Compte rendu TP2 : Programmation générique

Commandeur N. - Verdant B

10/11/20

1 Introduction

Pour compiler les sources, il suffit de faire un make. Chaque réponse fera l'affaire d'une section avec la commande correspondante.

2 Définition d'une classe pour représenter des images arbitraires

Utilisation de la classe générique Image2D pour représenter une image en niveau de gris :

```
$ ./testGrayLevelImage2D
  5 5 5 5 5 5 5 5
  5 5 5 5 5 5 5 5
  5 5 5 5 5 5 5 5
  5 5 5 5 5 5 5 5
  5 5 5 5 5 5 5 5
  5 5 5 5 5 5 5 5
  5 5 5 5 5 5 5 5
  5 5 5 5 5 5 5 5
```

3 Introduction des images couleurs

Utilisation de la classe générique Image2D pour représenter une image en couleur.

```
$ ./testColorImage2DBash
255 0 255| 255 0 255| 255 0 255|
255 0 255| 255 0 255| 255 0 255|
255 0 255| 255 0 255| 255 0 255|
```

4 Premier itérateur sur images quelconques

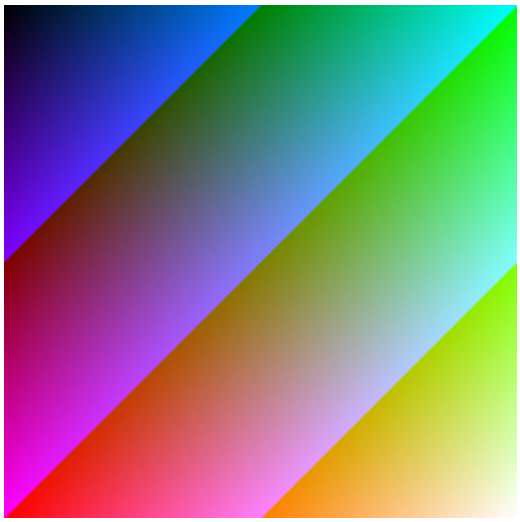
Utilisation de la classe générique Image2D pour représenter une image en couleur, pour l'instant pas d'affichage.

```
$ ./testColorImage2D
```

5 Un importeur / exporteur PBM générique

On va écrire une classe pour écrire et lire un fichier, en spécialisant pour la couleur et les nuances de gris. On va réutiliser la fonction permettant de créer l'image et l'écrire dans le fichier *colors.ppm*

```
$ ./testColorImage2D
$ display colors.ppm
```



6 Premier test: on inverse les canaux rouge et bleu

Grâce à nos fonctions de lecture et écriture, on peut commencer à traiter les images. On va lire une image et écrire une nouvelle en changeant la couleur de chaque pixel en inversant le bleu et le rouge. Il faut donner une image en entrée et une image en sortie.

```
$ ./invert-blue-red patinoire.ppm patinoire-inv.ppm
$ display patinoire-inv.ppm
```