NFA032 : TP n°9 (interface graphique du projet)

20 avril 2018

Le but du TP est de prendre en main les classes fournies pour le projet : l'interface graphique (classe `Afficheur`) et la classe de lecture d'une image au format PNG ou JPEG (classe `ImageReader`).

Composition d'une image

Une image est composée de points colorés appelés pixels. Caque pixel est caractérisé par ses composantes de couleur (rouge, vert, bleue) et son opacité. Si vous ne vous rappelez pas bien de cela, relisez l'énoncé du projet.

Description de l'interface graphique

L'interface graphique est un classe dont chaque instance représente une fenêtre ouverte à l'écran. Pour une exécution du programme demandé, il faudra utiliser une seule instance, un seul objet de type `Afficheur`.

L'interface graphique comporte un constructeur et deux méthodes.

Commençons par le constructeur. Celui-ci prend en paramètre un tableau de type `Pixel[][] pixels`. Il s'agit d'un tableau à deux dimensions de pixels. la première dimension, c'est la largeur de l'image ou si vous préférez, les colonnes du tableau. Le deuxième dimension est la hauteur de l'image ou si vous préférez les lignes. Chaque case du tableau contient une instance d'une classe qui implément l'interface `Pixel`. Cette interface à le code suivant.

```
public interface Pixel{
    int getRed();
    int getGreen();
    int getBlue();
    int getAlpha();
}
```

Il est possible via les 4 méthodes définies d'accéder aux 4 composantes d'un pixel, chacune étant un nombre entier compris entre 0 et 255.

Appeller le constructeur `Afficheur` avec un paramètre qui convient entraîne la création d'une fenêtre qui affiche l'image donnée en paramètre au constructeur.

La première méthode a pour nom `update`. Elle prend en paramètre un tableau de type `Pixel[][] pixels` qui représente une image. Pour que tout se passe bien, il faut que cette image ait la même dimension que celle donnée en paramètre au constructeur lors de l'instanciation de l'objet sur lequel la méthode est invoquée.

L'appel de cette méthode provoque l'affichage de l'image donnée en paramètre dans la fenêtre créée par le constructeur, à la place de celle qu'il y avait avant.

La seconde méthode s'appelle `fermer`. Elle a pour effet de fermer la fenêtre créée par le constructeur. C'est un moyen de fermer la fenêtre par programme. Il est également possible de fermer la fenêtre en cliquant sur la croix qui s'affiche ans le bandeau du haut de la fenêtre.

Mise en oeuvre de l'interface graphique

Question 1 : implémenter Pixel

Écrivez une classe qui implémente l'interface `Pixel` de façon minimale : elle doit comporter les quatre méthodes de l'interface (`getAlpha, getRed, getGreen` et `getBlue`) et un constructeur permettant de fixer les 4 composantes.

Question 2 : créer une image unie rouge et l'afficher

Créer une classe `TestAfficheur` avec une méthode `main`. Créez un tableau de type `Pixel[][]` de dimension 100x200 et mettez dedans des pixels représentant tous un point rouge opaque. Cela se fera en fixant pour chaque pixel les composantes alpha et rouge à 255, les composantes verte et bleue à 0. Vous pouvez créer un pixel différent par case du tableau ou mettre le même objet dans toutes les cases du tableau.

Créez un objet instance de `Afficheur` en utilisant le tableau de pixels rouges comme paramètre pour le constructeur de cette classe.

Admirez la belle image qui s'affiche à l'écran.

Vous pouvez fermer la fenêtre de l'image en cliquant sur la croix en haut à droite. Cela permet au programme de s'arrêter.

Question 3 : changer l'image affichée

Complétez la méthode `main` de `TestAfficheur` en créant un deuxième tableau de dimension 100x200 et contenant des pixels bleus. Appelez la méthode `update` de `Afficheur` sur l'objet créé avec l'image rouge pour afficher cette nouvelle image.

Question 4 : voir les deux images affichées

Votre programme a créé une fenêtre avac l'image rouge puis a remplacé l'image rouge par l'image bleue, mais cela s'est fait tellement vite que vous n'avez pas vu l'image rouge.

Pour créer un petit intervalle de temps entre la création de la fenêtre et l'appel de la méthode `update`, utilisez la méthode statique `Thread.sleep(x)` qui prend en paramètre un nombre de millisecondes d'attente. Cette méthode est susceptible de lever l'exception `InterruptedException`. Utilisez-là dans un `try...catch` qui traite cette exception (en ne faisant rien).

Pour vous inspirer, nous vous donnons un programme qui affiche un nombre aléatoire toutes les 2 secondes.

```
public class Inspiration{
    public static void main(String[] args){
        for (int i=0; i<10; i++){
            System.out.println(Math.random());
            try{
                Thread.sleep(2000);
            }catch(InterruptedException e){}
        }
    }
}
```

Question 5 : fermer la fenêtre

Pour terminer, à la fin de la méthode `main`, après une attente de deux secondes, fermez l'image en appelant la méthode `fermer` de la classe `Affichage`.

Lecture d'images

La classe `ImageReader` fournit une méthode permettant de lire une image au format PNG ou une image au format JPEG. Cette méthode appelée `readImage` prend en paramètre un chemin d'accès comportant le nom du fichier et éventuellement les dossiers à parcourir pour le trouver. Ce chemin peut être absolu ou relatif.

La méthode peut lever une exception `IOException` par exemple si le fichier n'existe pas ou le chemin est incorrect.

Le résultat de la méthode est un tableau à trois dimensions : la largeur, la hauteur et chaque pixel est représenté par la troisième dimension qui comprend 4 cases, une pour chaque composante d'un pixel.

Par exemple, si le résultat est mis dans une variable `tab` et que `tab[50][30]` est le tableau `{255,255,0,0}` le pixel colonne 50 ligne 30 a 255 d'alpha et de rouge et 0 de vert et de bleu.

Question 6 : lecture d'une image

Écrivez une classe `TestImageReader` avec une méthode `main` qui lit l'image du fichier `renard.png` fournie avec les classes décrites dans le TP et qui compte le nombre de points ayant l'opacité (composante alpha) à 255. Les images fournies sont dans un dossier appelé `images`. Au sein du projet eclipse, le chemin relatif à donner pour accéder au fichier `renard.png` est `"images/renard.png"`

Question 7 : affichage de l'image

Transformez le tableau renvoyé par la méthode `readImage` en un tableau de type `Pixel[][]` et affichez l'image au moyen de la classe `Afficheur`.

Question 8 : incrustation dans l'image

Créez une image en prenant comme fond le renard lu dans `renard.png` et en incrustant un carré rouge de 75x50 pixels au milieu du renard, de telle sorte que le premier pixel du carré soit à la position (200, 100) de l'image du renard.