

## EXAMEN TEMA 3 DWECC

1)

Los cambios los he comentado en el código y he hecho commits.

2)

He creado el repositorio y he realizado el modulado del código.

3)

Vamos a hacer el debug poniendo un punto de interrupción en el bucle iterativo que genera las bolitas.

Sería en la línea 11 del archivo de JavaScript llamado "index.html".

En este archivo creamos un Array dentro del cuál vamos a meter las bolas generadas de manera aleatoria hasta llegar al cupo máximo de 25.

Este es el b́ucle que realiza esta acci3n:

```
while (balls.length < 25) { // ATENCI3N: cambiamos el ńmero de bolas
  const size = random(10, 20);
  const ball = new Ball(
    random(0 + size, width - size),
    random(0 + size, height - size),
    random(-7, 7),
    random(-7, 7),
    size
  );
  balls.push(ball);
}
```

El siguiente paso es mostrar esas bolas en el lienzo; para ellos necesitamos recorrer el Array en el cuál hemos introducido las bolas.

```
//recorremos el array mostrando las bolas en el lienzo
function loop() {
    ctx.fillStyle = 'rgba(0, 0, 0, 0.25)'; //FALLO: cambiamos el color
del fondo a negro
    ctx.fillRect(0, 0, width, height);

    for (const ball of balls) {
        ball.draw();
        ball.update();
        for (const otherBall of balls) {
            if (ball !== otherBall) {
                ball.collisionDetect(otherBall);
            }
        }
    }

    requestAnimationFrame(loop);
}

loop();
```

Después de esta introducción, vamos a realizar el debug.

Como se ha mencionado antes, el punto de interrupción en la línea 11.

```
11 while (balls.length < 25)
```

Al iniciar el debuggeo el VS Code nos crea un archivo .json en el cual debemos poner nuestra dirección para que funcione correctamente:

```
{
    // Use IntelliSense to learn about possible attributes.
    // Hover to view descriptions of existing attributes.
    // For more information, visit:
https://go.microsoft.com/fwlink/?linkid=830387
    "version": "0.2.0",
    "configurations": [
        {
            "type": "chrome",
            "request": "launch",
```

```

        "name": "Launch Chrome against localhost",
        "url": "http://127.0.0.1:5500",
        "webRoot": "${workspaceFolder}"
    }
]
}

```

Aquí podemos comprobar que el Array de bolas está vacío:

```

> Ball: class Ball {
> balls: (0) []

```

Una vez ejecutamos el programa entra en la primera repetición del bucle.

Podemos comprobar que se ha creado una bola porque la longitud del Array aumenta.

```

> [[Prototype]]: Object
length: 1

```

Nos crea la primera bola, dándole unos valores aleatorios de color ('rgb(153,64,92)'), de tamaño (16), de velocidad en los ejes X e Y (16, 3) y de posición en estos ejes (829, 279).

```

Module
  > Ball: class Ball {
  > balls: (1) [Ball]
    > 0: Ball {x: 829, y: 279, velX: 3, velY: 5, color: 'rgb(153,64,92)', ...}
      color: 'rgb(153,64,92)'
      size: 16
      velX: 3
      velY: 5
      x: 829
      y: 279
    > [[Prototype]]: Object
    length: 1

```

Ejecutamos una vez más y nos crea la segunda bola.

```
> [[Prototype]]: Object  
length: 2
```

Esta bola tiene un color aleatorio ('rgb(57,63,249)'), un tamaño de 14, unas velocidades en los ejes X e Y de 14 y de -4, una posición 1014 en el eje X y 269 en el Y.

```
▼ 1: Ball {x: 1014, y: 269, velX: -4, velY: 1, color: 'rgb(57,63,249)', ...}  
  color: 'rgb(57,63,249)'  
  size: 14  
  velX: -4  
  velY: 1  
  x: 1014  
  y: 269  
> [[Prototype]]: Object  
length: 2
```

La siguiente bola es la tercera:

```
> [[Prototype]]: Object  
length: 3
```

Tiene un color aleatorio ('rgb(116,194,80)'), un tamaño de 19, una velocidad en Y de -6, en X de -7, una posición de 493 en el eje X y 675 en el eje Y.

```
▼ 2: Ball {x: 493, y: 675, velX: -6, velY: -7, color: 'rgb(116,194,80)', ...}  
  color: 'rgb(116,194,80)'  
  size: 19  
  velX: -6  
  velY: -7  
  x: 493  
  y: 675  
> [[Prototype]]: Object  
length: 3
```

Si avanzamos hasta que el tamaño de Array sea 25, es decir, que esté completamente lleno de bolas, sale del bucle.