

Enseignant(s)

**RAYNAL Benjamin**

Email(s)

[ben.raynal@gmail.com](mailto:ben.raynal@gmail.com)

## Classe Image

### 1 Matières, formations et groupes

Matière liée au projet :

Formations : -

Nombre d'étudiant  
par groupe : **2 à 3**

Règles de constitution des groupes: **Libre**

Charge de travail  
estimée par étudiant : **5,00 h**

### 2 Sujet(s) du projet

Type de sujet : **Imposé**



Le but de ce projet est de créer une classe Image en C++, représentant une image matricielle couleur. On considérera que les valeurs stockées seront des entiers non signés sur un octet.

Cette classe ne devra pas utiliser de librairie extérieure et être codée entièrement par vous.

Cette classe devra avoir les attributs suivants, accessibles par getters/setters :

- width (largeur)
- height (hauteur)
- channels (nombre de canaux)
- model (modèle colorimétrique) : exemples "NONE", "RGB", "GRAY", etc...

Cette classe devra avoir les méthodes suivantes :

- un constructeur par défaut, mettant les trois dimensions à 0, et le modèle à "NONE"
- un constructeur avec les trois dimensions, le modèle, et une valeur de remplissage
- un constructeur avec les trois dimensions, le modèle, et un buffer à recopier
- tout ce qu'il faut pour compléter la règle des 3
- les getters et setters des attributs
  - deux fonctions at (const et non const), prenant 3 coordonnées, pour accéder à une valeur. Si les coordonnées sont en dehors de l'image, on lancera une exception

On aura également les opérations suivantes :

- l'accès aux valeurs (via l'opérateur () avec trois paramètres), en deux versions : const et non const
- l'addition avec une autre image (via + et +=) : on additionne les valeurs des pixels correspondants
- l'addition avec une valeur (via + et +=) : on additionne la valeur à chaque canal de chaque pixel
- l'addition avec un pixel (tableau de valeurs) (via + et +=) : on additionne les valeurs du pixel à chaque pixel
- la soustraction avec une autre image (via - et -=)
- la soustraction avec une valeur (via - et -=)
- la soustraction avec un pixel (via - et -=)
- la différence avec une autre image (via ^ et ^=)
- la différence avec une valeur (via ^ et ^=)
- la différence avec un pixel (via ^ et ^=)
- la multiplication par une valeur réelle (via \* et \*=)
- la division par une valeur réelle (via / et /=)
- le seuillage par une valeur (via <, <=, >, >=, ==, !=) : le résultat sera une image de même taille, avec un seul canal en mode "GRAY", avec les valeurs 0 ou 255
- l'inversement des valeurs (via l'opération unaire ~), le résultat est une image de même taille, où toutes les valeurs sont (255-valeur d'origine)
- l'affichage (via <<) au format "<width>x<height>x<channels> (<mode>)"

Dans le cas où les opérations impliquent deux images de taille différente, le résultat sera de la taille max des deux, sur les deux dimensions.

Dans le cas où les opérations impliquent deux images de mode ou de nombre de canaux différents, l'opération lancera une exception.

On fera en sorte de gérer les dépassements : si l'opération donne un résultat inférieur à 0, on mettra la valeur 0, et l'opération donne un résultat supérieur à 255, on mettra la valeur 255.

Vous ajouterez également deux fonctions pour pouvoir charger et sauvegarder vos images, en utilisant la librairie de votre choix.

Vous joindrez à votre projet un fichier main.cpp séparé, démontrant toutes les fonctionnalités de votre classe.

La notation se fera sur :

- la réalisation des fonctions et opérations demandées
- la qualité du code (règle des trois, public/private, utilisation des const et des ref)
- la qualité du main

### 3 Détails du projet

**Objectif du projet (à la fin du projet les étudiants sauront réaliser un...)**

mise en application de ce qui a été vu ce semestre

**Descriptif détaillé**

**Ouvrages de référence (livres, articles, revues, sites web...)**

**Outils informatiques à installer**

### 4 Livrables et étapes de suivi

1

Rendu final

rendu

**dimanche**  
**02/03/2025**  
**23h59**

### 5 Soutenance

Durée de présentation  
par groupe :

**10 min**

Audience : **A huis clos**

Type de présentation :

**Démonstration**

Précisions :