

## TP 3 – Analyse d’images

### Création de Plugins sous ImageJ

**Question 1**

ImageJ est un logiciel dont les fonctions peuvent être étendues par le biais de plugins. Les plugins sont des classes Java placées dans un dossier précis : le dossier plugins d’ImageJ (ou un de ses sous dossiers). Nous allons créer un premier script afin d’étudier la structure d’un plugin.

**Voir doc ImageJ fournie**

Créez le fichier MonScript\_.java en recopiant le code suivant :

```
import ij.*;
import ij.process.*;
import ij.plugin.filter.*;

public class MonScript_ implements PlugInFilter {
    public void run(ImageProcessor ip){
        byte [] pixels = ( byte []) ip.getPixels ();           // Notez le cast en byte ()
        int width = ip.getWidth();
        int height = ip.getHeight();
        int ndg;
        for (int y=0; y< height ; y++)
            for (int x=0; x< width ; x++) {                    // pas complètement optimal mais pédagogique...
                ndg = pixels [ y*width + x ] & 0xff;
                if (ndg<120)
                    pixels [ y*width + x ]= (byte)0 ;
                else
                    pixels [ y*width + x ]= (byte)255 ;
            }
    }

    public int setup(String arg, ImagePlus imp){
        if (arg.equals("about")){
            IJ.showMessage("Traitement de l'image");
            return DONE;
        }
        return DOES_8G;
    }
}
```

**Question 2**

Expliquez l’algorithme notamment :

Quelle est la taille du tableau pixels[] ? Donnez une formule générale permettant d’accéder à n’importe quel pixel d’une image via le tableau pixels : ndg[x][y]=pixels[ ??? ]

**Question 3**

Compilez et exécutez ce code (le fichier java doit être installé dans le répertoire plugin d’imageJ). Que fait-il ?

**Question 4**

Copiez et modifiez le code précédant afin qu’il permette de calculer la moyenne des niveaux de gris d’une image

**Question 5**

Complétez le code suivant afin qu'il permette de déterminer quelle image parmi celles d'un répertoire ressemble le plus à l'image ouverte avec imageJ. Le code permettant d'obtenir les fichiers présents dans un répertoire vous est donné.

```
public class FindSimilar_ implements PlugInFilter{

    public void run ( ImageProcessor ip ) {
        String path = "/home/TP/images/ ";
        File [] files = listFiles(path);
        if ( files.length != 0 ) {
            //initialization variables locales
            //TODO

            for (i =0;i < files.length ;i++)
            {
                // creation d'une image temporaire
                ImagePlus templmg = new ImagePlus ( files[i].getAbsolutePath() );
                new ImageConverter( templmg ).convertToGray8 ();
                ImageProcessor ipTemp = templmg.getProcessor() ;

                // Calcul du niveau de gris moyen de l'image et comparaison
                // TODO
            }
            IJ. showMessage ("L'image la plus proche est " + files[posImage].getAbsolutePath( ) + " avec une distance
de " + gap );
        }
    }

    public File [] listFiles ( String directoryPath ) {
        File [] files = null ;
        File directoryToScan = new File ( directoryPath );
        files = directoryToScan.listFiles();
        return files ;
    }

    // Retourne la moyenne des NdG d'une image en NdG
    public double AverageNdG ( ImageProcessor ip ) {
        //TODO
    }
}
```

### Question 6 / Image Tagging

A vous de jouer, durant les séances de TP restantes, en créant un plugins associant des mot-clés décrivant le contenu (couleur, contenu, qualité ...) des images et montrant vos compétences en traitement d'images. Ce plugins sera décrit de manière détaillé dans votre rapport.

Base d'apprentissage disponible sur Celene. Ce plugins devra générer des fichiers « .txt » au format similaire à ceux de la base d'apprentissage. Une image = un fichier .txt portant le même nom que l'image ; seul l'extension change ; le fichier txt est créé dans le même dossier que le fichier image.

Bon courage