

Taller de processing - Primera Pantalla 2014

Esta obra está bajo una licencia de Creative Commons
[Reconocimiento-CompartirIgual 4.0 Internacional](https://creativecommons.org/licenses/by-sa/4.0/).



Primeros pasos

Vamos a ir siguiendo los pasos juntos, con ayuda de sketchpad.

<http://studio.sketchpad.cc/primerapantalla2014>

Dibujamos una línea desde 0,0 hasta 100,100.

```
line(0,0,100,100);
```

Estructura básica de un programa

Los programas que hemos hecho hasta ahora son programas estáticos. Esto significa que nunca cambian. Se ejecutan una única vez y cuando llegan a la última línea de código, se paran.

Si queremos que un programa sea interactivo, sólo es posible si la ejecución es continua. Vamos a usar dos funciones que tiene processing. Una de ellas se llama setup y lo que pongamos dentro sólo se ejecuta una vez, al principio de todo. La segunda se llama draw y se ejecuta de forma continua 60 veces por segundo (esto se puede cambiar, lo veremos después).

```
void setup(){
  size(300,300);
}

void draw(){
  line(0,0,300,300);
}
```

Línea hasta la posición del ratón

Veamos cómo añadirle ahora interactividad. Vamos a usar las variables mouseX y mouseY que nos van a dar la posición actual del ratón.

```
void setup(){
  size(300,300);
}

void draw(){
  line(0,0,mouseX,mouseY);
}
```

Dibujar un círculo en la posición del ratón

Podemos pintar otras formas, como por ejemplo elipses y circunferencias.

```
void setup(){
  size(300,300);
  noStroke(); // no pintamos línea de contorno
}

void draw(){
  background(255); // borra todo y fondo en blanco
  fill(255,0,0); // color para relleno rojo
  ellipse(mouseX,mouseY,10,10);
}
```

Dibujar sólo si el ratón está pulsado

Si queremos que pinte sólo cuando el ratón está pulsado, tendremos que leer el estado del ratón en la variable `mousePressed`.

```
void setup(){
  size(300,300);
  noStroke(); // no pintamos linea de contorno
}

void draw(){
  fill(255,0,0); // color para relleno rojo
  if (mousePressed){
    ellipse(mouseX,mouseY,10,10);
  }
}
```

Fondo continuo que va pasando

Vamos a ver cómo hacer que nuestro minijuego tenga un fondo que va pasando de forma continua de derecha a izquierda. Esto genera la sensación de que nuestro personaje va hacia la derecha, aunque lo pintemos siempre en el mismo sitio. Nosotros entonces, sólo controlaremos el movimiento vertical de nuestro personaje. Comenzaremos pintando una elipse que se mueva como el fondo y luego usaremos la misma estructura pero pintando el fondo.

Elipse que va de derecha a izquierda

Introducimos el concepto de variable. Almacenamos en la variable `fondoX` la posición en la coordenada X del círculo. Vamos a ir pintando un círculo y después desplazándolo por la pantalla hacia la izquierda, para eso cada vez que se ejecute `draw`, `fondoX` valdra un pixel menos. Si queremos que vaya más deprisa, podemos probar a restarle dos o más.

```
int fondoX;

void setup(){
  size(300,300);
  noStroke();
  fondoX = 300;
}

void draw(){
  background(255);
  fill(255,0,0);
  ellipse(fondoX,50,10,10);
  fondoX = fondoX - 1;
}
```

Cuando desaparece por la izquierda reaparece por la derecha

El problema del programa anterior es que una vez que el círculo ha pasado de derecha a izquierda, se sale de la ventana y no lo vemos. Vamos a usarl de nuevo la estructura condicional. Cuando el valor de `fondoX` sea menor que cero, lo cambiamos por la anchura de la pantalla que me la da la variable `width`. De esta forma, cuando el círculo se pierda por la izquierda, volverá a salir por la derecha

```

int fondoX;

void setup(){
  size(300,300);
  noStroke();
  fondoX = width;
}

void draw(){
  background(255);
  fill(255,0,0);
  ellipse(fondoX,50,10,10);
  fondoX = fondoX - 1;
  if (fondoX < 0){
    fondoX = width;
  }
}

```

Ahora ponemos la imagen de fondo

Cambiamos el círculo por el fondo. Para colocar una imagen primero tenemos que añadir la imagen al sketch. Luego lo cargamos en una variable de imagen en setup y la colocamos con la función image(). El ajuste para que cuando termina el fondo se vuelva a pintar es un poco más complejo que con un círculo. Una posible solución es pintar dos veces la imagen, para que no salga cortada.

```

int fondoX;
PImage fondo; // variable para guardar una imagen

void setup(){
  size(200,225);
  smooth();
  fondoX = 0;
  fondo = loadImage("fondo1.jpg"); // cargamos la imagen
}

void draw(){
  background(255);
  image(fondo,fondoX,0); //pintamos la imagen
  image(fondo,fondoX+600,0);
  fondoX = fondoX - 1;
  if (fondoX < -600){
    fondoX = 0;
  }
}

```

Personaje movido por raton

Podemos introducir un personaje que esté movido por el ratón. Podemos usar solamente mouseX y dejarlo en una posición fija en horizontal o moverlo por toda la pantalla. En el ejemplo solamente usamos movimiento vertical. Usamos la función imageMode(CENTER) o imageMode(CORNER) para que al dibujar la imagen, el punto de inicio sea la esquina superior izquierda o el centro de la imagen. En el caso del personaje además especificamos el tamaño al que se mostrará 45x45 píxeles. Faltaría limitar la posición del personaje para que no se salga por arriba y por abajo

```

int fondoX;
PImage fondo, superman;

void setup(){
  size(200,225);
  smooth();
  fondoX = 0;
  fondo = loadImage("fondo1.jpg");
  superman = loadImage("superman.png");
}
void draw(){
  background(255);
  imageMode(CORNER);
  image(fondo, fondoX, 0);
  image(fondo, fondoX+600, 0);
  imageMode(CENTER);
  image(superman,width/3,mouseY,45,45);
  fondoX = fondoX - 1;
  if (fondoX < -600){
    fondoX = 0;
  }
}

```

Fondo que se va moviendo según el personaje avanza

Otra posibilidad para nuestro juego es que nosotros controlemos el movimiento de avance o retroceso de nuestro personaje. En este caso también pintaremos al personaje en el centro de la pantalla y cuando queramos que se mueva, lo que haremos es mover el fondo. Para eso necesitamos leer el teclado vamos a considerar que la tecla A mueva al personaje hacia la izquierda y la tecla D hacia la derecha. Para leer el teclado, usaremos una función de processing que se llama `keyPressed`. Esta función se ejecuta cuando una tecla es pulsada. En ella comprobaremos qué tecla se ha pulsado y cambiaremos la posición del fondo.

Fondo con un cubo de personaje

En este ejemplo hemos colocado una simple caja como personaje, para probar nuestro programa. Hemos tenido en cuenta que el personaje no pueda salirse del fondo.

```

int fondoX;
PImage fondo;

void setup(){
  size(200,200);
  smooth();
  fondoX = 0;
  fondo = loadImage("fondo2.jpg");
}

void draw(){
  background(255);
  image(fondo,fondoX,0); //pintamos el fondo
  rectMode(CENTER);

```

```

    rect(width/2,160,20,20); //pintamos el personaje caja
    //comprobamos si el personaje se sale del fondo
    if (fondoX > 0){
        fondoX = 0;
    }
    if (fondoX < -714){
        fondoX = -714;
    }
}

void keyPressed(){
    if (key=='a'){
        fondoX = fondoX + 2;
    }
    if (key=='d'){
        fondoX = fondoX - 2;
    }
}

```

Fondo con imagen de personaje

En este segundo ejemplo, hemos puesto un personaje. Fíjate que hemos tenido que incluir dos imágenes distintas, según si el personaje mira a la derecha o a la izquierda. Dependiendo del último movimiento, lo pintaremos mirando hacia un lado o hacia el otro.

```

int fondoX;
PImage fondo, perAv, perRet;
boolean avanzar = true;

void setup(){
    size(200,200);
    smooth();
    fondoX = 0;
    fondo = loadImage("fondo2.jpg");
    perAv = loadImage("personaje1.png");
    perRet = loadImage("personaje1b.png");
}

void draw(){
    background(255);
    imageMode(CORNER);
    image(fondo,fondoX,0); //pintamos el fondo
    imageMode(CENTER);
    if (avanzar) {
        image(perAv,width/2,160);
    } else {
        image(perRet,width/2,160);
    }
    if (fondoX > 0){
        fondoX = 0;
    }
    if (fondoX < -714){
        fondoX = -714;
    }
}

```

```

void keyPressed(){
  if (key=='a'){
    fondoX = fondoX + 2;
    avanzar = false;
  }
  if (key=='d'){
    fondoX = fondoX - 2;
    avanzar = true;
  }
}

```

Obstáculos y colisiones

Lo siguiente que vamos a añadir a nuestro juego es que el personaje recolecte cosas o que tenga que esquivarlas. Para ello tendremos que calcular cuando choca nuestro personaje con un obstáculo. Vamos a almacenar en variables las posiciones en horizontal y vertical (x,y) del personaje y del obstáculo y usaremos la función dist de processing para calcular la distancia entre ellos. Cuando sea menor que la anchura de uno de ellos consideraremos que han chocado y por tanto restamos una vida. También podría ser que sumáramos puntos al recolectar los objetos. Fíjate como usamos una función nueva, `text(texto, x, y)`, para mostrar en la posición x, y el texto con las vidas que tenemos. Tal y como está el programa tiene algún que otro problemilla. Busca una forma imaginativa de solucionarlo.

```

int fondoX;
PImage fondo, superman, obstaculo;
int perX, perY;
int obsX, obsY;
int vidas = 5;

void setup(){
  size(200,225);
  smooth();
  fondoX = 0;
  fondo = loadImage("fondo1.jpg");
  superman = loadImage("superman.png");
  obstaculo = loadImage("pelota2.png");
  perX = width/4;
  obsY = int(random(height));
  obsX = width;
}

void draw(){
  background(255);
  imageMode(CORNER);
  image(fondo, fondoX, 0);
  image(fondo, fondoX+600, 0);
  imageMode(CENTER);
  // ajuste del movimiento del personaje
  if (mouseY < 23) {
    perY = 23;
  } else if (mouseY > height - 40) {
    perY = height - 40;
  } else {
    perY = mouseY;
  }
}

```

```

image(superman,perX,perY,45,45); //pinto al personaje
// movimiento del obstáculo
obsX = obsX - 2;
image(obstaculo,obsX,obsY,45,45); //pinto el obstáculo

// compruebo la distancia si es menor resto vidas
if (dist(perX, perY, obsX, obsY) < 45) {
    vidas = vidas - 1;
}
//pintamos el numero de vidas
textSize(14);
text("vidas: "+vidas,15,20);
// si se pierde el obstáculo genero uno nuevo
if (obsX < 0) {
    obsY = int(random(height));
    obsX = width;
}

fondoX = fondoX - 1;
if (fondoX < -600){
    fondoX = 0;
}
}

```

Añadiendo música

Este paso es totalmente opcional, pero qué sería de un videojuego sin su música... Para eso tenemos que usar una librería añadida a processing. Añade el archivo mp3 al sketch y luego escribe en el programa los elementos en negrita, cambiando el nombre del archivo `musica.mp3` por el que tenga tu canción.

```

import ddf.minim.*;

Minim minim;
AudioPlayer musica;

// aquí van el resto de declaraciones de variables

void setup(){
    minim = new Minim(this);
    musica = minim.loadFile("musica.mp3");
    musica.loop();
    musica.play();

    // aquí va el resto del setup
}

void draw(){

    // nuestro programa sin añadir nada
}

```