

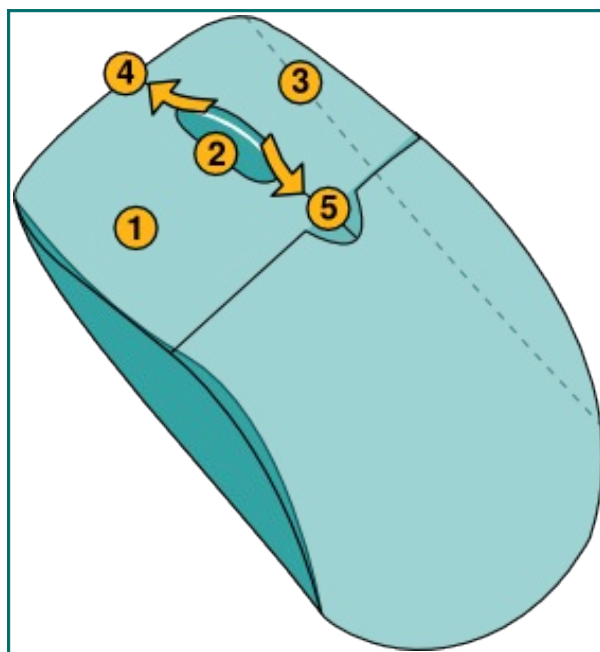
Eventos de Ratón

Por: Julio Fabio De La Cruz Gomez



En processing el ratón es un periférico que puede ser fácilmente accedido para así lograr interactividad ya sea con el movimiento o presión de los botones del ratón.

Gráfico 1 - Botones del ratón



tomada de: wooptoo_Mouse_buttons.svg

1 - LEFT 2 - CENTER 3 - RIGHT 5,6 - MOUSEWHEEL

EL RATÓN

Desde processing podemos acceder a estados del ratón como su posición en X y Y así como que botón o botones son presionados.

Según el **gráfico 1** se muestra la disposición de los botones que es de la siguiente manera:

- **Botón 1** : Es el botón izquierdo para referirnos a si este a sido presionado o no en processing se utiliza la palabra **LEFT**.
- **Botón 2** : Es el botón central para referirnos a si este a sido presionado o no en processing se utiliza la palabra **CENTER**.
- **Botón 3**: Es el botón derecho para referirnos a si

este a sido presionado o no en processing se utiliza la palabra **RIGHT**.

- **Eventos 4 y 5**: estas corresponden a lo que se llama **MouseWheel**, que aunque processing no tiene por defecto estos eventos de ratón implementados, pero podemos acceder a estos con algo de programación extra. Con el **MouseWheel** se detecta el sentido de giro.

VARIABLES DE SISTEMA DEL RATÓN

Son variables que indican o almacenan algún estado o acción del ratón. Estas variables no necesitan ser declaradas.

La **tabla 1** lista estas variables con una pequeña descripción.

mouseX - pmouseX

Esta variable es de tipo entero y almacena la posición cartesiana X del ratón con respecto al área de graficación. Si se utiliza dentro de la función **draw()** constantemente actualiza la posición.

La variable **pmouseX** almacena la posición que tenía el ratón antes de moverse y actualizarse en **mouseX**. Si el ratón deja de moverse tanto **pmouseX** y **mouseX** tendrán el mismo valor.

mouseY - pmouseY

Esta variable es de tipo entero y almacena la posición cartesiana Y del ratón con respecto al área de graficación. Si se utiliza dentro de la función **draw()** constantemente actualiza la posición.

La variable **pmouseY** almacena la posición que tenía el ratón antes de moverse y actualizarse en **mouseY**. Si el ratón deja de moverse tanto **pmouseY** y **mouseY** tendrán el mismo valor.

mousePressed

Es una variable de tipo booleano, que almacena el valor **TRUE** cuando cualquiera de los botones del ratón es presionado, en caso contrario almacena el valor **FALSE**. Se utiliza frecuentemente dentro de estructuras condicionales.

```
if(mousePressed == TRUE){
}
else{
}
```

también es equivalente a las anteriores instrucciones. Pues una estructura condicional evalúa si una expresión es falsa o verdadera.

```
if(mousePressed){
}
else{
}
```

mouseButton

Tabla 1 - Variables de sistema para el ratón.

Variable	Descripción
mouseX	Almacena la posición de la coordenada x del ratón.
mouseY	Almacena la posición de la coordenada y del ratón.
pmouseX	Almacena la posición de la coordenada x previa del ratón.
pmouseY	Almacena la posición de la coordenada y previa del ratón.
mousePressed	Variable de tipo booleano que indica si algún botón es presionado con el dato TRUE .
mouseButton	Variable que indica que botón a sido presionado. LEFT , CENTER RIGHT . Los valores son de tipo entero y se mantienen hasta que sea presionado otro botón.

raton01.pde - Programa de dibujo con el ratón

```
void setup(){
  size(400,400);
  background(255);
}
void draw(){
  if(mousePressed){
    if(mouseButton == LEFT){
      stroke(0);
      line(pmouseX,pmouseY,mouseX,mouseY);
    }
    if(mouseButton == RIGHT){
      stroke(255);
      line(pmouseX,pmouseY,mouseX,mouseY);
    }
    if(mouseButton == CENTER){
      saveFrame("dibujo.tif");
    }
  }
}
```

En esta variable se indica que botón se ha presionado del ratón, se utiliza normalmente junto con la variable **mousePressed**.

Los valores almacenados son de tipo entero, estos valores se comparan con valores ya definidos empleando las palabras **LEFT**, **CENTER** y **RIGHT**.

```
if(mousePressed){
if(mouseButton == LEFT){
}
}
}
```

El ejemplo **raton01** consiste en un programa de dibujo en el cual cuando se presiona el botón izquierdo del ratón dibuja, con el botón derecho borra y con el botón central guarda la imagen como **dibujo.tif**. Se puede utilizar **dibujo-##.tif** para guardar diferentes imágenes enumeradas.

EVENTOS DE RATÓN

Son funciones que se ejecutan cuando ocurren acciones sobre el botón, estas acciones se conocen como eventos.

Esas funciones son empleadas para hacer mas fácil la escritura del programa. En la **tabla 2** se listan las funciones y una pequeña descripción de estas, mientras el código **raton02** muestra un esquema de la utilización de estas funciones.

Para utilizar estas funciones solo se coloca el código que se requiera que se ejecute la acción que este asociada al ratón.

EJEMPLOS

raton03

Ejemplo con **mouseMoved()**. La intensidad del color de fondo cambia cuando el ratón es movido. La variable intensidad se incrementa en uno cada vez que se mueve el ratón.

Tabla 2 - Tabla de instrucciones

Función	Descripción
mousePressed()	El botón es presionado.
mouseClicked()	El botón es presionado y soltado.
mouseReleased()	El botón es soltado.
mouseMoved()	Cuando se mueve el ratón.
mouseDragged()	El botón se mantiene presionado y se mueve.

raton02.pde - Esquema programación de eventos de ratón.

```
void setup(){
// Parametros de configuracion
// Inicializacion de variables
}
void draw(){
// Funcion principal de animacion y dibujo
//Codigo principal del programa.
}
void mouseClicked() {
//Codigo que se quiere ejecutar cuando
// Cuando el boton es presionado y soltado
}
void mousePressed() {
//Codigo que se quiere ejecutar cuando
// Cuando el boton es presionado
}
void mouseReleased() {
//Codigo que se quiere ejecutar cuando
// Cuando el boton es soltado
}
void mouseMoved() {
//Codigo que se quiere ejecutar cuando
// Cuando el raton es movido
}
void mouseDragged() {
//Codigo que se quiere ejecutar cuando
// Cuando el boton es presionado, sostenido y movido
}
```

raton04

Ejemplo con **mousePressed()**.

La intensidad del color de fondo cambia cuando cualquier botón del ratón es presionado. La variable intensidad se incrementa en uno cada vez que se presiona el ratón, para que suceda nuevamente el evento hay que soltar el botón y volverlo a presionar nuevamente.

raton05

Ejemplo con **mousePressed**.

La intensidad del color de fondo cambia cuando cualquier botón del ratón es presionado. La variable intensidad se incrementa en uno cada vez que se presiona el ratón, pero a diferencia del **ejemplo 4** la variable se incrementa permanentemente si un botón es presionado permanentemente a diferencia de **mousepressed()**.

raton06

Ejemplo con **mouseDragged()**.

La intensidad del color de fondo cambia cuando cualquier botón del ratón es presionado y el ratón movido. La variable intensidad se incrementa en uno cada vez que se mantiene presionado un botón del ratón y este se mueve.

raton07

Ejemplo con **mouseReleased()**.

La intensidad del color de fondo cambia cuando cualquier botón del ratón es soltado. La variable intensidad se incrementa en uno cada vez que se presiona el ratón, para que suceda nuevamente el evento hay que presionar el botón y volverlo a soltar nuevamente.

raton08

Ejemplo con **mouseX** y **mouseY**.

Se muestra las coordenadas X y Y de la posición del ratón en la zona de graficación de processing.

raton03.pde - Programa

```
int intensidad;
void setup(){
  size(400,400);
  intensidad = 0;
}
void draw() {
  background(intensidad);
}
void mouseMoved() {
  intensidad = intensidad + 1;
  if (intensidad > 255) {
    intensidad = 0;
  }
}
```

raton04.pde - Programa

```
int intensidad;
void setup(){
  size(400,400);
  intensidad = 0;
}
void draw() {
  background(intensidad);
}
void mousePressed() {
  intensidad = intensidad + 5;
  println("Intensidad = "+intensidad);
  if (intensidad > 255) {
    intensidad = 0;
  }
}
```

raton05.pde - Programa

```
int intensidad;
void setup() {
  size(400,400);
  intensidad = 0;
}
void draw() {
  background(intensidad);
  if(mousePressed) {
    intensidad = intensidad + 5;
    println("Intensidad = "+intensidad);
    if (intensidad > 255) {
      intensidad = 0;
    }
  }
}
```

raton09

Ejemplo con **mouseWheel()**.

Por defecto processing no tiene una función para registrar el evento de **SCROLL** de la rueda del ratón. Para hacer esto se utiliza una clase de Java para implementar esta función.

El código fue tomado de: http://wiki.processing.org/index.php/Wheel_mouse @author Rick Comanje

raton10

Ejemplo con **mousePressed** y **mouseButton**

El color del fondo de la pantalla cambia según el botón del ratón presionado.

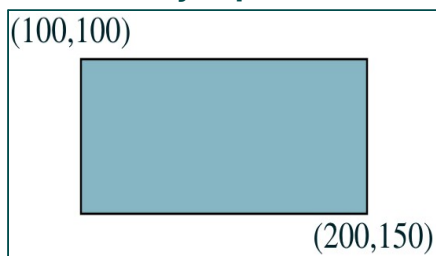
raton11

Ejemplo con **mousePressed**, **mouseX** y **mouseY**.

En este ejemplo se muestra como hacer un botón rectangular que al presionarse cambie de color.

La instrucción **rectMode(CORNERS)** se utiliza para especificar las esquinas del rectángulo para dibujarlo el **gráfico 2** muestra las coordenadas para el rectángulo de este ejemplo.

Gráfico 2 - Ejemplo botón

**raton06.pde - Programa**

```
int intensidad;
void setup(){
  size(400,400);
  intensidad = 0;
}
void draw() {
  background(intensidad);
}
void mouseDragged() {
  intensidad = intensidad + 5;
  println("Intensidad = "+intensidad);
  if (intensidad > 255) {
    intensidad = 0;
  }
  ellipse(mouseX,mouseY,50,50);
}
```

raton07.pde - Programa

```
int intensidad;
void setup() {
  size(400,400);
  intensidad = 0;
}
void draw() {
  background(intensidad);
}
void mouseReleased() {
  intensidad = intensidad + 5;
  println("Intensidad = "+intensidad);
  if (intensidad > 255) {
    intensidad = 0;
  }
}
```

raton08.pde - Programa

```
void setup() {
  size(400,400);
}
void draw() {
  println("X="+mouseX);
  println("Y="+mouseY);
}
```

raton09.pde - Programa

```
import java.awt.event.*;
void setup() {
  addMouseListener(new MouseWheelListener() {
    public void mouseWheelMoved(MouseWheelEvent mwe) {
      mouseWheel(mwe.getWheelRotation());
    }
  });
}
void draw() {
}
void mouseWheel(int delta) {
  println("se ha girado " + delta + " unidades");
}
```

raton10.pde - Programa

```
void setup() {
  size(400,400);
}
void draw() {
  if(mousePressed) {
    if(mouseButton==LEFT) {
      background(255,0,0);
    }
    if(mouseButton==CENTER) {
      background(0,255,0);
    }
    if(mouseButton==RIGHT) {
      background(0,0,255);
    }
  }
}
```

raton11.pde - Programa

```
void setup(){
  size(400,400);
  rectMode(CORNERS);
}
void draw(){
  background(255);
  fill(255,0,0);
  rect(100,100,200,150);
  if(mousePressed){
    if( (mouseX>=100) && (mouseX<=200) && (mouseY>=100) && (mouseY<=150) ){
      fill(0,255,0);
      rect(100,100,200,150);
    }
  }
}
```