

# **Rendu Projet Android**

## **Alice LACOUR**

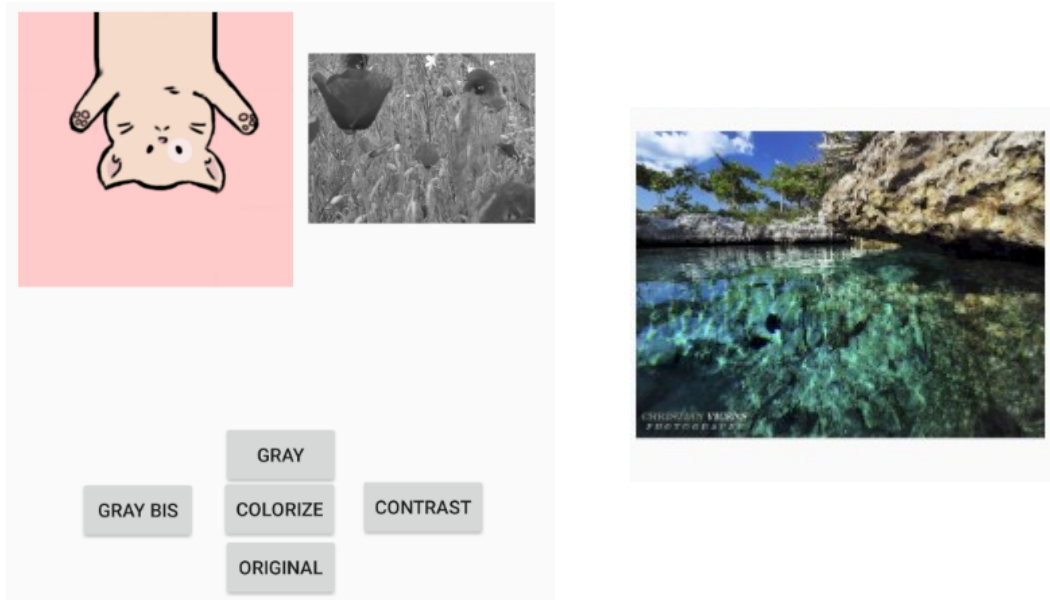
### **Phone : Pixel 2 / Oreo (Android 8.1x86)**

image "chat" 500 x 500 pixels

image fleur en nuance de gris : 800 x 600 pixels

image paysage : 1400 x 1050 pixels

### **Images avant transformation :**



### **Algorithmes implémentés :**

- toGray :  
Récupère tous les pixels de la photo passée en paramètre, modifie leur couleur en nuance de gris à l'aide de la fonction « pixelToGray » et place chaque pixel dans un tableau « pixels ». On applique ensuite la fonction « setPixels » sur ce tableau. On utilise cette méthode pour réduire le temps d'exécution de la fonction car « setPixels » est mieux optimisée que « setPixel ».

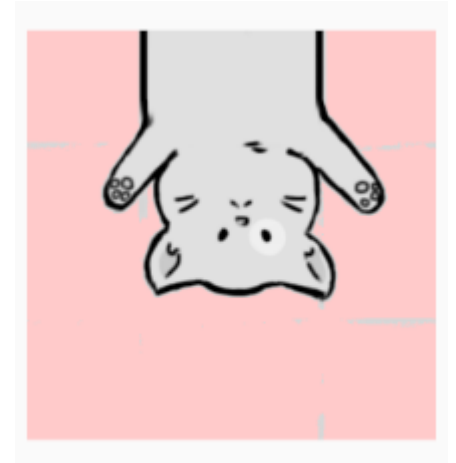
**Temps : Instantané**

- pixelToGray :  
Modifie l'entier représentant le pixel passé en paramètre pour le traduire en nuance de gris.



- **toGray\_bis :**  
Modifie partiellement l'image passée en paramètre pour mettre en nuance de gris une certaine partie de l'image. On stocke dans une variable `pixel_color` la valeur du pixel que l'on souhaite conserver. Ensuite on applique la nuance de gris aux pixels différents de `pixel_color`. (sur l'exemple on choisit de garder la couleur du pixel (0,0))

**Temps : Instantané**

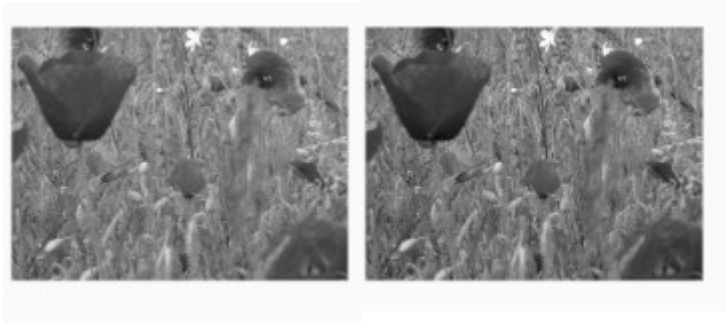


- **rgb\_to\_hsv :**  
Traduit les valeurs d'un pixel rgb en pixel hsv. Utilisée par la fonction `colorize`.
- **hsv\_to\_rgb :**  
Traduit les valeurs d'un pixel hsv en pixel rgb. Utilisée par la fonction `colorize`.
- **colorize :**  
Choisi une couleur de teinte de manière aléatoire. Place cette teinte dans tous les pixels (`hsv[0]`). On utilise toujours la méthode `setPixels`.
- **Temps : Instantané**



- **contrast :**  
Intensifie le contraste d'un image en nuance de gris. Parcours tous les pixels de la photo et créé un histogramme de contraste de celle-ci. On détermine la valeur du pixel le plus foncé, de même pour le plus clair, puis à l'aide du min et du max on rééchelonne l'histogramme à l'aide d'une formule on crée la « Look Up Table ». Enfin on réimplémente les nouveaux pixels avec leurs nouvelles valeurs de contraste .

**Temps : 3 secondes**



### **Bugs et limites :**

La fonction toGraybis pourrait être mieux optimiser, en effet lorsque l'on essaye avec une image plus grande et complexe (en terme de couleurs), celle-ci ne parvient pas à garder un fourchette colorimétrique de l'image intacte. (Elle met toute l'image en gris) Mon algorithme dois être trop précis (il ne garde que les pixels de la couleur exacte du pixel choisi au début)

N'ayant pas assisté au cours sur le RenderScript (raison médicale) je ne suis pas parvenue à rattraper mon retard et donc à comprendre entièrement comment l'utiliser. J'ai néanmoins essayé avec la fonction toGrayRS donnée dans la leçon mais l'applications ne run pas lorsque j'essaye de la lancer avec cet exemple. J'ai donc du rendre une application sans fichiers renderscript.