# SpriteSheeter

## Clase Sprite

Sprite es la clase encargada de manejar como imágenes, cada uno de los cuadrantes que forman un sprite sheet. Esta clase hace posible que cada uno de esos cuadrantes tenga asociados una serie de características, en lugar de tratarlos como simples imágenes.

public class Sprite {  
 private final int id**;** private final BufferedImage sprite**;**public Sprite(BufferedImage spriteSheet**,** int topXCoordinate**,** int topYCoordinate**,** int spriteSide**,** int id){  
 this.id = id**;** sprite = spriteSheet.getSubimage(topXCoordinate**,** topYCoordinate**,** spriteSide**,** spriteSide)**;** }  
  
 public int getId() {  
 return id**;** }  
  
 public BufferedImage getSprite() {  
 return sprite**;** }  
}

private final int id**;**private final BufferedImage sprite**;**

**id**: La ID define el valor por el que se identifica al sprite. Este dato será de utilidad cuando se declare un sprite anónimo.

**Sprite**: El objeto BufferedImage que representa la porción de imagen tomada del sprite sheet.

public Sprite(BufferedImage spriteSheet**,** int topXCoordinate**,** int topYCoordinate**,** int spriteSide**,** int id){  
 this.id = id**;** sprite = spriteSheet.getSubimage(topXCoordinate**,** topYCoordinate**,** spriteSide**,** spriteSide)**;**}

El constructor de la clase incluye los datos necesarios para poder crear la nueva imagen a partir de la imagen principal, y el número de ID que se le asignará al sprite.

**spriteSheet**: El objeto BufferedImage usado como referencia para crear el sprite.

**topXCoordinate**: La coordenada X superior izquierda del cuadrante que formará el sprite.

**topYCoordinate**: La coordenada Y superior izquierda del cuadrante que formará el sprite.

**spriteSide**: El valor del lado del cuadrante.

**id**: La ID que se le asigna al sprite.

0

1

2

3

3

2

1

0

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  | 16 pX |  |  |
|  | 16 pX |  |  |

**topXCoordinate** = 1

**topYCoordinate** = 2

**spriteSide** = 16 pixels

sprite = spriteSheet.getSubimage(topXCoordinate**,** topYCoordinate**,** spriteSide**,** spriteSide)**;**

Devuelve una subimagen definida por la region rectangular especificada.

public int getId() {  
 return id**;**}  
  
public BufferedImage getSprite() {  
 return sprite**;**}

Getters para los objetos id y sprite.

## Clase SpriteSheet

private final Map<String**,** int[][]> ID\_ARRAY\_MAP = new LinkedHashMap<>()**;**private final Map<String**,** BufferedImage> LAYERS = new LinkedHashMap<>()**;**private final Map<Integer**,** Sprite> SPRITES\_HASMAP = new HashMap<>()**;**private final List<String> HIDDEN\_LAYERS = new ArrayList<>()**;**private final BufferedImage POINTER\_LAYER**;**private final int INITIAL\_CANVAS\_SIZE**;**private final int SPRITE\_SIDE**;**private int tilesInColumn**;**private int tilesInRow**;**private String picturePath**;**

**ID\_ARRAY\_MAP**: El mapa encargado de vincular al nombre de cada capa de diseño, a un array bidimensional. Este array será el encargado de almacenar el conjunto de IDs de los sprite que forman el lienzo. Se explicará más adelante.

**LAYERS**: El mapa encargado de vincular al nombre de cada capa de diseño, el objeto BufferedImage de la misma.

**SPRITES\_HASMAP**: El mapa encargado de vincular cada sprite del sprite sheet, a una id.

**HIDDEN\_LAYERS**: Este array contiene el nombre de aquellos mapass