

Imagerie Numérique : TD 4 (sur machine)

Opacité et transparence, manipulation d'images, Javascript : créer et effacer des balises
<http://www.labri.fr/perso/preuter/in2019>

Programme :

- Dans le TD1, nous avons fait un rappel HTML/CSS ainsi qu'une introduction sur le **DOM (Document Object Model) et sa manipulation en JavaScript**. Nous avons également fait une introduction à la **programmation événementielle**.
- Dans le TD2, nous avons vu comment **placer des images** relativement les une par rapport aux autres, ainsi que les **transformations 2D de similarité (translation, rotation, échelle)** et les coordonnées homogènes. Nous avons également vu deux **espaces de couleurs** parmi les espaces de couleurs existantes, à savoir RGB et HSL.
- Dans ce TD3, nous allons voir comment gérer **l'opacité/la transparence des images**. Nous allons également voir comment **créer et manipuler les images raster 2D**.
- Dans ce TD4, nous allons faire du **traitement d'images** sur des images existantes. Pour cela, nous allons voir comment **charger une image dans un canvas**, et d'en analyser son contenu. En **analyse d'images, il y a trois types de transformations** :
 1. les transformations géométriques : chaque pixel est transformé en fonction de sa position
 2. les transformations chromatiques : chaque pixel est transformé en fonction de sa couleur
 3. le filtrage (ou convolution) : chaque pixel est transformé en fonction du voisinage

1 Chargement d'images

Exercice 4.1 *Rappel HTML et préparation*

Dans cet exercice, vous allez établir un document HTML5 contenant deux images, une image source et une image résultat, et nous allons prévoir des opérations à faire entre pour transformer une image source dans une image résultat. Pour cela :

1. Dans votre dossier `in`, créer un dossier `td4`.
2. Avec Notepad++, créer un nouveau fichier `td4-traitement-images.html` dans ce dossier. Votre **arborescence de fichiers** sera donc :

```
in/
├── td1/
│   └── intro-javascript.html
├── td2/
│   └── td2-transformations2D.html
├── td3/
│   └── td3-manipulation-images.html
└── td4/
    └── td4-traitement-images.html
```

3. Dans ce fichier, mettez les balises pour un document HTML5 vierge simpliste et enregistrez-le. Dans l'en-tête du document (`head`), déterminez le titre (balise `title`) "IN - TD4".
4. Dans le corps du document (après `body`), créez un titre `h1` "Imagerie numérique", et ensuite un titre `h2` "TD4 - Traitement d'images".
5. Après le titre `h2`, ajoutez une balise paragraphe `p` avec le texte "L'objectif de ce TD est de :"

6. Après ce paragraphe, ajoutez une liste non-triée `ul`, avec deux éléments de liste avec le texte suivant. Le deuxième élément de liste contiendra lui-même une liste à trois éléments :

- Charger des images dans un canvas 2D
- Comprendre les 3 types de transformations en analyse d'images :
 - Transformation géométrique : chaque pixel est transformé en fonction de sa position
 - Transformation chromatique : chaque pixel est transformé en fonction de sa couleur
 - Filtrage (ou convolution) : chaque pixel est transformé en fonction du voisinage

7. Dans votre dossier `td4`, créez un sous-dossier `css` et dans ce dossier, un fichier `style.css`.

8. Dans ce fichier, ajoutez la règle CSS suivante :

```
h1, h2, h3
{
    color: rgb(0,0,0);
    background-color: rgb(221,221,221);
}
```

9. Ouvrez votre page web dans un navigateur.

Exercice 4.2 *Charger une image dans un canvas*

Dans cet exercice, nous allons voir comment charger une image dans un canvas, et nous allons préparer un second canvas destiné à accueillir une image résultat après transformation.

1. Après votre liste, ajoutez un titre `h3` "Chargement d'un image dans un canvas".
2. Insérez le code suivant permettant d'afficher deux images côte à côte.

```
<table>
  <tr>
    <td>Image source</td>
    <td>Image résultat</td>
  </tr>
  <tr>
    <td><canvas id="imageSource"></canvas></td>
    <td><canvas id="imageResultat"></canvas></td>
  </tr>
</table>
```

3. Ensuite, ajoutez le script JavaScript suivant dans votre fichier HTML. Notez que ce script contient deux nouveaux concepts, le **fonctionnement asynchrone**, et une **fonction anonyme**. Dans l'attribut `onload` de l'objet `imageSource`, nous affectons et définissons une fonction anonyme qui sera appelé uniquement après que l'image soit chargée. Ceci est nécessaire car nous ne pouvons qu'affecter la taille du canvas et son contenu après le chargement de l'image !

```
let imageSourceCanvas = document.getElementById("imageSource");
let imageSourceContexte = imageSourceCanvas.getContext("2d");

// Remplir le canvas source à partir d'un fichier
let imageSource = new Image();
imageSource.src = "neige.jpg";

// fonctionnement asynchrone: dès que l'image est chargé, on exécute une fonction anonyme
imageSource.onload = function() {
    imageSourceCanvas.width = imageSource.width ;
    imageSourceCanvas.height = imageSource.height ;
    imageSourceContexte.drawImage(imageSource,0,0);
};
```

4. Testez votre page web avec une image de votre choix, mais utilisez une définition d'une largeur de 600 pixels maximum. Par exemple, vous pouvez utiliser <https://www.labri.fr/perso/preuter/in2019/neige.jpg>.

2 Traitement d'images

Transformations géométriques

Exercice 4.3 Transformations géométriques : symétrie 1

1. A la suite de votre page HTML, ajoutez des titres et un bouton comme suit :

```
<h3>Transformations géométriques</h3>

<h4>1. Symétrie</h4>

<button id="boutonRetournerHorizontalement">Retourner horizontalement</button>
```

2. Ajoutez également le script JavaScript suivant. Comprenez-vous son fonctionnement ? Testez votre page dans votre navigateur.

```
<script>
let boutonRetournerHorizontalement = document.getElementById("boutonRetournerHorizontalement");
boutonRetournerHorizontalement.addEventListener("click", retournerHorizontalement);

function retournerHorizontalement()
{
    let imageSourceLargeur = imageSourceCanvas.width;
    let imageSourceHauteur= imageSourceCanvas.height;

    imageSourceDonnees = imageSourceContexte.getImageData(0, 0, imageSourceLargeur, imageSourceHauteur);

    // Dans cet exemple, les largeurs et hauteurs de l'image résultat sont équivalent à ceux de l'image source
    let imageResultatLargeur = imageSourceLargeur;
    let imageResultatHauteur= imageSourceHauteur;

    let imageResultatCanvas = document.getElementById("imageResultat");
    let imageResultatContexte = imageResultatCanvas.getContext("2d");

    imageResultatCanvas.width = imageResultatLargeur;
    imageResultatCanvas.height = imageResultatHauteur;

    imageResultatDonnees = imageResultatContexte.getImageData(0, 0, imageResultatLargeur, imageResultatHauteur);

    for (let y=0; y<imageResultatHauteur; y++)
    {
        for (let x=0; x<imageResultatLargeur; x++)
        {
            let pixelIndexSource = (x+y*imageSourceLargeur)*4;
            let r = imageSourceDonnees.data[pixelIndexSource];
            let g = imageSourceDonnees.data[pixelIndexSource+1];
            let b = imageSourceDonnees.data[pixelIndexSource+2];
            let alpha = imageSourceDonnees.data[pixelIndexSource+3];

            let pixelIndexResultat = ((imageResultatLargeur-x-1)+y*imageResultatLargeur)*4;
            imageResultatDonnees.data[pixelIndexResultat] = r;
            imageResultatDonnees.data[pixelIndexResultat+1] = g;
            imageResultatDonnees.data[pixelIndexResultat+2] = b;
            imageResultatDonnees.data[pixelIndexResultat+3] = alpha;
        }
    }
    imageResultatContexte.putImageData(imageResultatDonnees,0,0);
}
</script>
```

Exercice 4.4 Transformations géométriques : symétrie 2

Dans cet exercice, vous allez programmer le même concept que dans l'exercice précédent afin de retourner une image verticalement.

1. Créez un bouton "Retourner verticalement". Lorsque vous cliquez sur ce bouton, appeler une fonction `retournerVerticalement`. Définissez cette fonction ! Bien sûr, inspirez-vous largement de la fonction pour retourner horizontalement.

Exercice 4.5 Transformations géométriques : rotation

Dans cet exercice, vous allez établir des rotations sur des images.

1. Créez un bouton "Rotation 180 degrés". Lorsque vous cliquez sur ce bouton, appeler une fonction `rotation90degres`. Définissez cette fonction !
2. Créez un bouton "Rotation 90 degrés". Lorsque vous cliquez sur ce bouton, appeler une fonction `rotation90degres`. Définissez cette fonction !

Exercice 4.6 Transformations géométriques : homothétie

Dans cet exercice, nous allons faire un changement d'échelle de l'image, donc d'agrandir ou réduire la taille de l'image.

1. Créez un bouton "Agrandir". Lorsque vous cliquez sur ce bouton, appeler une fonction `agrandir`. Définissez cette fonction pour 2 paramètres `echelleX`, `echelleY` que vous initialisez avec la valeur 2 chacun, afin de doubler la hauteur et largeur de l'image.

Transformations chromatiques

Exercice 4.7 Transformations chromatique : changement d'espaces colorimétriques

Le passage d'un espace de couleur à un autre est réalisé, le plus souvent, par une transformation linéaire. Ici, on s'intéresse au passage de l'espace RGB vers l'espace YUV. La définition d'un pixel dans cet espace permet de séparer l'intensité d'un pixel (ou luminance identifiée par Y) par rapport à sa couleur (ou chrominance, U et V).

$$\text{RGB} \rightarrow \text{YUV} : \begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.147 & -0.289 & 0.436 \\ 0.615 & -0.515 & -0.100 \end{bmatrix} \cdot \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

$$\text{YUV} \rightarrow \text{RGB} : \begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1.140 \\ 1 & -0.395 & -0.581 \\ 1 & 2.032 & 0 \end{bmatrix} \cdot \begin{bmatrix} Y \\ U \\ V \end{bmatrix}$$

1. Ajouter un nouveau titre de niveau `h3` avec le texte "Transformations chromatiques".
2. Complétez l'affectation suivante qui calcule la composante Y à partir des valeurs RGB.

`Y =`

3. Créez un bouton "RGB vers niveau de gris". Lorsque vous cliquez sur ce bouton, appeler une fonction `rgb2gris`. Définissez cette fonction ! Notez que dans l'image résultat, la valeur de chaque composante couleur (hors l'opacité) correspond à la composante Y. Utilisez donc votre affectation définie ci-dessus.