

```
//code ECOVISUALIZACION-ECOANALOGIZACION: MODELO DE CIUDAD BIOCENTRICO  
BY FRANCASTILLO.NET
```

```
//code desarrollado en taller visualizar08: database city Medialab-  
prado Madrid  
//inspirad en code P.j.onori.  
//ecovisualizacion-ecoanalogizacion:ModelodeCiudad Biocentrico  
//Fran Castillo + Elena Moron.  
//martin.nadal+roberto.theron
```

```
import processing.video.*;  
PImage b;  
PFont font;  
PFont fontB;
```

```
//video
```

```
import processing.video.*;  
Movie myMovie;  
Movie myMovie2;  
Movie myMovie3;
```

```
MovieMaker mm;
```

```
float signature = (random(100));
```

```
int xSize = 1090;  
int ySize = 603;
```

```
float xAnchor = xSize/2+random(-xSize/4, xSize/4);  
float yAnchor = ySize/2+random(-ySize/4, ySize/4);
```

```
float separacion;  
float altura;  
float separacionBoton;
```

```
boolean portada=true;
```

```
Meanderer[] meanderers;
```

```
int total = int(random(40, 80));  
int delayInterval = int(random(3, 15));  
int pushCount = 0;
```

```
boolean capture = false;
```

```
boolean first=true;
```

```
DataSet ds=new DataSet();  
CityMap cm=new CityMap();  
ParallelCoordinates pc=new ParallelCoordinates();  
BarraBotonesContaminantes bc=new BarraBotonesContaminantes();  
BarraBotonesCountries bcount=new BarraBotonesCountries();
```

```
int [] countries;  
int [] sustancias;
```

```
color colores[];
```

```

/***** S E T U P
*****/
void setup(){

    b = loadImage("madridcity.jpg");

//video

    myMovie = new Movie(this, "02video.mov");
    myMovie2 = new Movie(this, "03video.mov");
    myMovie3 = new Movie(this, "04video.mov");
    myMovie.loop();
    myMovie2.loop();
    myMovie3.loop();

    ds.Parse("data/contaminantes.csv");

    separacion = 800 / ds.contaminantes.length; //calculamos la
separacion entre cada eje vertical
    altura =490-height;
    separacionBoton = 150 / ds.countries.length;

//colores
    colores = new color[ds.contaminantes.length];
    for(int i=0;i<colores.length;i++){
        colores[i]=color(random(255),random(255),random(255));
    }

    cursor(CROSS);
    size(xSize,ySize);

    font = loadFont("HelveticaCY-Bold-10.vlw");
    fontB =loadFont("Verdana-48.vlw");
    textFont(font, 8);

    frameRate(30);
    smooth();
    noFill();
    background(1);

    meanderers = new Meanderer[total];

    countries = new int[ds.countries.length];
    for (int i = 0; i < ds.countries.length;i++)
        countries[i] = 0;

    sustancias = new int[ds.contaminantes.length];
    for (int i = 0; i < ds.contaminantes.length;i++)
        sustancias[i] = 0;

}

/***** D R A W
*****/

```

```

void draw() {

fill(255);
noStroke();
rect(30,20,488,12);

fill(1);
text("ECOVISUALIZACION-ECOANALOGIZACION:MODELO DE CIUDAD BIOCENTRICO",
32, 30);

    if(portada){
fill(255);

text("...",80,50);
rect(0,0,800,449);
fill(1);
text("ECOVISUALIZACION-ECOANALOGIZACION:MODELO DE CIUDAD BIOCENTRICO",
32, 30);

        }else{
            cm.drawCityMap();
            //video
            image(myMovie, 800, 0);
            image(myMovie2, 800, 110);
            image(myMovie3, 800, 350);
        }
        pc.drawParallelCoordinates();
        bc.drawBarraBotonesContaminantes();
        bcount.drawBarraBotonesCountries();

        if(frameCount%delayInterval==1&&pushCount!=total-1||frameCount==0)
        {
            meanderers[pushCount] = new Meanderer(xAnchor, yAnchor);
            pushCount++;
        }

        println("pushcount"+pushCount);
        for(int i=0; i<pushCount; i++) meanderers[i].update();

        first=false;
    }

    /***** INTERACTION *****/
void mousePressed() {

    boolean update = true;
    float r = 0; //= random(255);
    float g = 0; //= random(255);
    float b = 0; //= random(255);

    color colorPulsado;
    int paisPulsado;

    if(mouseX <= 800)
    {

```

```

colorPulsado = bc.getColorBotonPulsado(mouseX,mouseY);
r = red(colorPulsado);
g = green(colorPulsado);
b = blue(colorPulsado);

if (update) {
  if (r != 0 && g != 0 && b!=0)
    for(int i=0; i<pushCount; i++) meanderers[i].setNewCol(r,g,b);
}
else
{
  portada=false;
  paisPulsado = bcount.getConuntryBotonPulsado(mouseX,mouseY);
  cm.actualizarMapa(paisPulsado);
  if (countries[paisPulsado] == 0)
    countries[paisPulsado] = 1;
  else
    countries[paisPulsado] = 0;
}
}

void keyPressed(){
  if(key == 'r')
  {
    image(b, 0,0);
  }
}

////////////////////////////////////// MEANDERER
//////////////////////////////////////
class Meanderer
{
  int count, seed, d1, toggle, baseAngle;
  float x, y, xSpeed, ySpeed, theta, angle, speed, d2, noiseScale,
noiseCount, noiseSpeed, xCount, yCount, angleMultiplier, noiseVal,
noiseCompoundX, noiseCompoundY;
  float colR, colG, colB;

  void setNewCol(float r, float g, float b) {
    colR = r;
    colG = g;
    colB = b;
  }

  Meanderer (float xPos, float yPos)
  {
    toggle = (random(-1,1)<0) ?-1:1;
    speed = random(.4, 2.5)*toggle;
    count=0;
    seed = int(random(200,1000));
    d1 = int(random(4,9));
    d2 = random(.2,.4);
    noiseScale=random(.01,.03);
    noiseCount=0;
    noiseSpeed=random(.001, .035);
  }
}

```

```

    xCount=int(random(-10, 10));
    yCount=int(random(-10, 10));
    angleMultiplier = random(.3,.9);
    baseAngle = int(random(2,4))*360;
    noiseCompoundX=random(.1,.3);
    noiseCompoundY=random(.1,.3);

    x = xPos;
    y = yPos;
}

void update()
{
    noiseDetail(d1, d2);
    noiseSeed(seed);
    noiseVal=noise((x-xCount)*noiseScale, (y-yCount)*noiseScale,
noiseCount);

    angle -= (angle - noiseVal*baseAngle)*angleMultiplier;
    theta = -(angle * PI)/180;
    xSpeed = cos(theta)*speed;
    ySpeed = sin(theta)*speed;

    x -= xSpeed;
    y -= ySpeed;

    float avgSpeed = (xSpeed+ySpeed)/2;
    noFill();
    stroke(colR,colG,colB,35.0/speed);
    ellipse(x,y,40*avgSpeed,40*avgSpeed);

    noiseCount+=noiseSpeed;
    xCount+=noiseCompoundX;
    yCount+=noiseCompoundY;
    count++;
}
}

//////////////////////////////////// C I T Y M A P
////////////////////////////////////
class CityMap
{
public void CityMap(){}
public void drawCityMap()
{
    b = loadImage("madridcity.jpg");

    if (first)
        image(b, 0, 0);

    textFont(font, 12);
}

public void actualizarMapa(int nuevo)
{
    switch(nuevo)
    {
        case 0: b =loadImage("roma_italia.jpg");
    }
}
}

```

```

        break;
    case 1: b =loadImage("londres_reinounido.jpg");
        break;
    case 2: b =loadImage("berlin_alemania.jpg");
        break;
    case 3: b =loadImage("madrid_espana.jpg");
        break;
    case 4: b =loadImage("dublin_irlanda.jpg");
        break;
    case 5: b =loadImage("paris_francia.jpg");
        break;
    case 6: b =loadImage("viena_austria.jpg");
        break;
    case 7: b =loadImage("helsinki_finlandia.jpg");
        break;
    case 8: b =loadImage("lisboa_portugal.jpg");
        break;
    case 9: b =loadImage("bruselas_belgica.jpg");
        break;

    case 10: b =loadImage("estocolmo_suecia.jpg");
        break;
    case 11: b =loadImage("amsterdam_holanda.jpg");
        break;
    case 12: b =loadImage("copenhague_dinamarca.jpg");
        break;
    case 13: b =loadImage("oslo_noruega.jpg");
        break;
    case 14: b =loadImage("budapest_hungria.jpg");
        break;
    // case 15: b =loadImage("londres_reinounido.jpg");
    //         break;
    default: exit();

}

image(b, 0,0);
}
}

//////////////////////////////// P A R A L L E L   C O O R D I N A
T E S //////////////////////////////////
class ParallelCoordinates
{

    int offset = 10;
    public void ParallelCoordinates(){}

    public void drawParallelCoordinates()
    {
        pushMatrix();
        translate(0,140+b.height);

        //Dibujar los ejes

        //dibujamos los ejes verticales
        fill(255);
        noStroke();
        rect(0,-140,800,200);
    }
}

```

```

stroke(#D1D1D1);
for (int i= 0; i < ds.contaminantes.length; i++) {
    line(i*separacion+offset,0,i*separacion+offset,altura-20);
}

//dibujamos la grafica
for(int j=0;j<ds.countries.length;j++){
    //stroke(colores[j]);

    if (countries[j]== 1)
    {
        strokeWeight(4);
        stroke(230,50,50);

    }
    else
    {
        strokeWeight(1);
        stroke(50,50,50);
    }

    for (int i= 0; i < ds.contaminantes.length-1; i++) {
        line(i*separacion+offset,-
1*ds.values[i][j],i*separacion+separacion+offset,-
1*ds.values[i+1][j]);
    }
}
popMatrix();
}
}

/***** B O T O N   C O N T A M I N A N T E
*****/
class BarraBotonesContaminantes
{
    int offset = 10;

    BarraBotonesContaminantes(){}

    public void drawBarraBotonesContaminantes()
    {
        pushMatrix();
        translate(0,140+b.height);

        noStroke();

        //dibujamos las zonas de color
        for (int i= 0; i < ds.contaminantes.length; i++) {
            fill(colores[i]);

            String[] p = splitTokens(ds.contaminantes[i], " "); // the data
is parsed correctly
            String shortName="";
            if (p[0].length() > 4)
                shortName=p[0].substring(4);
            else
                shortName = p[0];

```

```

        text(shortName,i*separacion + (offset/4), -130);

        rect(i*separacion,altura-10,separacion, 10);
    }

    textFont(fontB,12);
    fill(240);
    for (int i= 0; i < ds.countries.length; i++) {

        text(ds.countries[i],b.width+offset, -130+i*offset);

    }

    popMatrix();
}

public color getColorBotonPulsado(int x, int y)
{
    color colorPulsado;

    if (y > 465 && y < 475 && x < 800)
        colorPulsado = colores[(int)(x)/(int)separacion];
    else
        colorPulsado = color(0);

    return colorPulsado;
}

/***** UTILS *****/
boolean isMouseInside(float x, float y, float width, float height) {
    return (mouseX>=x && mouseX<=x+width && mouseY>=y &&
mouseY<=y+height);
}

/***** BOTON COUNTRIES *****/
class BarraBotonesCountries
{
    int offset = 10;

    BarraBotonesCountries(){}

    public void drawBarraBotonesCountries()
    {
        pushMatrix();
        translate(0,140+b.height);

        textFont(fontB,12);

        for (int i= 0; i < ds.countries.length; i++) {

            fill(255,255,255);

            text(ds.countries[i],b.width+offset, -130+i*offset);
            if(countries[i] == 1)
            {

```

```

        fill(255,50,50);
        text("+",b.width+offset-8, -130+i*offset);
    }
    else
    {
        fill(0,0,0);
        text("+",b.width+offset-8, -130+i*offset);
    }
    // fill(240,240,240,100);
    // rect(b.width+offset, -130+i*offset,offset,
offset*10*ds.countries[i].length());

    }

    popMatrix();
}

public int getConutryBotonPulsado(int x, int y)
{
    int countryPulsado;

    if (y > b.height && y < height && x > 800 && x < 880)
        countryPulsado = (int)(y-b.height)/(int)separacionBoton;
    else
        countryPulsado = -1;

        println("sep="+separacionBoton);

    return countryPulsado;
}

}

// Called every time a new frame is available to read
void movieEvent(Movie m) {
    m.read();
}

//clase dataset

class DataSet {
    float[][] values;
    String[] contaminantes;
    String[] countries;

    //Fills data structure with a csv data file
    void Parse(String File){
        String lines[]=loadStrings(File);
        //la primera linea es de los contaminantes
        contaminantes=split(lines[0],',');
        values = new float[contaminantes.length][lines.length-1];
        countries = new String[lines.length-1];
        for(int j=1;j<lines.length;j++){
            String tempstr[]=split(lines[j],',');
            countries[j-1]=tempstr[0];
            for(int i=1;i<tempstr.length;i++){
                values[i-1][j-1]=float(tempstr[i]);
            }
        }
    }
}

```

```
}  
}
```

**ECOVISUALIZACION-ECOANALOGIZACION** es una obra que está bajo Licencia Creative Commons.  
[lenguaje de programacion:Processing]