

Travaux pratiques N°4

Communication Multimédia

Données : Programmes écrits avec Processing par (Indae Hwang, Jon McCormack), 2014, Monash University

Objectif : Transformer les sources afin d'obtenir des images (fixes ou animées) issues de votre création

Exercice 1 :

Transformer le code suivant en agissant sur la taille du rectangle, sur les couleurs (qui peuvent changer), sur le type de forme à visualiser en alternance, format d'image à sauvegarder (png, gif, bmp, jpg), etc
Générer l'image par un déplacement aléatoire ou guidé du curseur sans faire recours au périphérique souris.

```
// setup function -- called once when the program begins
void setup() {
    // set the display window to size 500 x 500 pixels
    size(500, 500);
    // set the background colour to white
    background(255);
    // set the rectangle mode to draw from the centre with a specified radius
    rectMode(RADIUS); // rectMode (RADIUS) uses the first two parameters of rect () as the shape's center
    // point, but uses the third and fourth parameters to specify half of the shapes's width and height.
}

void draw() {
    /* draw a rectangle at your mouse point while you are pressing
     * the left mouse button */
    if (mousePressed) {
        // draw a rectangle with a small random variation in size
        stroke(120); // set the fill colour to grey (200) with transparency 150 and stroke colour to a light grey for the
        outline
        fill(200, 150); // fill(gray, alpha), alpha is the opacity of the fillstroke(120); // set the fill colour to grey (200)
        with transparency 150 and stroke colour to a light grey for the outline
        fill(200, 150); // fill(gray, alpha), alpha is the opacity of the fill
        rect(mouseX, mouseY, random(10), random(10));
    }

    // save your drawing when you press keyboard 's'
    if (keyPressed == true && key=='s') {
        saveFrame("monimage.jpg");
    }

    // erase your drawing when you press keyboard 'r'
    if (keyPressed == true && key == 'r') {
        background(255);
    }
}
```

Exercice 2 :

Proposez votre propre version du programme suivant et produisez votre création (image) en intervenant sur tous les outils utilisés.

```
// variables for the angle (in radians) and increment
float angle;
float inc;
void setup() {
    size(500, 500);
    background(0);
    rectMode(CENTER); // rectangles drawn from the centre
    // initialise angle and inc to 0
    angle = 0; inc = 0;
}

void draw()
{
    /* draw a rectangle at your mouse point while you are pressing the left mouse button */
    // map the mouse x position to the range (0.01, 0.08)
    inc = map(mouseX, 0, width, 0.01, 0.08);
    // increment the current angle
    angle = angle + inc;
    if (mousePressed) { stroke(170); fill(10, 60); rect(mouseX, mouseY, 5, 5);
        line(mouseX, mouseY, pmouseX, pmouseY); // pmouse is the mouse position at the previous frame
        // oscillate the radius over time
        float radius = 100 * abs( sin(frameCount) );
        float first_tempX = mouseX + radius * cos( angle );
        float first_tempY = mouseY + radius * sin( angle );
        float second_tempX = mouseX + radius * cos(-angle);
        float second_tempY = mouseY + radius * sin(-angle);

        // draw some lines and circles using transparency
        stroke(110, 200, 0, 100);
        line(mouseX, mouseY, first_tempX, first_tempY);
        line(mouseX, mouseY, second_tempX, second_tempY);
        float temp_w = random(10);
        // ellipse(): Draws an ellipse (oval) to the screen. An ellipse with equal width and height is a circle. By
        // default, the first two parameters set the location, and the third and fourth parameters set the shape's
        // width and height. The origin may be changed with the ellipseMode() function.
        ellipse(first_tempX, first_tempY, temp_w, temp_w);
        ellipse(second_tempX, second_tempY, temp_w, temp_w);
    }
    // save your drawing when you press keyboard 's'
    if (keyPressed == true && key == 's') {
        saveFrame("yourName.jpg");
    }
    // erase your drawing when you press keyboard 'r'
    if (keyPressed == true && key == 'r') {
        background(0);
    }
}
```