

TP 3

Algorithmique et développement web (PHP)

IUT MMI S2

2 avril 2019

1 Objectif du TP

Ce TP concerne le traitement et l'analyse d'images. Vous avez à disposition le fichier classImage.php. Cette classe a été réalisée spécialement pour vous permettre d'expérimenter des algorithmes simples sur des images.

2 Description de la classe Image

Cette classe simplifie considérablement l'accès aux couleurs d'une image et donc permet de programmer en PHP des traitements simples.

2.1 Diagramme UML

Image
- \$width : largeur de l'image en pixel - \$height : hauteur de l'image en pixel - \$image : image - \$image_type : type de l'image (gif,jpg,png)
+ __construct() : prend 1 ou 2 paramètres + getWidth() : retourne la largeur de l'image en pixel + getHeight() : retourne la hauteur de l'image en pixel + getImage() : retourne l'image + changeCouleurImageXY(\$i,\$j,\$niveauR,\$niveauV, \$niveauB) : change la couleur du pixel i,j avec les niveaux RVB + getRougeImageXY(\$i,\$j) : retourne le niveau de rouge du pixel i,j + getVertImageXY(\$i,\$j) : retourne le niveau de vert du pixel i,j + getBleuImageXY(\$i,\$j) : retourne le niveau de bleu du pixel i,j + afficherImage()

2.2 Exemple d'utilisation

Exemple 1 :

```
<?php
include "classImage.php";

$monImage = new Image(100,100);

for($i=0;$i<$monImage->getWidth();$i++)
{
    for($j=0;$j<$monImage->getHeight();$j++)
    {

        $monImage->changeCouleurImageXY($i,$j,255,0,0);
    }
}

/* pour faire de l'affichage texte ne pas afficher l'image
echo "Rouge (10,10) = ".$monImage->getRougeImageXY(10,10)."</br>";
echo "Vert (10,10) = ".$monImage->getVertImageXY(10,10)."</br>";
echo "Bleu (10,10) = ".$monImage->getBleuImageXY(10,10)."</br>";
*/

$monImage->afficherImage(); //renvoie une image sous format png
imagedestroy($monImage->getImage());
?>
```

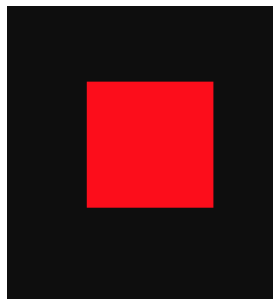


FIGURE 1 – Résultat

Exemple 2 :

```
<?php
include "classImage.php";

$monImage = new Image("test.jpg");

echo "Largeur de l'image en pixel = ". $monImage->getWidth(). "</br>"
echo "Hauteur de l'image en pixel = ". $monImage->getHeight(). "</br>"

$monImage->afficherImage();
imagedestroy($monImage->getImage());
?>
```

3 Travail à réaliser

- Fabriquez une image de 100x100 entièrement orange (pas rouge, pas jaune, pas vert – orange).
- Fabriquez une image de 200x200 qui contient des pixels de couleurs aléatoires (rand(0,255) retourne un nombre compris entre 0 et 255).
- Fabriquez une image avec des bandes horizontales qui couvre toutes les couleurs saturées du spectre (rouge, jaune, vert, bleu, magenta).
- Fabriquez un menu qui lit une image de votre choix en négatif (vous trouverez votre bonheur sur google image). Coder un algorithme qui permet d'afficher cette image avec les bonnes couleurs.
- L'image "Vichy_Labo.jpg" est à l'envers. Coder un algorithme qui permet d'afficher cette image à l'endroit.
- Vous disposez de 2 images : carte_droite et carte_gauche. Faites en sorte d'afficher les deux cartes sur la même image.
- Fabriquez un menu qui lit une image de votre choix en couleur (vous trouverez votre bonheur sur google image). Coder un algorithme qui permet d'afficher cette image en niveaux de gris.
- Colorisez cette image noir et blanc en rouge. En fait, on veut voir cette image comme à travers un filtre rouge.