

# 0 - Comparateur d'images (TD3)

Ce jalon est proposé comme sujet de TD et servira de programme préparatoire.

Nous vous fournirons le code de correction, ce qui vous permettra d'avancer si vous n'avez pas pu le faire lors du TD.

Néanmoins, pour progresser il est essentiel de pratiquer, et il est donc conseillé d'essayer de le faire par vous-même.

## Objectif

La première étape de ce projet consiste à réaliser un comparateur d'image.

Votre lanceur de rayons va générer des images, pour pouvoir l'évaluer, il faudra pouvoir les comparer à des images de référence fournies.

## Spécification

### Format

Nous utiliserons un format d'image ouvert, non destructeur, le [PNG](#). La plupart des langages disposent de bibliothèques pour gérer des images dans ce format.

### Comparaison

La comparaison de deux images est aisée : il suffit de comparer les deux images pixel par pixel, et de compter les pixels différents.

**Si les deux images diffèrent de moins de 1000 pixels, alors elles seront considérées comme identiques.**

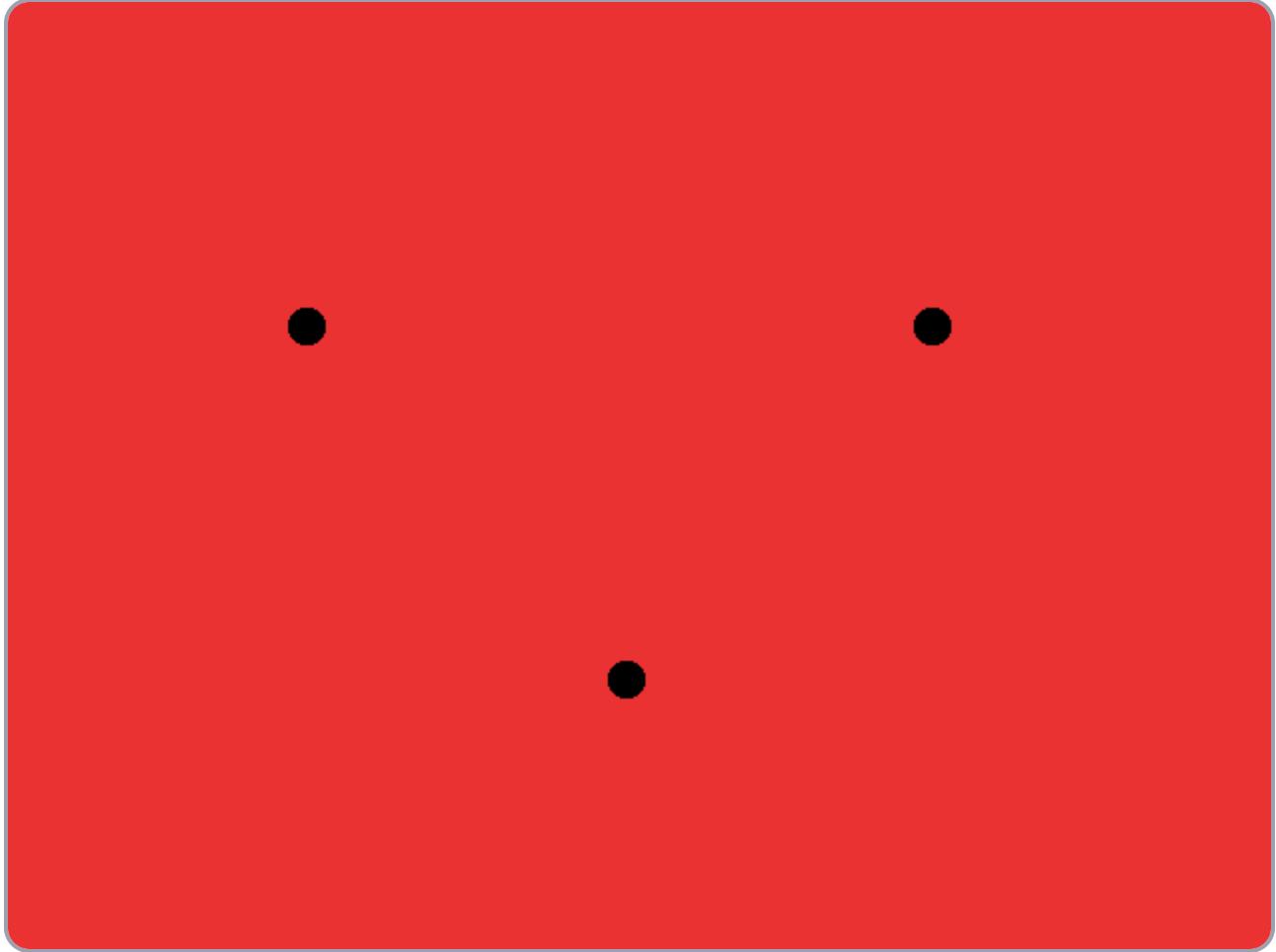
### Image différentielle

Pour rendre la comparaison des deux images plus simple, on générera une image qui représente la différence entre les deux images si elles diffèrent d'au moins un pixel.

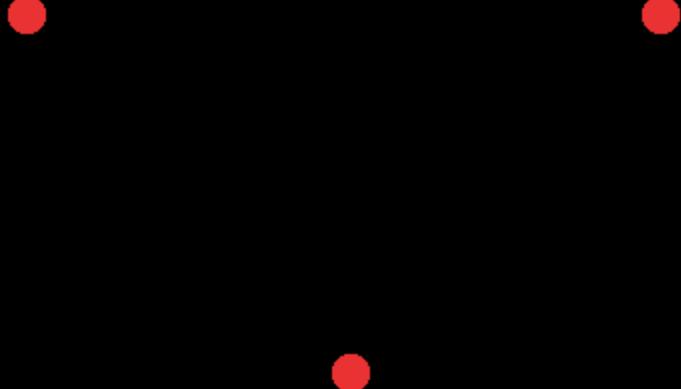
- Un pixel noir représentera un pixel identique dans les deux images
- La couleur des autres pixels sera la différence de couleur entre deux pixels

Par exemple, si l'on compare les deux images suivantes :

et



le comparateur doit permettre d'obtenir l'image suivante :



## Affichage en console

Le comparateur doit afficher deux lignes :

- La première doit contenir
  - OK si les deux images sont identiques
  - KO sinon.
- La deuxième ligne doit expliciter combien de pixels différents sont observés.

Voici deux exemples d'utilisation du programme :

```
$ java ImageCompare image1.png image2.png
OK
Les deux images diffèrent de 879 pixels.
```

```
$ java ImageCompare image1.png image2.png
KO
Les deux images diffèrent de 3056 pixels.
```

# Aide

## 1 - Mise en place du projet

- Créez un nouveau répertoire de travail et ouvrez-le dans l'éditeur de votre choix (VSCode ou IntelliJ par exemple)
- Si vous travaillez avec Git, initialisez votre dépôt avec `git init`
- Créez un projet avec Maven (ou Gradle au choix) intitulé `imgcompare`
- Le projet n'a besoin d'aucune dépendance externe à la librairie fournie en standard avec Java

## 2 - Organisation du programme et classes à créer

- Créez une classe `Main` dans `src/main/java/imgcompare`
- Créez une classe `ImageComparator`
- `Main` sera votre programme principal.
  - Vous ouvrirez et lirez les deux fichiers à comparer passés en paramètre (tableau `args`)
  - Vous les stockerez dans des `BufferedImage`
  - Vous pourrez appeler votre `ImageComparator` qui comparera les images et pourra vous renvoyer (deux méthodes différentes)
    - Le nombre de pixels différents
    - Une `BufferedImage` contenant l'image différentielle
  - Il ne restera plus qu'à
    - afficher le résultat comme spécifié
    - écrire votre image différentielle dans un fichier PNG
- `ImageComparator` se chargera de la comparaison des deux images. Il faudra créer des méthodes (au moins 2) :
  - pour compter les pixels différents
  - pour générer l'image différentielle

## 3 - Librairies et méthodes à utiliser

Pour travailler avec les images vous aurez besoin de

- `java.awt.image.BufferedImage` pour les images en mémoire
  - `getRGB(x, y)`
  - `getWidth()/getHeight()`
- `javax.imageio.ImageIO` pour lire/écrire l'image en PNG
  - `read(file)`
  - `write(bufferedImage, format, outputFile)`
- `java.nio` pour travailler avec les fichiers

## 4 - Informations utiles

N'oubliez pas :

- Gérer les exceptions (`IIOException`)
- Vérifier que les images ont la même taille avant de les comparer
- Pour travailler sur des images, on travaille pixel par pixel.
  - Il faudra donc faire
    - une boucle `for` (pour parcourir les lignes),
    - dans laquelle on fera une autre boucle `for` (pour parcourir chaque pixel de la ligne)