Processing
P06

1- Video

Cédric MASSART

Video

Capture webcam

```
import processing.video.*;

Capture video;

void setup() {
  size(320, 240);
  println(Capture.list());

  video = new Capture(this, 320, 240);
  video.start();
}

void captureEvent(Capture video) {
  video.read();
}

void draw() {
  image(video, 0, 0);
}
```

objet vidéo

création de la capture vidéo

lecture de l’image

dessine l’image

Capture webcam

```
import processing.video.*;

Capture video;

void setup() {
  size(320, 240);
  println(Capture.list());

  video = new Capture(this, 320, 240);
  video.start();
}

void captureEvent(Capture video) {
  video.read();
}

void draw() {
  image(video, 0, 0);
}
```

Liste des formats vidéo disponible :

```
name=Caméra FaceTime HD (intégrée),size=1280x720,fps=30
name=Caméra FaceTime HD (intégrée),size=1280x720,fps=15
name=Caméra FaceTime HD (intégrée),size=1280x720,fps=1
name=Caméra FaceTime HD (intégrée),size=640x360,fps=30
name=Caméra FaceTime HD (intégrée),size=640x360,fps=15
name=Caméra FaceTime HD (intégrée),size=640x360,fps=1
name=Caméra FaceTime HD (intégrée),size=320x180,fps=30
name=Caméra FaceTime HD (intégrée),size=320x180,fps=15
name=Caméra FaceTime HD (intégrée),size=320x180,fps=1
name=Caméra FaceTime HD (intégrée),size=160x90,fps=30
name=Caméra FaceTime HD (intégrée),size=160x90,fps=15
name=Caméra FaceTime HD (intégrée),size=160x90,fps=1
name=Caméra FaceTime HD (intégrée),size=80x45,fps=30
name=Caméra FaceTime HD (intégrée),size=80x45,fps=15
name=Caméra FaceTime HD (intégrée),size=80x45,fps=1
```

Manipulation de la vidéo

```
import processing.video.*;

Capture video;

void setup() {
  size(320, 240);
  video = new Capture(this, 320, 240);
  video.start();
}

void captureEvent(Capture video) {
  video.read();
}

void draw() {
  background(255);

  translate(width/2,height/2);
  imageMode(CENTER);
  rotate(PI/4);

  image(video, 0, 0, mouseX, mouseY);
}
```

centrage et rotation de 45° de la vidéo

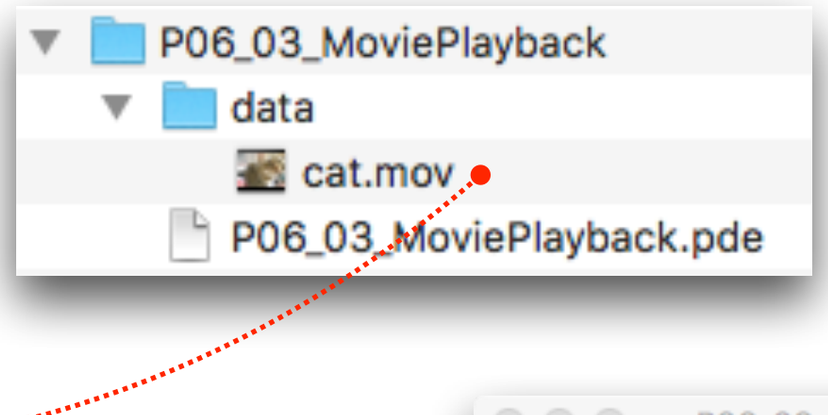
contrôle de la largeur et hauteur
avec la souris

P06_02_ManipulateCapture.pde

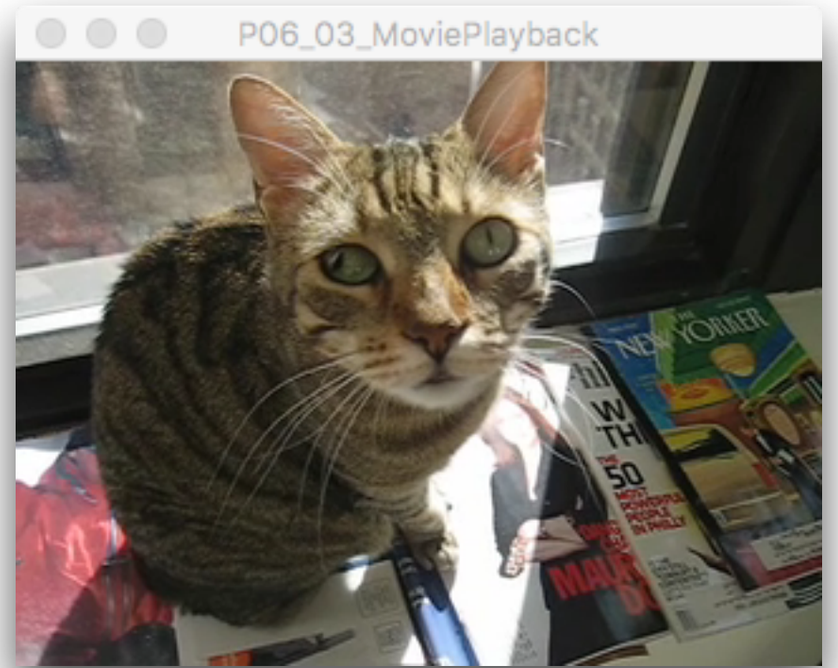
Lecture d’un fichier vidéo

chargement de la vidéo

```
import processing.video.*;

Movie movie;

void setup() {
  size(320, 240);

  movie = new Movie(this, "cat.mov");
  movie.loop();
}

void movieEvent(Movie movie) {
  movie.read();
}

void draw() {
  image(movie, 0, 0);
}
```



dessine l’image



P06_03_MoviePlayback.pde

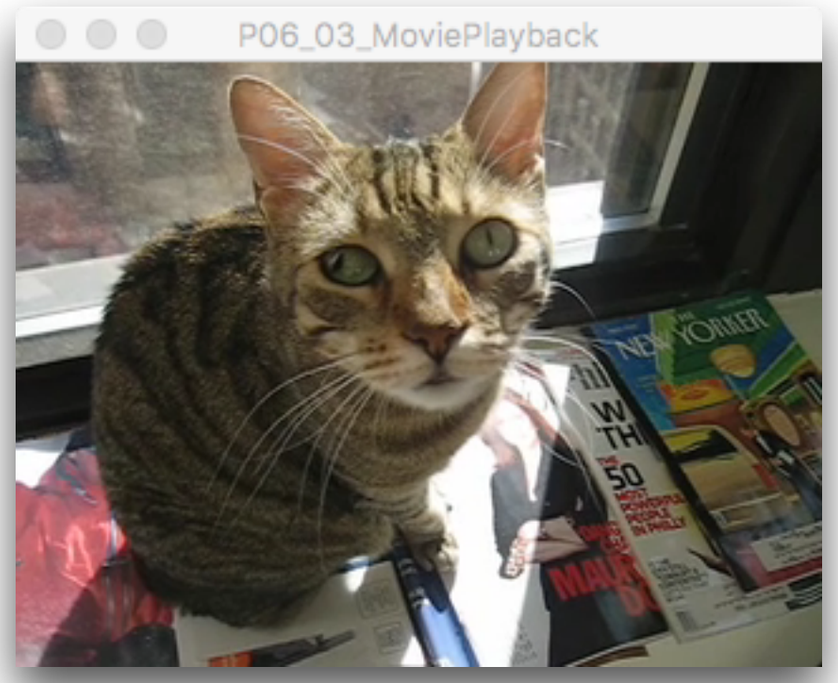
Position dans la vidéo

```
import processing.video.*;

Movie movie;

void setup() {
  size(320, 240);
  movie = new Movie(this, "cat.mov");
  movie.play();
}

void draw() {
  float ratio = mouseX / (float) width;
  movie.jump(ratio * movie.duration());

  movie.read();
  image(movie, 0, 0);
}
```

calcul de la position de la vidéo avec la position de la souris en x

lecture de l'image

dessine l'image



P06_04_MovieScrub.pde

Grille de base

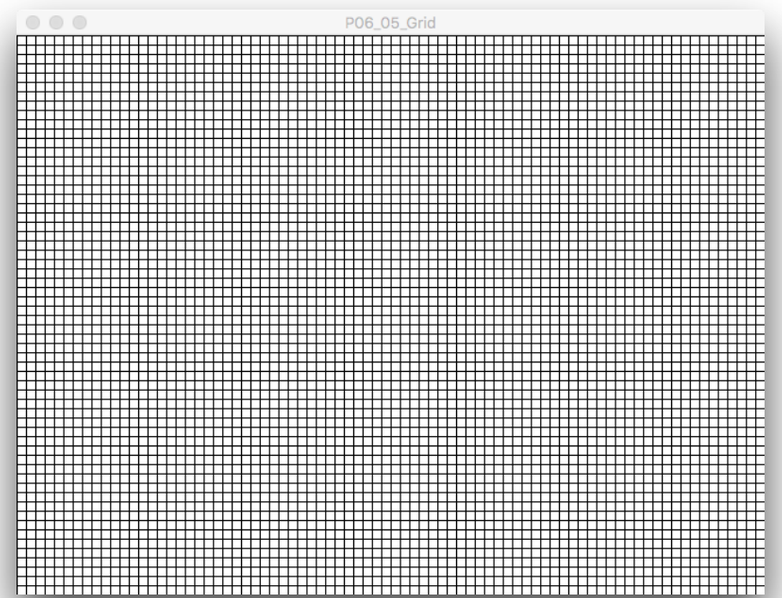
```
int videoScale = 8;

int cols, rows;

void setup() {
  size(640, 480);

  cols = width/videoScale;
  rows = height/videoScale;
}

void draw() {

  for (int i = 0; i < cols; i++) {
    for (int j = 0; j < rows; j++) {

      int x = i*videoScale;
      int y = j*videoScale;
      fill(255);
      stroke(0);

      rect(x, y, videoScale, videoScale);
    }
  }
}
```

taille d'un élément en pixel

nombre de colonne et de rang

double boucle
horizontale (x)
et verticale (y)

position



P06_05_Grid.pde

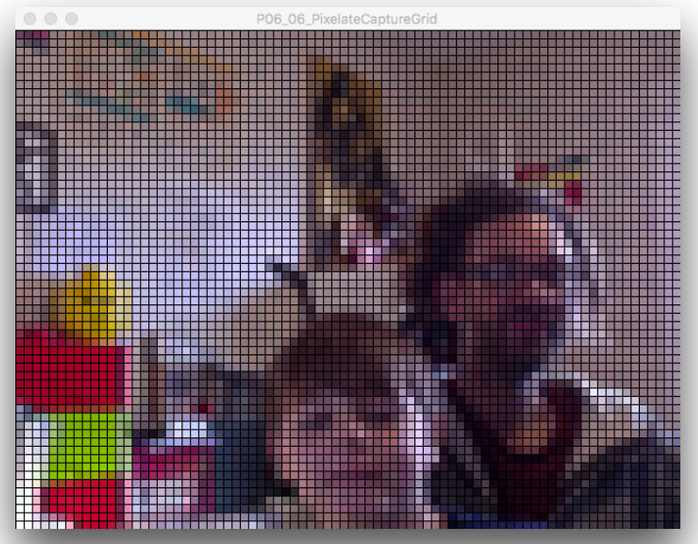
Vidéo dans une grille

```
import processing.video.*;

int videoScale = 8;
int cols, rows;
Capture video;

void setup() {
  size(640, 480);

  cols = width / videoScale;
  rows = height / videoScale;
  video = new Capture(this, 80, 60);
  video.start();
}

void captureEvent(Capture video) {
  video.read();
}

void draw() {
  video.loadPixels();

  for (int i = 0; i < cols; i++) {
    for (int j = 0; j < rows; j++) {

      int x = i * videoScale;
      int y = j * videoScale;
      color c = video.pixels[i + j * video.width];
      fill(c);
      stroke(0);
      rect(x, y, videoScale, videoScale);
    }
  }
}
```

objet vidéo

double boucle
horizontale (x)
et verticale (y)

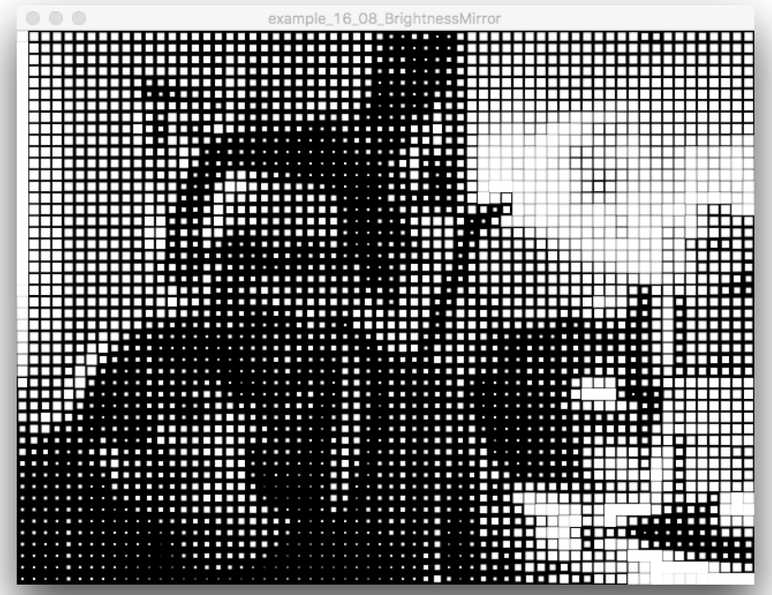
couleur du pixel
de la video



P06_06_PixelateCaptureGrid.pde

Vidéo dans une grille avec la brillance

```
import processing.video.*;

int videoScale = 10;
int cols, rows;
Capture video;

void setup() {
  size(640, 480);
  cols = width / videoScale;
  rows = height / videoScale;
  video = new Capture(this, cols, rows);
  video.start();
}

void captureEvent(Capture video) {
  video.read();
}


void draw() {
  background(0);
  video.loadPixels();

  for (int i = 0; i < cols; i++) {
    for (int j = 0; j < rows; j++) {

      int x = i * videoScale;
      int y = j * videoScale;

      int loc = (video.width - i - 1) + j*video.width;

      color c = video.pixels[loc];

      float sz = (brightness(c)/255)*videoScale;
      rectMode(CENTER);
      fill(255);
      noStroke();
      rect(x + videoScale/2, y + videoScale/2, sz, sz);
    }
  }
}
```



inversion des pixel X

taille du carré suivant la brillance

P06_07_BrightnessMirror.pde