Travaux dirigés C++ n°5

Informatique

—IMAC 2e année—

Héritage et polymorphisme

Le but de ce TD est de comprendre l'intérêt et de mettre en place les mécanismes d'héritage et du polymorphisme dans l'application de traitement d'image précédemment développée.

Questions de révisions :

- Quelle est la syntaxe qui permet de mettre en place l'héritage entre deux classes?
- Quelle est la signification du mot clé "protected"?
- Comment est appelé le constructeur de la classe mère?
- Quelle est la signification du mot clé "virtual"?
- Qu'est-ce qu'une méthode "virtuelle pure"?
- Qu'est ce qu'une classe "Abstraite"?

Reprenez le code développé au cours du TD précédent ou téléchargez la correction sur la plateforme e-learning (https://elearning.u-pem.fr/).

▸ Exercice 1. *Class Filter*

1. *Créer une super-classe* ***Filter*** *qui sert comme classe mère de la hiérarchie des filtres : un ensemble de classes permettant des filtrages sur une image couleur (Exemples de traitements de Filtrage : seuillage, suppression d'une composante couleur ...).*

   La classe Filter n'a pas d'attribut, et comporte une seule méthode :

   - *applyFilter : prenant en paramètre une référence vers une ImageRGBU8. La définition de cette méthode sera donnée dans les classes filles.*

▸ **Exercice 2.** ***Class Threshold***

1. *Créer la classe* ***Threshold*** *qui hérite de la classe* ***Filter*** *et qui devra servir à appliquer un seuillage à une ImageRGBU8.*

   Les attributs :

   - *threshold : le seuil à appliquer à l'image (unsigned char).*

   Les méthodes :

   - *applyFilter : prenant en paramètre une référence vers une ImageRGBU8. Cette fonction modifiera l'image de façon appropriée.*

2. *Tester l'utilisation de la classe* ***Threshold****. Exemple de code test dans la fonction main :*

```
// read image
ImageRGBU8 image1;
readPPM(argv[1], image1);

// apply filter and save
Threshold t(100);
t.applyFilter(image1);
writePPM("test1.ppm", image1);
```

▸ **Exercice 3.** ***Class Filters***

1. *Créer la classe* ***Filters*** *qui servira d'un conteneur à filtres. L'objectif de cette classe est d'illustrer le polymorphisme. Elle contient un tableau de Filter* dans lequel on pourra mettre des filtres. Elle permettra ainsi d'appliquer une série de traitements sur une image.*

   Les attributs :

   - *size : Le nombre maximum de filtres qu'elle pourra contenir (unsigned int).*
   - *nbFilters : Le nombre de filtres dans le conteneur (variable à incrémenter à chaque fois on ajoute un nouveau filtre) (unsigned int).*
   - **filters : Un tableau de Filter* dans lequel on met les différents filtres.*

   Les méthodes :

   - *addFilter : la méthode qui prendra en paramètre une référence vers un Filter.*
   - *applyFilters : la méthode qui prendra en paramètre une référence vers une image ImageRGBU8. La méthode applyFilters devra bien entendu appliquer sur l'image l'ensemble des filtres du conteneur. Elle utilise ainsi la méthode applyFilter sur l'ensemble des filtres dans le conteneur.*

2. *Tester votre série de filtres de seuillage. Exemple de code test dans la fonction main :*

```
// load image
ImageRGBU8 image1;
readPPM(argv[1], image1);

// apply filters
Filters filters(2);
Threshold t1(150);
Threshold t2(100);
filters.addFilter(t1);
filters.addFilter(t2);
filters.applyFilters(image1);

// save image
writePPM("test_ex3.ppm", image1);
```

▸ **Exercice 4.** ***Class RemoveChannel***

Appliquer une série de seuillages sur une image est un traitement qui atteint vite ses limites en terme d'utilité. Il serait plus intéressant de pouvoir appliquer d'autres traitements et de les combiner entre-eux au sein de la classe Filters.

1. *Créer la classe* ***RemoveChannel*** *qui hérite de la classe* ***Filter****.*

   Cette classe doit définir la méthode applyFilter. Cette méthode doit permettre de retirer une composante couleur (rouge, vert ou bleu) de l'image couleur.

2. *Ajoutez des objets de type RemoveChannel à votre tableau de filtres et testez.*

   Remarque : *Si vous n'avez pas modifié votre classe "Filter" vous ne devriez pas voir de changement, pourquoi et que devez vous la modifier et comment?*

▸ **Exercice 5.** ***Encore du travail?*** ⋆⋆⋆

Rajoutez deux filtres un peu plus évolués :

- *Mean : Filtre de moyennage. Donnez la possibilité de régler la taille du kernel.*
- *Sobel : Filtre de detection de contours.*