biere.gif'*. La commande *identify* d'ImageMagick permet d'obtenir de l'information sur une ou plusieurs images (<http://www.imagemagick.org/script/identify.php>).
 - a. Déterminez le format du fichier (« Format »).
 - b. Déterminez le type de compression (« Compression »).
 - c. Déterminez la taille de l'image en pixels (« Geometry »).
 - d. Déterminez le nombre de couleurs de la palette (« Colormap ») et le nombre de bits par pixels (BPP).
 - e. Notez toutes les couleurs uniques (indices et valeurs RVB) de la palette. Ces couleurs sont indiquées sous la ligne « Colormap ».
 - f. Notez les valeurs RVB de la couleur transparente (« Transparent Color »).
 - g. Déterminez le nombre de couleurs uniques (vous pouvez aussi utiliser la commande suivante : *identify -format "%k" 'biere.gif'*) et le nombre de bits par pixels (BPP) réellement nécessaire pour stocker l'image.
 - h. Pourquoi le logiciel affiche-t-il une valeur différente de nombre de couleurs en d) et en g) ?
 - i. Déterminez la taille (en octets) de l'image compressée sur le disque (« Filesize »).
 - j. Calculez la taille de mémoire requise pour stocker les données d'image décompressée (formée d'indices) et la palette de couleur définie comme (en octets) :

$$\text{Taille image GIF} = \text{TailleX} \times \text{TailleY} \frac{\text{BPP}}{8} + \text{NBcouleursPalette} \times 3 \text{ octets}$$

- k. On définit le facteur de compression par : $\frac{\text{Taille image originale}}{\text{Taille image compressée}}$

Considérez la taille originale comme celle calculée en j) et calculez le facteur de compression.

- l. Si l'image originale était composée de pixels RVB (i.e. une image avec 24 bits par pixel) plutôt que représentée avec une palette de couleurs. Quelle serait sa taille en octets? Quel serait le facteur total de compression (i.e. refaites le calcul en j) en considérant que la taille de l'image originale est celle de la représentation en format RVB)?

- m. Comparez le facteur de compression obtenu en représentant les données avec une palette de couleur, résultat de k), plutôt qu'en pixels RVB, résultat de l).
 - n. Discutez brièvement de l'utilité d'utiliser une palette de couleur par rapport à la représentation RVB et des inconvenients.
2. La commande « *convert* » d'ImageMagick permet de convertir une image (<http://www.imagemagick.org/script/convert.php>). À l'aide de cette commande, sauvegardez l'image GIF en format JPEG avec facteur de qualité 40.
- a. Notez la ligne de commande que vous avez utilisée pour effectuer la conversion.
 - b. Notez la taille du fichier résultant en octets.
 - c. Comparez la taille et la qualité visuelle (faites un zoom pour mieux comparer les images) du nouveau fichier JPEG par rapport à GIF.
3. Pour cette partie, vous utilisez l'image « Chien.gif ».
- a. Trouvez le facteur de qualité à utiliser pour obtenir une image en format JPEG de la même taille que « Chien.gif ». (à 5% près).
 - b. Comparez la qualité visuelle entre les versions GIF et JPEG (regardez particulièrement les endroits de détails et de transitions).
 - c. Codez l'image en format JPEG avec un facteur de qualité de 20. Notez la taille de l'image. Comparez la qualité et la taille au format GIF et concluez.
4. Pour cette partie, vous utiliserez l'image « kodim18.bmp ».
- a. Quelle est la taille de l'image en pixels et le nombre de bits par pixels (« Depth » × 3) ?
 - b. Calculez la taille, en octets, d'une telle image non compressée.
 - c. Comparez avec la taille du fichier de l'image. Expliquez cette différence.
 - d. Sauvegardez en format JPEG avec facteur de qualité 80 (assez typique). Notez la taille du fichier résultant et la qualité visuelle.
 - e. Calculez le facteur de compression.
 - f. Pour des facteurs de qualité de 1, 10, 20, 30, 40, 50, 60, 70, 80, 90 et 100, faites un graphique de la taille du fichier JPEG en fonction du facteur de qualité. Note : conservez ces fichiers pour l'étape g. Prenez soin d'identifier vos axes et de titrer le graphique.
 - g. Pour les mêmes facteurs de qualité, faites un graphique de la qualité visuelle (sur une échelle de 0 à 10, selon votre opinion) du fichier JPEG en fonction du facteur de qualité. Notez la présence de patrons de blocs visibles. Prenez soin d'identifier vos axes et de titrer le graphique.
 - h. Est-ce une bonne idée d'utiliser un facteur de qualité de 90 ou 100? Pourquoi?
 - i. Sauvegardez le fichier en format GIF (sans couleur transparente) et notez la taille du fichier.

- j. Réduisez le nombre de couleurs du fichier GIF (voir l'option « colors » de convert) à 16 couleurs (4 BPP) en utilisant la méthode Floyd-Streinberd (voir l'option « dither » de convert). Évaluez la qualité et notez la taille du fichier.
- k. Réduisez le nombre de couleurs du fichier GIF créé en i) (voir l'option « colors » de convert) à 8 couleurs (3 BPP) en utilisant la méthode Floyd-Streinberd (voir l'option « dither » de convert). Évaluez la qualité et notez la taille du fichier.
- l. Pour cette image, est-ce que JPEG ou GIF est un meilleur choix pour la compression? Pourquoi?
- m. Notez que l'on ne peut pas généraliser les résultats. Chaque image a ses propres propriétés et le facteur de qualité acceptable pour JPEG et le taux de compression en dépendent. Néanmoins, de façon générale, JPEG et GIF se comportent tels que dans les exemples étudiés. Selon vous, est-ce que JPEG ou GIF est le plus approprié pour des images naturelles (ex. photos numériques)? Qu'en serait-il pour des petits graphiques ou icônes?

Projet ImageViewer

Le projet « ImageViewer » est une interface graphique permettant d'afficher des images. L'application, écrite en Java, a été développée de façon à ce qu'il soit possible d'ajouter du support pour d'autres formats d'images.

La version qui vous est fournie ne supporte aucun format d'image. Vous devez ajouter le code nécessaire afin qu'il soit possible d'afficher des images de format BMP. Vous devez seulement supporter les images BMP non-compressé de 24 bits par pixels.

Ce type de fichier est bâti de la façon suivante :

Entête :

Étiquette	Description	Nombre d'octets
bfType	Nombre magique correspondant à l'utilisation du fichier BMP	2
bfSize	Taille du fichier en octets	4
bfReserved1	Réservé	2
bfReserved2	Réservé	2
bdOffBits	Adresse de départ des données BMP	4
biSize	Taille de l'entête BMP. Typiquement 40 octets	4
biWidth	Largeur horizontal de l'image, en pixels.	4
biHeight	Largeur vertical de l'image, en pixels.	4
biPlanes		2
biBitCount	Le nombre de bits par pixel.	2
biCompression	Type de compression. Typiquement 0.	4
biSizeImage	Taille de la partie données du BMP. En octets, doit être un multiple de 4.	4
biXPelsPerMeter	Résolution horizontale	4
biYPelsPerMeter	Résolution verticale	4
biClrUsed	Nombre de couleurs utilisées.	4
biClrImportant	Nombre de couleurs importantes.	4

Données :

Bleu Vert Rouge Bleu Vert Rouge Bleu Vert Rouge Bleu Vert Rouge ...

Il est important de noter que chaque ligne d'une image doit contenir un nombre d'octets multiple de 4. Lorsque les octets des pixels ne respecte pas cette règle, des 0 de bourrage sont ajoutés (*padding*, en anglais) à la fin de chacune des lignes. Pour plus d'informations sur BMP, visitez le site suivant :

<http://atlc.sourceforge.net/bmp.html>

Le diagramme de classes ci-dessous vous présente l'architecture actuelle du programme. Les tâches que vous devez compléter sont annotées sur le dessin.

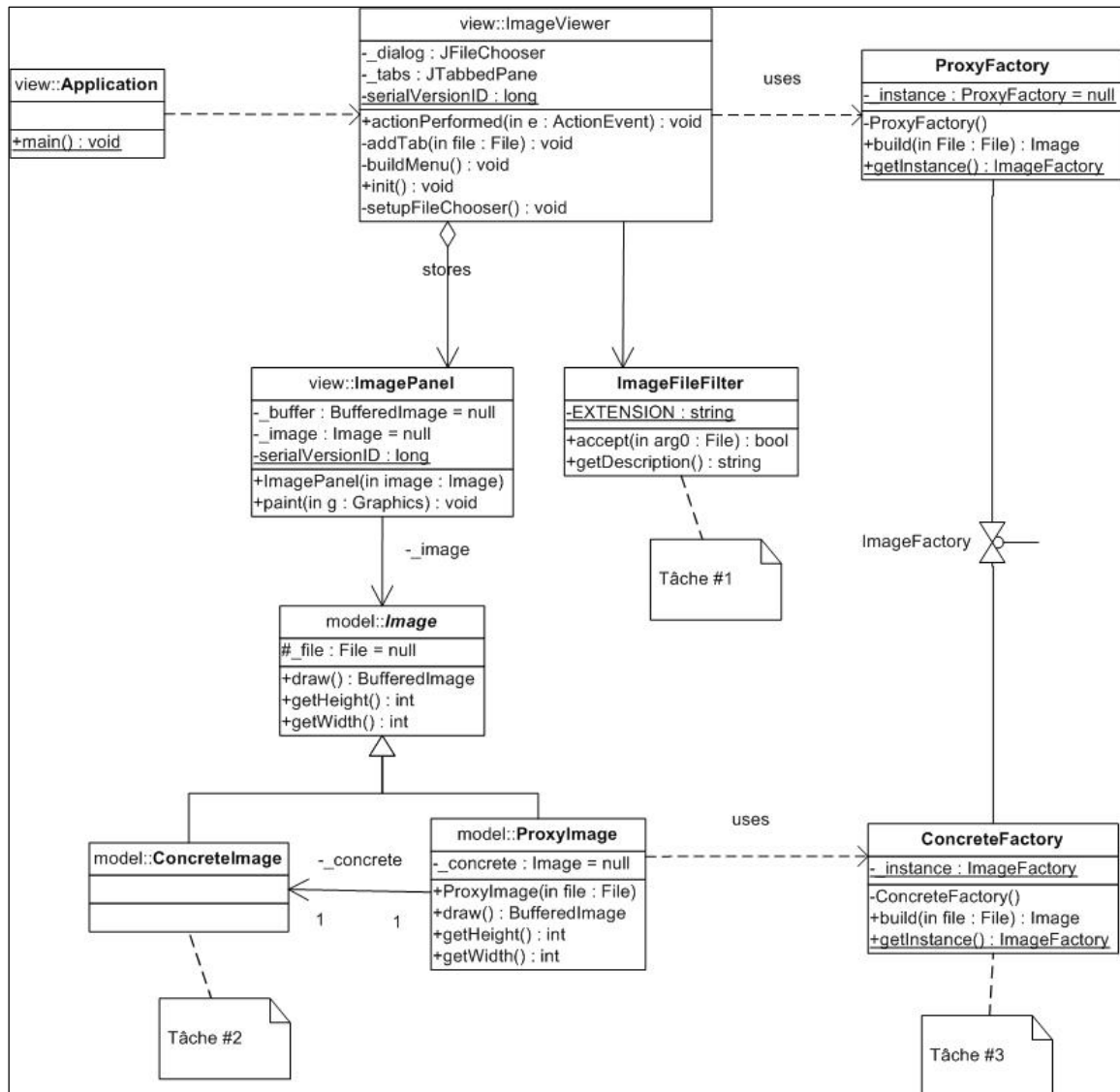


Figure 1 Diagramme de classes

Pour vous aider à mieux comprendre les liens entre les objets du programme, étudiez le diagramme de séquence suivant :

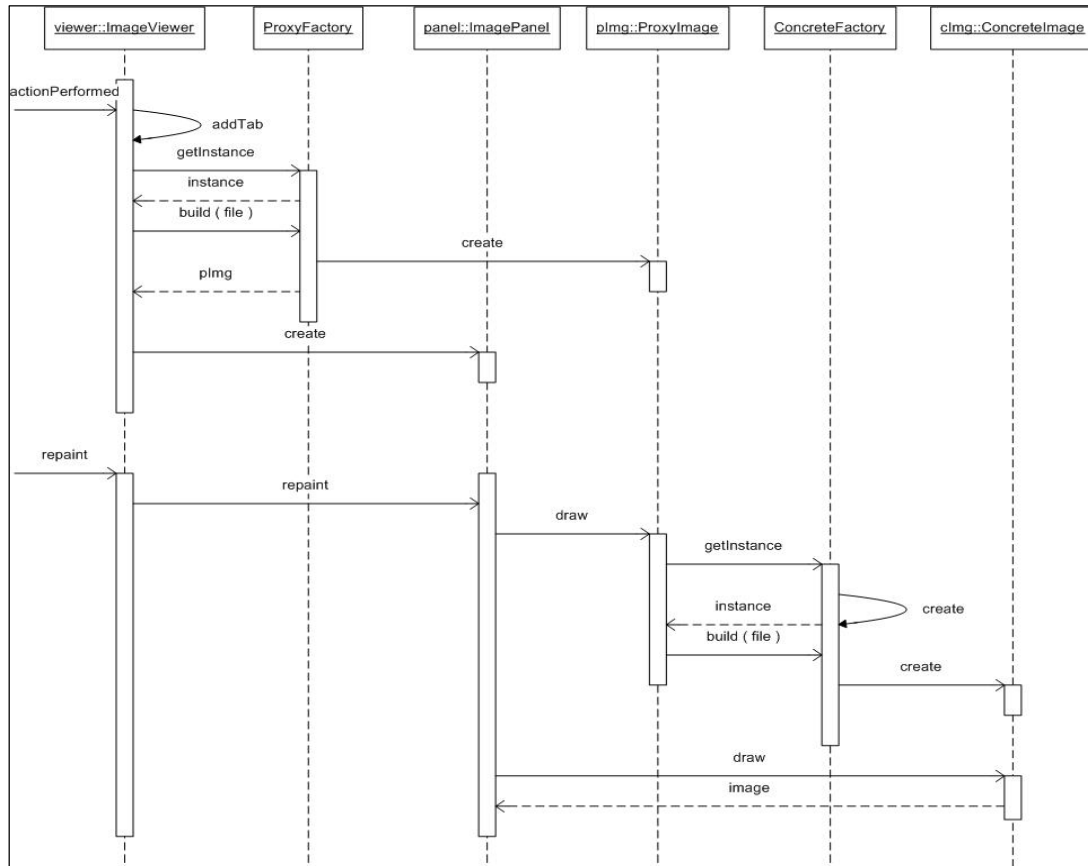


Figure 2 Diagramme de sequences

À faire :

Tâche 1 : Mettez à jour l'extension des fichiers acceptés dans la classe « ImageFileFilter ». La constante « EXTENSION » est utilisée pour ce faire.

Tâche 2 : Créez une classe concrète (donnez un nom significatif à votre classe) qui devra redéfinir la classe abstraite « Image ». Votre objet sera instancié par votre créateur d'image (voir la prochaine étape). Vous devrez définir les méthodes ainsi que les attributs dont vous aurez besoin pour afficher correctement les fichiers image BMP.

Tâche 3 : Implémentez votre version de l'interface « ImageFactory ». Votre classe devrait être un « Singleton ». Vous pouvez prendre exemple sur le code dans la classe « ProxyFactory » pour fabriquer votre « Singleton ». La méthode « build » sera la méthode qui instanciera un objet « Image » du type que vous avez défini lors de l'étape précédente. Cette méthode utilisera l'objet « File » passé en paramètre pour construire une nouvelle « Image ». C'est cette méthode qui ouvrira et effectuera la lecture d'un fichier « .bmp ».

Quantification et échantillonnage audio

Nous avons vu en classe que l'on peut contrôler la quantification et la fréquence d'échantillonnage de signaux numériques. Dans cette partie du TP, vous expérimenterez ces différents paramètres sur des fichiers audio et en constatarez l'effet sur la qualité. Vous utiliserez le logiciel Audacity afin de modifier les paramètres d'un fichier Wave (.wav).

À faire :

- Téléchargez la version d'Audacity depuis Moodle et lancez le logiciel Audacity (aussi disponible pour plusieurs plateformes dont Mac, Windows, et Linux : <http://audacityteam.org/>)
- Ouvrez le fichier « conversion.wav » (« File », « Open »)
- Convertissez « conversion.wav » dans les formats suivants :
 - Wave, 8 bits linéaire, stéréo, 44100 Hz
 - Wave, 16 bits linéaire, stéréo, 8000 Hz
 - Wave, 8 bits linéaire, stéréo, 8000 Hz
 - Wave, u-law, mono, 8000 Hz

Vous devez changer, si nécessaire, la fréquence d'échantillonnage. Pour ce faire, vous devez modifier la valeur affichée dans « *Project Rate (Hz)* » pour la valeur désirée.

Pour le dernier cas, vous devez aussi convertir la séquence stéréo en séquence mono. Pour ce faire, vous devez sélectionner dans le menu *Tracks* → *Stereo Track to Mono*.

Enfin, dans certains cas, vous devez modifier un paramètre du fichier Wave. Pour ce faire, vous devez exporter votre séquence audio en sélectionnant le menu *File* → *Export...* Puis, dans « *Type* », vous devez sélectionner « *Other uncompressed files* » et cliquer sur « *options* ». Finalement, vous devez sélectionner « *WAV (Microsoft)* » dans le champ « *Header* » et sélectionner l'encodage désiré (« *Unsigned 8 bit PCM* », « *Signed 16 bit PCM* » ou « *U-law* », c'est selon la conversion demandée) dans le champ « *Encoding* ».

Nommez les fichiers dans l'ordre fichier1.wav, fichier2.wav, etc.

Questions

- a. Notez votre évaluation de la qualité de chacun des fichiers (original et chacun des fichiers convertis) sur une échelle de 0 à 10 où 0 est la pire qualité et 10, la meilleure, avec le logiciel Audacity.
- b. Quel est l'effet d'une réduction de bits par échantillon sur la qualité audio ? Expliquez.

- c. Quel est l'effet d'une réduction de la fréquence d'échantillonnage sur la qualité audio? Expliquez.
- d. Notez et justifiez la taille de chacun des fichiers convertis en fonction de celle du fichier de départ. Vous devez justifier la taille des fichiers à l'aide de l'équation suivante :

$$Taille = NbCanaux \times \frac{NbOctets}{Echantillon} \times \frac{NbEchantillons}{Seconde} \times NbSecondes$$

- e. Quel fichier correspond à la qualité obtenue à travers le service téléphonique et pourquoi? Quel est le débit en bits par seconde?

À remettre

- ✓ La réponse aux questions de la section sur ImageMagick.
- ✓ Le projet ImageViewer avec le code ajouté. Le tout devra être remis dans un fichier ZIP avec nom NOM1_PRÉNOM1_et_NOM2_PRÉNOM2.zip.
- ✓ NE PAS REMETTRE D'IMAGES!
- ✓ La réponse aux questions de la section sur Audacity.

Critères d'évaluation :

1. Section sur ImageMagick : 40%
2. Section sur ImageViewer : 35%
3. Section sur Audacity : 25%

Date de remise

Au plus tard le jeudi 4 février 2016 à 13h30. Vous devez effectuer une remise électronique sur Moodle de votre travail (réponses aux questions et projet ImageViewer).

Pénalité de retard

Les dates de remises des travaux doivent être respectées. Une pénalité de 10 % par jour ouvrable sera appliquée aux travaux qui ne sont pas remis à temps.

Plagiat et fraude

Les clauses du « Chapitre 10 : Plagiat et fraude » du « Règlement des études de 1^{er} cycle » s'appliquent dans ce cours ainsi que dans tous les cours du département de génie logiciel et des TI.