

1 Introduction à Qt

1.1 Introduction

Au cours du premier TP, vous avez pu prendre en main quelques éléments d'OpenCV et constater que son API en terme d'interface graphique est limitée. Pour fournir une interface plus ergonomique, vous allez utiliser Qt qui offre beaucoup plus de possibilité en terme d'interface graphique.

1.2 Configuration du projet

Dans *qtcreator*, créer un nouveau projet en choisissant l'option *Application Qt avec widgets*. Lors de la phase *Informations sur la classe*, supprimez l'option *Générer l'interface graphique*. Vous nommerez votre projet : `projet_technologique`

Afin de s'assurer que tout le monde travaille avec la même version de *Qt*, veillez à vous assurer que vous utilisez bien la version 5.7.1 de *Qt*. Pour cela, utilisez le chemin d'accès suivants : `Outils > Options > Compiler & Exécuter > Kits` et vérifiez le champ : `Version de Qt` dans le profil `Desktop`. Vous pouvez ensuite exécuter votre programme en mode `Debug` à l'aide de la touche `F5`.

Vous devez aussi pouvoir builder votre programme dans un terminal en utilisant la commande suivante :

```
mkdir build && cd build && qmake -qt=qt5 .. && make
```

1.3 Charger un chemin d'accès vers un fichier

Commencez par ajouter un menu 'Fichier' qui contiendra une action 'Ouvrir'. Cette action appellera un *slot* `openFile`. Dans ce slot, vous demanderez à l'utilisateur d'indiquer un fichier de son choix à l'aide de la méthode `getOpenFileName` de la classe `QFileDialog`. Vous devrez vous assurer de filtrer les fichiers pour que l'utilisateur soit forcé de choisir des images.

Vous validerez le résultat obtenu en lisant et en affichant l'image à l'aide de la bibliothèque `OpenCV`, comme vu dans le TP1.

Les éléments suivants seront probablement pertinent pour arriver à un résultat satisfaisant : `QMenu` et `addAction`.

1.4 Afficher l'image dans Qt

Pour le futur, ce ne sera pas pratique de naviguer entre la fenêtre `Qt` et celle d'`OpenCV`, il vous faut donc maintenant réussir à afficher une `cv::Mat` dans votre fenêtre `Qt`.

Insérer correctement du contenu dans une fenêtre `Qt` n'est pas trivial, pour bien comprendre ce qui est nécessaire, vous pouvez commencer par parcourir les documentations des classes suivantes :

- `QMainWindow`
- `QWidget`
- `QLayout`
- `QLabel`

Pour obtenir une structure satisfaisante, il vous faudra suivre les étapes suivantes :

- Définir un `QWidget` comme widget central de votre `MainWindow`
- Définir un layout pour le widget central
- Ajouter un `QLabel` au layout

Pour essayer votre affichage, vous pouvez définir un `text` contenu dans votre `QLabel`. Ensuite, il vous faudra afficher l'image en se basant sur les éléments suivants :

- Écrire une fonction permettant de convertir une `cv::Mat` en `QPixmap`.
- Utiliser la méthode `setPixmap` de votre `QLabel`
- Redimensionner la taille de votre `QLabel`

Vous devriez alors voir apparaître l'image que vous avez choisie dans votre fenêtre.

1.5 Gestion de la taille des images et de la fenêtre

En tant qu'utilisateur, vous souhaitez pouvoir contrôler la taille de la fenêtre tout en affichant l'ensemble de l'image.

Pour améliorer la gestion des redimensionnements de votre fenêtre et supporter des tailles d'image variable, suivez le processus suivant :

- Créer une nouvelle classe `ImageLabel` qui contiendra vos images. Elle aura pour rôle de stocker une image et de mettre à jour l'affichage en la redimensionnant de manière à ce qu'un maximum du label soit occupé, sans que l'image ne soit déformée.
- Pour s'adapter à la taille de fenêtre en temps réel, vous devrez redéfinir la méthode `resizeEvent` et vous assurez que le résultat obtenu est fluide.

1.6 Gestion des filtres

Ajouter à votre fenêtre les éléments suivants :

- Une `QComboBox` permettant de choisir entre les méthodes de filtrage suivantes : `cv::Blur`, `cv::Laplacian`, `cv::Sobel`, il devra aussi être possible de choisir de n'appliquer aucun filtre.
- Un slider permettant d'ajuster la taille du noyau utilisé (on ne considérera que le cas où le noyau est carré pour le `cv::Blur`)

Afin de mieux comprendre comment faire communiquer les différents composants, lisez la documentation sur les *signals* et les *slots*¹.

Une fois votre slider fonctionnel, modifier votre code pour vous assurez qu'il disparaît lorsque vous n'appliquez aucun filtre.

1.7 Optionel : Pour aller plus loin...

Vous avez sûrement remarqué que le `QSlider` utilisé initialement ne contient qu'une simple barre, sans aucune information associée. Afin d'améliorer l'interface, vous pouvez implémenter des sliders spécialisés.

- Un slider un peu amélioré par rapport à `OpenCV` avec un nom, l'affichage de la valeur actuelle, mais aussi un affichage de l'intervalle autorisé. Vous pourrez par exemple l'obtenir en créant une classe héritant de `QWidget` qui contient deux `QLabel`, un pour le nom et un pour la valeur et l'intervalle.
- Un slider ne permettant de choisir que des nombres impairs, pratique pour `Sobel`, `Laplacian`.
- Un widget permettant de choisir une taille (largeur et hauteur). Il contiendra deux `QSlider`, mais un seul nom. Il pourra être utilisé pour choisir un noyau rectangulaire lors du flou.

2 Consignes de Rendu

L'évaluation prendra en compte le respect des consignes de rendu, particulièrement les points suivants :

1. <https://doc.qt.io/qt-5/signalsandslots.html>

- La deadline doit être rigoureusement respectée.
- Le travail doit être rendu à `antoine.billy@labri.fr` ET `lhofer@labri.fr`
- Le titre du mail doit être préfixé par `[ProjetsTechnologiques]`
- La pièce jointe doit être nommée `nom_prenom.tar.gz` et avoir les propriétés suivantes :
 - L'extraction avec `tar -xzf nom_prenom.tar.gz` doit s'exécuter sans erreurs
 - L'archive doit contenir un dossier nommé `nom_prenom` dans lequel sont présents tous les fichiers nécessaires à la compilation, mais aucun fichier produit lors de la compilation, ni les images utilisées pour tester votre programme.
 - Dans le dossier `nom_prenom`, la commande :

```
mkdir build && cd build && qmake -qt=qt5 .. && make
```

 doit s'effectuer avec succès et créer un exécutable `projet_technologique` qui est votre programme.