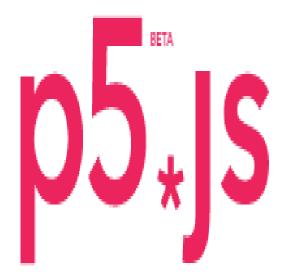
2025/11/04 18:28 1/4 nombre proyecto 2



Processing creativity times JavaScript dynamism

Esta guia se hace en referencia a este gran tutorial de Allison Parrish:Link

La idea sera conocer la librería con ejemplos sencillos y muy funcionales. Al final se compartirá un juego escrito usando la biblioteca y se explicara a modo raso el código y como se crearon las animaciones usando flash para generar los spritesheets. Si desea conocer el juego directamente, haga click aqui

Programación de juegos con play.js

En este tutorial, vamos a ver como funciona una biblioteca de la librería p5 llamada p5.play.js. Escrita por **Paolo Pedercini**p5-play

Esta va a ser una guía muy simple, con algunos ejemplos para entender como funciona a modo raza la librería. Por favor siempre este consultando la referencia para conocer mas a fondo la librería.

Play.p5.js es una librería que ofrece una serie de objetos y funciones útiles para escribir videojuegos y otras aplicaciones interactivas. Los objetos y funciones que introduce se incorporan en p5.js, es como si fuera una biblioteca del framework.

Instalación

En el sitio oficial esta un guía de referencia para instalar la biblioteca. Puede descargar la biblioteca haciendo click

agui

; después de descomprimir el archivo, vaya al directorio libraries y copie el archivo *p5.play.js* y péguelo en la biblioteca libraries de su directorio. Hay que recalcar que en su propio vocero index.html, no olvide usar la etiqueta script para importar la biblioteca a su vocero. Si desea mas información al respecto, Aquí hay un gran tutorial de como importar esta librería a su proyecto.

Sprite

Para crear un elemento en el mundo de *p5.play* logramos esto usando la función **createSprite()**. Esta función devuelve un objeto Sprite, que a su vez posee una serie de atributos y métodos que nos permite consultar y modificar las propiedades del sprite.



Es muy importante que si desea usar los ejemplos aquí descritos, use un servidor local para correr los ejemplos.

Un ejemplo muy sencillo aquí para la creación de un sprite:

```
var forma;
function setup() {
  createCanvas(400, 400);
  forma = createSprite(
    width/2, height/2, 40, 40);
  forma.shapeColor = color(255);
  forma.velocity.y = 0.5;
}
function draw() {
  background(50);
  drawSprites();
}
function mousePressed() {
  forma.position.x = mouseX;
  forma.position.y = mouseY;
}
```

La función **createSprite()** para poderse crear toma cuatro parámetros. La posición en **X** y en **Y** ademas del **ancho** y el **alto**. El atributo **.shapeColor** es una función que recibe como parámetro un color, que en este caso establecerá el color de nuestro sprite, que por defecto siempre sera un cuadrado. Para que la librería p5.play pueda mostrar el sprite, tenemos que añadir la función drawSprites() antes del final de la función draw().

Cada objeto creado con **createSprite()**, posee ademas atributos como la **posición** y la **velocidad**. Ambos se pueden configurar para controlar o establecer la posición y la velocidad del sprite. La biblioteca *p5.play* se encarga de actualizar la velocidad o de preguntar la posición actual para nosotros. Y por ello no tendremos que preocuparnos de las matemáticas complejas que hay detrás de ese proceso.

En el ejemplo anterior el sprite se mueve constantemente hacia abajo. Su posición inicial siempre

https://wiki.unloquer.org/ Printed on 2025/11/04 18:28

2025/11/04 18:28 3/4 nombre proyecto 2

sera fijada a la posición en X y en Y del mouse cuando se hace click sobre el lienzo.

Moviendo los sprites

Como mencionamos anteriormente, podemos establecer una velocidad inicial o una posición inicial para nuestro sprite. Para establecer la velocidad directamente, referenciamos el atributo .velocity.x para la coordenada en x y .velocity.y para la coordenada en y. Si lo que queremos es indicar velocidades fijas a una dirección determinada, usamos la función setSpeed(). En el ejemplo que vamos a ver a continuación, desplazamos el sprite usando las teclas de nuestro teclado:

```
var forma:
function setup() {
  createCanvas(400, 400);
  forma = createSprite(
    width/2, height/3, 40, 40);
  forma.shapeColor = color(255);
}
function draw() {
  background(50);
  fill(255);
  noStroke();
  textAlign(CENTER, CENTER);
  text("use arrow keys, or SPACE to stop",
    width/2, height*0.67);
  drawSprites();
}
function keyPressed() {
  if (keyCode == RIGHT ARROW) {
    forma.setSpeed(1.5, 0);
  }
 else if (keyCode == DOWN ARROW) {
    forma.setSpeed(1.5, 90);
  }
 else if (keyCode == LEFT ARROW) {
    forma.setSpeed(1.5, 180);
  }
 else if (keyCode == UP ARROW) {
    forma.setSpeed(1.5, 270);
  }
 else if (key == ' ') {
    forma.setSpeed(0, 0);
  }
  return false;
}
```

La variable key en *p5.js* solo funciona para los caracteres alfanuméricos. Con el fin de detectar las teclas de las flechas, usamos la variable **KeyCode**. No olvide escribirla con K mayúscula al iniciar.

Para añadir gravedad al dibujo, vasta con usar la función **.setSpeed()** para añadir una fuerza constante.

⁻ https://wiki.unloquer.org/

En las siguientes lineas de código hay un ejemplo que hace que un sprite se dibuje en la pantalla, para que se mueva hacia abajo en cada fotograba y luego rebota cuando llega a la parte inferior.

```
var forma;
function setup() {
  createCanvas(400, 400);
  forma = createSprite(width/2, height/2,
    40, 40);
  forma.shapeColor = color(255);
  forma.velocity.y = 0;
}
function draw() {
  background(50);
  if (forma.position.y >= height) {
    forma.velocity.y *= -1;
    // set to height to prevent "tunneling"
    forma.position.y = height;
  }
  // constant downward speed
  // (i.e., gravity)
  forma.addSpeed(0.25, 90);
  drawSprites();
function mousePressed() {
  forma.position.y = mouseY;
}
```

From:

https://wiki.unloquer.org/ -

Permanent link:

https://wiki.unloquer.org/proyectos/nombre proyecto 2?rev=1473198927

Last update: 2016/09/06 21:55



https://wiki.unloquer.org/ Printed on 2025/11/04 18:28