ED MIHM – Traitement vidéo et processing

Exo1 : Ouvrir un fichier vidéo (par exemple `essai.mov`) et le jouer dans une fenêtre sous processing. On utilise pour cela la classe Movie dans le paquetage vidéo (`import processing.video.*`) qui contient les méthodes suivantes :

- un constructeur prenant en paramètre une référence vers la fenêtre de rendu (`this`), le nom du fichier à ouvrir et, optionnellement, le nombre d'images par seconde souhaité,
- une méthode `play` sans paramètre pour jouer la vidéo une seule fois ou une méthode `loop` sans paramètre pour la jouer continuellement,
- une méthode `read` qui permet de lire l'image suivante lorsqu'elle est disponible.

L'initialisation de la vidéo et son mode de lecture (une fois ou en boucle) sont des opérations faites dans la méthode `setup` de processing.
Le test de disponibilité d'une image peut être effectué de deux manières :

- soit par la méthode/événement `movieEvent(Movie m)`,
- soit dans `draw` par la méthode available de l'objet Movie qui renvoie true si une nouvelle image est disponible.

Dans les deux cas, il suffira alors de lire l'image suivante avec la méthode `read`.

Dans la méthode `draw` de rendu, il faudra également afficher l'image récupérée par un appel à la méthode `image` prenant en paramètres l'objet Movie puis les coordonées donnant la position du coin suppérieur gauche de l'image dans la fenêtre.
Puisque l'objet Movie est utilisé dans les méthodes `setup` et `draw`, il faudra le déclarer comme une référence globale.

Corrigé :

```
import processing.video.*;                // Utilisation du paquetage video
Movie myMovie;                            // Déclaration de l'objet Movie

void setup() {
  myMovie = new Movie(this, "essai.mov");      // Création de l'objet Movie
  myMovie.loop();                              // Lecture en boucle
}

void draw() {
  image(myMovie, 0, 0);       // Affichage de la trame courante à l'écran
}

// Méthode appelée automatiquement quand une nouvelle trame est disponible
void movieEvent(Movie m) {
  m.read();                   // Lecture de la trame suivante
}
```

Autre correction :

```
import processing.video.*;
Movie myMovie;

void setup() {
  myMovie = new Movie(this, "essai.mov");      // Création de l'objet Movie
  myMovie.loop();                              // Lecture en boucle
}

void draw() {
```

```
  if (myMovie.available()) {        // Si nouvelle trame dispo alors
    myMovie.read();                 // Lecture de la trame
    image(myMovie, 0, 0);           // Et affichage à l'écran
}
```

Exo2 : Modification du programme précédent.
On désire avant tout ajuster la taille de la fenêtre à la vidéo.
Pour cela, il faut lire la première image (`read`) lors de l'ouverture de la vidéo, récupérer sa taille (attributs `width` et `height` de l'objet `Movie`) et l'affecter à la taille de la fenêtre.

On veut maintenant effectuer des traitements de la vidéo avant affichage de chacune de ses images.

1. Modifier la teinte de l'image. Pour cela, on utilise la méthode `tint` dont les 3 paramètres sont les composantes RVB de la teinte, chacune entre 0 et 255. La spécification de la teinte se fait **avant** l'affichage de l'image. `noTint` permet de réinitialiser la teinte.
2. Inverser l'image. Pour cela, on a accès à une méthode `filter` qui implémente pour nous un traitement que l'on passe en paramètre (THRESHOLD, GRAY, INVERT, POSTERIZE, BLUR, OPAQUE, ERODE, or DILATE). Le filtre s'applique **après** l'affichage de l'image.

Corrigé :

```
import processing.video.*;                // Utilisation du paquetage video
Movie myMovie;                            // Déclaration de l'objet Movie

void setup() {
  myMovie = new Movie(this, "essai.mov"); // création objet Movie
  myMovie.read();                         // Lecture de la première trame
  size(myMovie.width, myMovie.height);    // taille écran = taille trame
  myMovie.loop();                         // Jouer en boucle
}

void draw() {
  tint(120, 32, 56);                      // Teinte pour l'image écran
  image(myMovie, 0, 0);                   // Copie image vidéo à l'écran
  filter(INVERT);                         // Filtre inversion couleurs
}

// Méthode appelée automatiquement quand une nouvelle trame est disponible
void movieEvent(Movie m) {
  m.read();
}
```

Exo3 : Charger deux vidéos et les jouer l'une superposée sur l'autre. La deuxième suit le curseur de la souris (`mouseX` et `mouseY` dans la méthode `draw`).
Pour l'affichage d'une nouvelle trame, lorsque les deux nouvelles images sont disponibles (`available`), on affiche la première et on fait un blend de la deuxième par-dessus.
La méthode blend prend les paramètres suivants : L'image que l'on blend dans la fenêtre de rendu, les 2 coordonnées X et Y du coin suppérieur gauche de l'image source considérée, sa largeur et sa longueur, les deux composantes X et Y où l'on effectue le blend dans la fenêtre, la largeur et la hauteur de la partie de l'image à blender.

Corrigé :

```
import processing.video.*;
Movie m1;               // 1er  objet Movie
Movie m2;               // 2eme objet Movie

void setup() {
  background(255);      // fond blanc
  m1 = new Movie(this, "essai.mov");      // Chargement movie1
  m2 = new Movie(this, "essai2.mov");     // Chargement movie2

  // La taille de l'écran est la taille de la 1ere vidéo
  // Même méthode que précédemment : on lit la première trame, on récupère
  // sa taille et on l'affecte à l'écran
  m1.read();
  size(m1.width, m1.height);

  // Lancement des vidéos
  m1.loop();
  m2.loop();
}

void draw() {
  // Si nouvelle trame dispo pour les 2 videos
  if (m1.available() && m2.available()) {
    // On les lit
    m1.read();
    m2.read();

    // On affiche la première à l'écran
    image(m1, 0, 0);
    // On blend (ajoute, mélange) la deuxième
    // La position de la 2eme video à l'écran est (mouseX, mouseY)
    blend(m2, 0, 0, m2.width, m2.height, mouseX, mouseY, m2.width,
m2.height, ADD);
  }
}
```