

PROGRESO 3-INTERACTIVIDAD

Nombre: Ivanova Llumipanta

Sigla: MTM 575-1

Matrícula: 702383

VARIABLES

Aquí declaro las variables para la boca, que me permiten mover ésta imagen que es la boca:

```
//bok
int nimgX, nimgY;
int nposx=250;
int nposX=350;

int nposy=470;
int nposY=520;
PImage nimg;

//Posicion offset
int nmX, nmY;
//para arrastrar la imagen
float mX, mY;
//para el limite o rango
float posX=590;
float posX=690;
float posy=600;
float posY=700;
float imgX;
float imgY;
//cierre boka

//cierre bok
```

Variables para el background, esta me permite especificar su color que es rojo.

```
//background
color bgcolor; //color de fondo
//Cierre background
```

Variables para la Imagen Pista



Presiona la barra
espaciadora para ver
la imagen que
debes armar



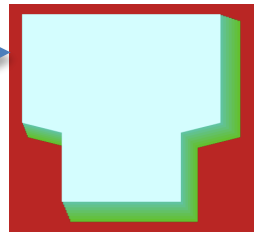
```
//imagen pista:
PImage photo;
int count=0; //inicializa un contador que permite que la imagen pista aparezca en la
pantalla con un if.
//Cierre imagen pista
```

Variables para la nariz



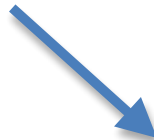
```
//nariz
int has;
int px; // Posición inicial de la nariz en la pantalla en el eje x.
int py; // Posición inicial de la nariz en la pantalla en el eje y.
int i; //Variable para trasladar la nariz
color c; //Define el color de la nariz, que es negro
```

Variables Base de la "CALAVERA"



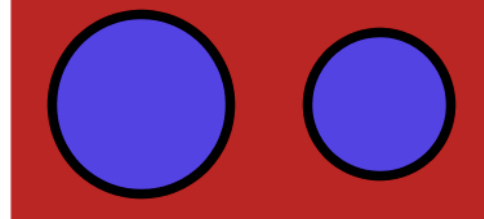
```
int q; //color degrade
```

Variables para los "OJOS"



```
//ojos
float bx;
float by;
int eyeSize = 120;
boolean overBox = false;
boolean locked = false;
float xOffset = 0.0;
float yOffset = 0.0;
```

Luego arrastra los círculos



VOID SETUP

```
void setup() {
  size(1000,700);
  photo = loadImage("imagen.png");
  //boka
  nimg = loadImage("boca2.png");
  centerImage();
```

```

//cierre BOk
//nariz:
px =900;//posición cuando inicia la pantalla en el eje x de toda la nariz
py =230;//posición cuando inicia la pantalla en el eje y de toda la nariz
c = 0;
//ojos:
bx = width/1.4;//hace que la imagen se ubique en esta posición dependiendo del
tamaño del background
by = height/1.7;
rectMode(RADIUS);
d = new Draggable(900,640,100,8);//dibuja el rect
noStroke();
//cierre ojos
}

```

“CLASS DRAGGABLE” es la función que permite que las figuras del rompecabezas se puedan arrastrar mediante el Mouse a través del background.

```

class Draggable {////CIERRE-ESTE EL LA FUNCION PRINCIPAL
  boolean dragging = false; //
  boolean rollover = false; //
  float x,y,w,h; //
  float offsetX, offsetY;

  Draggable(float tempX, float tempY, float tempW, float tempH) {
  }

  // visualizacion
  void visualizacion() {
    stroke(0);
    if (dragging) fill (50);
    else if (rollover) fill(120);
    else fill(0);
    strokeWeight(1);
    // rect(x,y,w,h); //rectantgulo 2
    noStroke();
  }
  void reset() {
    for(i=0;i<7;i++){

```

```

px=50;
py=150;
c=color( 204, 153, 0);
}
has = -1;
}

```

```

void clicked(int mx, int my) {
  if (mx > x && mx < x + w && my > y && my < y + h) {
    dragging = true;
    offsetX = x-mx;
    offsetY = y-my;
  }
}

```

```

void rollover(int mx, int my) {
  if (mx > x && mx < x + w && my > y && my < y + h) {
    rollover = true;
  } else {
    rollover = false;
  }
}

```

```

void stopDragging() {
  dragging = false;
}

```

```

void drag(int mx, int my) {
  if (dragging) {
    x = mx + offsetX;
    y = my + offsetY;
  }
}

```

```

}
//Cierre-Dibuja ojoizquierdo
//CIERRE-ESTE EL LA FUNCION PRINCIPAL
}

```

Draggable d;



```

void draw() {
  bgcolor = #CC0000;
  background(bgcolor);
}

```

Este if permite que al presionar la barra espaciadora identificada como " ' ' ", se inicie un contador que hace que la **imagen pista** aparezca en la pantalla.

```
if(keyCode==' '){  
  count++;  
}else{  
  if ( count >10){  
    image(photo, 870, 20);  
  }  
}
```

Presiona la barra
espaciadora para ver
la imagen que
debes armar



Elementos que van en la pantalla.

1.Líneas:

```
strokeWeight(4);  
stroke(255); //color de líneas  
fill(255);  
line(620, 0, 620, 700);  
line(620, 150, 1000, 150);  
line(620, 320, 1000, 320);  
fill(#E89B0C);  
line(620, 520, 1000, 520);
```

2.Texto:

```
noStroke();  
textSize(20);  
text("Finalmente arrastra el rectángulo", 630, 550); //  
text("mueve el cursor por el", 630, 570); //  
text("área donde iría", 630, 590); //  
text("la boca, por las 3 líneas", 630, 615); //  
text(" verticales y podrás arrastrarlo", 630, 630); //  
text("Luego arrastra los círculos", 630, 340);  
text("Primero arrastra el triángulo", 630, 180);  
text("Presiona la barra", 630, 30);  
text("espaciadora para ver", 630, 50);  
text("la imagen que", 630, 70);  
text("debes armar", 630, 90);  
noFill();
```

3.DEGRADADO:

El for hace que la variable “**q**”, que es la que da los colores al degradado. Se inicie en 12 y que hasta llegar a ser menor que 20, se incremente en 1. (**q=q+1**)

```
noStroke();  
for(int q=12;q<20;q++){  
  scale(.99); //va escalando la forma de la calavera
```

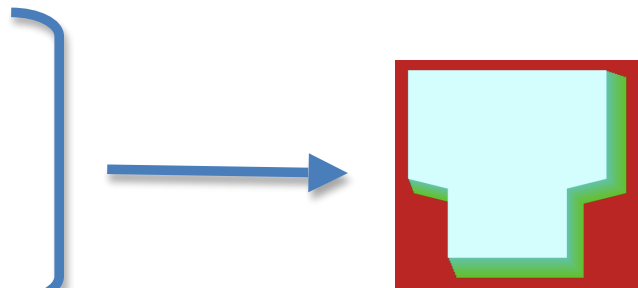
“(q>=10)?(20*(q-10)):0)” es un shortcut de un **if else**. Y se debe recordar que el fill es fill (v1,v2,v3), en rgb

“V1=67-10*q” es el valor en rojo.

“v2= 200” es el valor en verde.

“v3= (q>=10)?(20*(q-10)):0

```
fill(67-10*q,200,(q>=10)?(20*(q-10)):0);  
beginShape();  
vertex(150, 180);  
vertex(550, 180);  
vertex(550, 400);  
vertex(470, 420);  
vertex(470, 560);  
vertex(230, 560);  
vertex(230, 420);  
vertex(150, 400);  
endShape(CLOSE);  
}  
beginShape();  
fill(200, 255, 255);  
vertex(150, 180);  
vertex(550, 180);  
vertex(550, 400);  
vertex(470, 420);  
vertex(470, 560);  
vertex(230, 560);  
vertex(230, 420);  
vertex(150, 400);  
endShape(CLOSE);  
//lineas bok
```



```
strokeWeight(8);
stroke(0);
line(260, 480, 260, 530);
line(340, 480, 340, 530);
line(420, 480, 420, 530);
```

```
for(i=0;i<7;i++){
```

4.NARIZ:

Las fórmula que siguen, permiten que la nariz se mueva sobre el background.

```
pushMatrix();
translate(px,py);
fill(c);
scale(1.5);
switch(i){
case 1:
```

```
beginShape(TRIANGLES);
vertex(-20,-8);
vertex(8,48);
vertex(-50,48);
endShape();
break;
```

```
}
popMatrix();
}
if( mousePressed && (has != -1) ){
px+= mouseX-pmouseX;
py+= mouseY-pmouseY;
}
}
```

Las fórmula que siguen, permiten que los ojos se mueva sobre el background.

```
d.remove(mouseX,mouseY);
d.drag(mouseX,mouseY);
d.visualizacion();
```

```
// prueba si el cursor está sobre el ojo
if (mouseX > bx-eyeSize && mouseX < bx+eyeSize &&
    mouseY > by-eyeSize && mouseY < by+eyeSize) {
    overBox = true;
    if(!locked) {
        stroke(255);
        fill(153);
    }
} else {
    stroke(153);
    fill(153);
    overBox = false;
}

// Dibuja los ojos
strokeWeight(8);

stroke(0);
fill (100,12,230);
ellipse(bx+70, by+50, eyeSize+30, eyeSize+30); //ojo1
ellipse(bx+270, by+50, eyeSize, eyeSize); //ojo2
```

6.BOCA:

Las fórmula que siguen, permiten que la bova se mueva sobre el background.

```
if((mousePressed &&(((nposx<mouseX)&& (mouseX<nposX))&& ((nposy<mouseY)&&
(mouseY<nposY))))){

    nimgX = mouseX-nmX;
    nimgY = mouseY-nmY;

}else{

    image(nimg,nimgX,nimgY);//se dibuja la imagen q va a ser arrastrada
```

El if que sigue,

Establece el rango, donde el if prueba si la imagen arrastrada pasa por cierta posición

Para luego hacer que la misma sea reemplazada por otra imagen y así de la sensación de que la pieza es atraída hacia la posición correcta del rompecabezas.

if(((nposx<nimgX-10)&& (nimgX-10<nposX))&& ((nposy<nimgY-10)&& (nimgY-10<nposY))){//esto me permite establecer el rango por donde la imagen arrastrada va a pasar sobre otra 2da imagen y esta sea la que tome su lugar ;así da a entender q fue atraída

```
tint(0);
image(nimg,nimgX,nimgY);
noTint();
}

}

if(((nposx<nimgX-10)&& (nimgX-10<nposX))&& ((nposy<nimgY-10)&& (nimgY-10<nposY))){//le puse menos 10 para q el espacio de deteccion sea menor
fill(153);
rect (500,400,600,400);
fill(255);
 textSize(100);
text("GANASTE", 430, 550); //
image(nimg,nposx,nposy);
}else{
image(nimg,nimgX,nimgY);
//}
} //boka //cierre BOk
} //cierre draw
```

VOID MOUSEPRESSED: eventos que suceden cuando se presiona los botones del Mouse.

```
void mousePressed() {
```

1. Nariz

```
if( mouseButton == LEFT ){
color mouseC = get(mouseX,mouseY);
for(int j=0;j<7;j++){
if( c == mouseC ){
has = j;
}
}
}

}
```

2.Ojos

```

d.clicked(mouseX,mouseY);
if(overBox) {
    locked = true;
    fill(255, 255, 255);
} else {
    locked = false;}
xOffset = mouseX-bx;
yOffset = mouseY-by;

```

```

//boka//cierre BOk

```

```

nmX = mouseX-nimgX;
nmY = mouseY-nimgY;
//boka//cierre BOk

```

```

void mouseReleased() {
d.stopDragging();
locked = false;

//nariz:
has = -1;
for(i=0;i<7;i++){
    if( px < 0 || px> width || py< 0 || py> height ){
        px=150;
        py=150;
    }
}
}
}

```

```

void mouseDragged() {
    if(locked) {
        bx = mouseX-xOffset;
        by = mouseY-yOffset;
    }
}

```

```

}

```

3.Boca

```

void keyPressed()
{
    centerImage();
}

```

Void centerImage: Ubica la posición inicial de la boca en el background.

```
void centerImage()  
{  
  nimgX = 700;  
  nimgY = 680;  
}
```

LINK PARA VER VIDEO EXPLICATIVO:

<http://youtu.be/1WSTQ4BDRdQ>