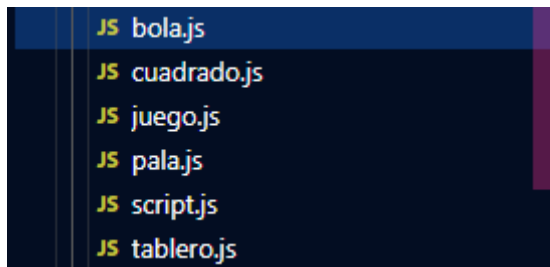


Practica_3_DIW_Victor_Valdes_Cobos

El juego es un Brick Breaker.

Hay 3 niveles y 3 vidas para cada nivel.

POO:



Bola, cuadrado, juego, tablero y pala son clases.

Cuando leo el tablero instancio todos los bloques en el tablero, la pala y la bola:

```

1  const nivel_3 =
2
3  [
4    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
5    [2, 2, 2, 2, 2, 2, 2, 2, 0, 0, 0, 2, 2, 2, 2, 2, 2, 2],
6    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
7    [2, 2, 2, 2, 2, 2, 2, 2, 0, 0, 0, 2, 2, 2, 2, 2, 2, 2],
8    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
9    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
10   [0, 2, 2, 0, 2, 2, 0, 2, 0, 0, 0, 2, 0, 2, 2, 0, 2, 2],
11   [0, 2, 2, 0, 2, 2, 0, 2, 0, 0, 0, 2, 0, 2, 2, 0, 2, 2],
12   [2, 2, 2, 2, 2, 2, 2, 2, 0, 0, 0, 2, 2, 2, 2, 2, 2, 2],
13   [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
14   [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1],
15   [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
16   [1, 1, 1, 1, 1, 1, 1, 2, 2, 1, 1, 2, 2, 1, 1, 1, 1, 1],
17   [0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 0, 0, 2, 0, 0, 0, 0, 0],
18   [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
19   [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
20   [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
21   [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
22   [0, 0, 0, 0, 0, 0, 0, 0, 7, 7, 7, 7, 7, 0, 0, 0, 0, 0],
23  ];

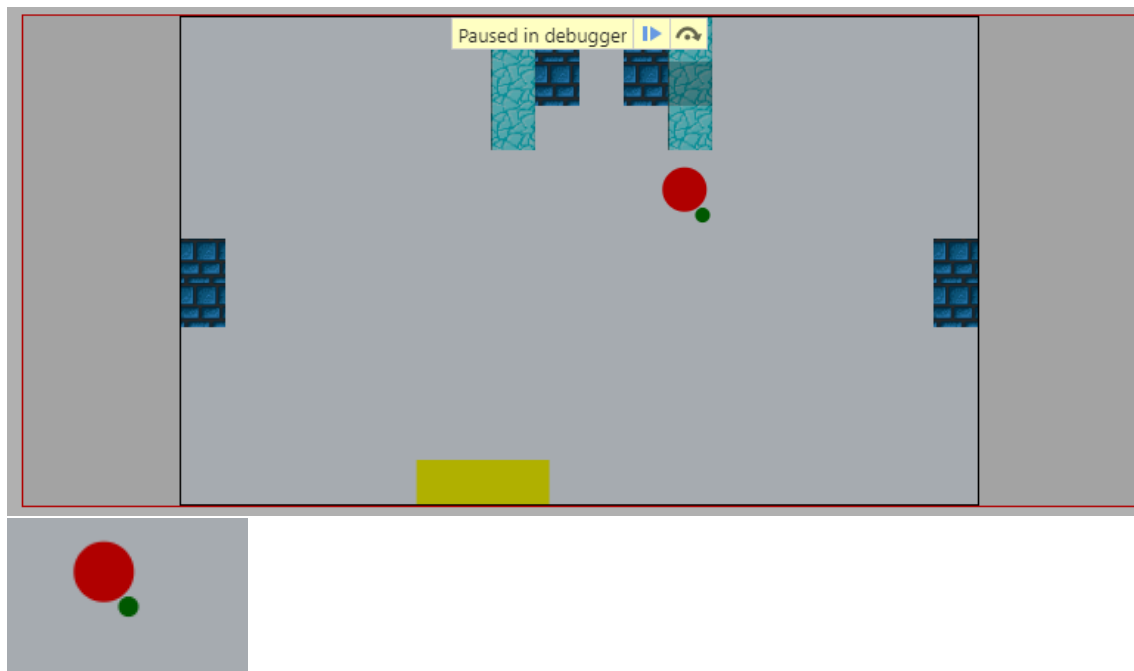
```

También uso muchas clases JSON, por ejemplo la propia clase comprobarColision devuelve un JSON colisión.

Lógica:

La gran mayoría de ella reside en tablero.js, donde se encuentra una de las funciones más importantes: comprobarColisionBola() en la que detecto la colisión y además asigno un nuevo vector unitario a la dirección de la bola, este ha sido uno de los mayores problemas de este juego.

Las 'hitbox' de la bola no es exacta debido a que mi forma de comprobar es esta:



Simulo el siguiente movimiento de la bola y le sumo su radio solo en ese vector, para que la hibox sea perfecta tendría que hacer un bucle que describiera un arco como el de la bola y comprobar colisión cada uno de estos puntos pero esto no me ha dado tiempo.

Los bloques son de tres tipos, siendo el primero el más débil y se van oscureciendo a medida que se van rompiendo.

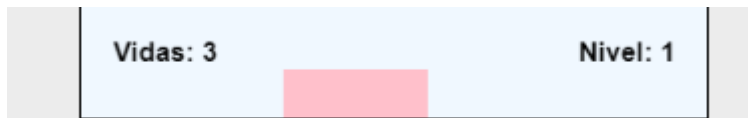
Los tableros son dinámicos y se pueden crear niveles en el archivo script.js al gusto, también se puede editar la variable MAXniveles para jugar más niveles aunque esto podría desbalancear el juego.

Cuando pierdes o ganas el juego aparece una pantalla que lo indica.



Indico un pequeño HUD.

Si pulsas a espacio tienes 5 segundos de habilidad:



La idea principal era que, al romper un bloque cayera un objeto y que al recogerlo con la tabla se activara pero esto no ha dado tiempo.

La bola describe un efecto cuando la pala se esta moviendo en la colisión.

Esta es la función principal del juego: (tablero.js)

```
524     animarTablero() {
525         return this._jugando = setInterval(() => {
526             this.actualizarTablero();
527             this.borrarTablero();
528             this.mostrarTablero();
529         }, 1000 / MAXfps);
530     }
```

Junto a esta función: (script.js)

```
21     function empezarJuego(e) {
22         menu.style.display="none";
23         main.style.display="none";
24
25
26         canvas.style.display="block";
27         canvas.width = nivel_1[0].length * tamCasilla;
28         canvas.height = nivel_1.length * tamCasilla;
29
30         console.log(nivel_1.length);
31         console.log(canvas.width + " " + canvas.height);
32
33         const juego = new Juego();
34
35         document.addEventListener("keydown", function (e) {
36             let LIMtablero = juego._arrNiveles[juego.nivelActual][juego.n
37             juego.Tablero._jsonElementos.pala.elemento.mover(e, LIMtabler
38         });
39
40         document.addEventListener("keyup", function () {
41             juego.Tablero._jsonElementos.pala.elemento.pararMovimiento(
42         });
43
44         document.addEventListener("keydown", function (e) {
45             if(e.key === " ") juego.Tablero._jsonElementos.pala.elemento
46         });
47     }
```