

# Juego 2D desarrollado en JavaScript puro: Introducción a la Informática

**JUAN DIEGO VELEZ SANCHEZ**  
SEPTIEMBRE DE 2020



# 1 CONTENIDO

---

1	CONTENIDO .....	1
2	PRESENTACIÓN .....	2
3	FASE 1: Dibujar y mover una bola .....	3
4	FASE 2: Rebotando en las paredes .....	7
5	FASE 3: Control de la paleta con el teclado .....	10
6	FASE 4: Fin de partes esenciales del juego .....	14
7	FASE 5: Muro de ladrillos.....	18
8	FASE 6: Detección de colisiones.....	23
9	FASE 7: Contar puntos, ganar, golpe y degradado de ladrillos.....	28
10	FASE 8: Controlando el ratón.....	36
11	FASE 9: Finalizando el juego .....	42
12	FASE 10: Movimiento de ladrillos, misil y explosión .....	49
13	FASE 11: Fin, menú, ganar y perder .....	57
14	CONCLUSIONES.....	72
15	BIBLIOGRAFÍA.....	73

## 2 PRESENTACIÓN

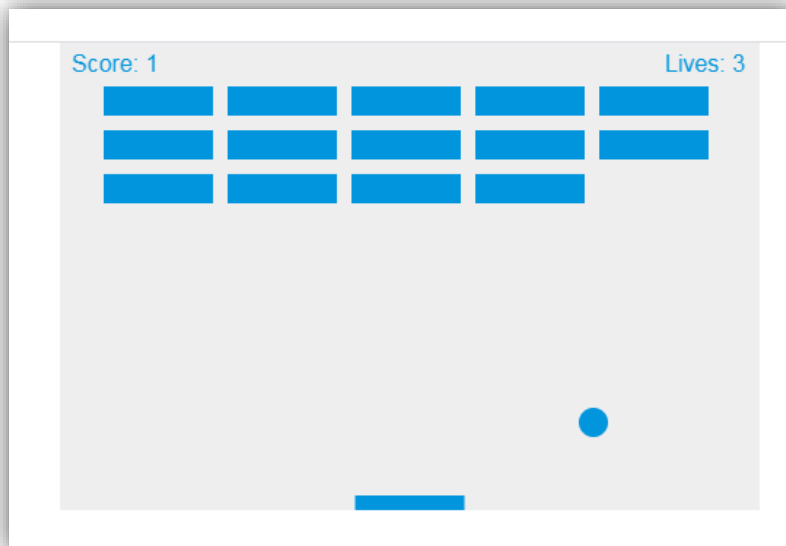
---

La presente monografía describe el desarrollo metódico de un juego 2D elaborado utilizando HTML5, CSS, CANVAS y JavaScript.

El juego elaborado se crea con JavaScript puro, utilizando un enfoque metódico en el cual se avanza de versión en versión, de modo que cada nuevo programa abarca un aspecto adicional del juego.

Cada una de las fases se cubre en un apartado diferente. Se plantea el alcance de cada una de ellas, se explican las instrucciones o conceptos que son necesarios para entender el significado del trabajo realizado, se agrega el código, y finalmente se presentan fotos de la ejecución del programa

Una vez cubiertas todas las fases, se dispondrá de un clásico juego 2D que servirá como base e inspiración para desarrollar otros programas aplicados en la Web.



Gráfica 1. Juego 2D en JavaScript

El documento web que sirve como referencia para el desarrollo del juego está en el siguiente enlace:

[https://developer.mozilla.org/es/docs/Games/Workflows/Famoso\\_juego\\_2D\\_usando\\_JavaScript\\_puro](https://developer.mozilla.org/es/docs/Games/Workflows/Famoso_juego_2D_usando_JavaScript_puro)

**AUTOR:** Juan Diego Vélez Sánchez



### 3 FASE 1: DIBUJAR Y MOVER UNA BOLA

---

El primer paso consiste en elaborar una página HTML básica. Agregaremos a dicha página un elemento HTML llamado CANVAS, un CANVAS es un elemento el cual se puede utilizar para dibujar generalmente con comandos en JavaScript, el cual nos servirá como base para el desarrollo del juego 2D.

El código JavaScript que operará sobre el CANVAS debe encerrarse entre las etiquetas `<script>...</script>`

Para una mejor y correcta visualización del CANVAS es necesario la adición de algunas características de estilo como por ejemplo el tamaño, el color etc. En el body de la página tenemos una etiqueta para darle un título, luego introducimos nuestro Canvas con su respectivo identificador. Una vez hecho esto, se procede a realizar la codificación que nos permitirá dibujar y comenzar con nuestro juego en JavaScript. En el inicio del código JavaScript se puede notar la inicialización de algunas variables para el correcto funcionamiento del juego con sus aspectos básicos, el contenedor del CANVAS con su respectiva instrucción “2d” que nos permitirá dibujar, las coordenadas en donde se encuentra la bola, el tamaño de esta y los valores de incremento para modificar su posición, vemos también variables similares a las de la bola original estas nos permitirá introducir una segunda bola.

Se definen tres funciones importantes. La primera de ellas, `dibujarBola()`, esta función recibe 4 argumentos que corresponden a las características de nuestra bola, los cuales los dos primeros son la posición, el tercer es el tamaño y el cuarto es el color, esta función permitirá crear diferentes bolas, dentro de esta tenemos diferentes métodos, los cuales nos permitirá dar el estilo a la bola, el color y las coordenadas según los argumentos introducidos, todas las instrucciones están entre los métodos `ctx.beginPath()` y `ctx.closePath()`, en el método `ctx.arc()`, los dos primeros argumentos que recibe son las coordenadas, o sea la posición donde aparecerá nuestra bola, los otros tres argumentos restantes servirán para darle el tamaño y la forma, para poder que sea redonda utilizamos la constante `PI`. La segunda función se llama `dibujar()`, pero antes de esta tenemos una variable llamada `azar` que contiene el resultado del método `Math.random()`, este método nos dará un número al azar que luego utilizaremos, la función `dibujar` es la encargada de limpiar el CANVAS, llamar la función `dibujarBola()` para que se pueda visualizar, dentro de esta hay una condición, de cumplirse permitirá dibujar una segunda bola, esta función también es la encargada de cambiar los valores de las coordenadas. Finalmente, la función `setInterval(dibujar, 20)`, llama a la función `dibujar` cada 20 milisegundos, sin este los dibujos no se visualizaran, como por ejemplo la bola nunca aparecerá en nuestro CANVAS.

El código es el siguiente:

```

1. <!DOCTYPE html>
2. <html>
3.
4. <head>
5.   <meta charset="UTF-8" />
6.   <title>Juego 2D</title>
7.
8.   <!-- Codigo Css para darle una mejor apariencia a La pagina
9.     Define los estilos de la interfaz visual
10.    padding es la distancia de un objeto en relación con el marco que lo contiene
11.    margin es la distancia que separa a un objeto de otro
12.    background es el color de fondo
13.    display: block; Estos elementos fluyen hacia abajo
14.    margin: 0 auto; Centra el canvas en la pantalla -->
15. <style>
16.   /* Importa una fuente(tipo de Letra) */
17.   @import url("https://fonts.googleapis.com/css2?family=Poppins&display=swap");
18.
19.   * {
20.     font-family: "Poppins", sans-serif;
21.     padding: 0;
22.     margin: 0;
23.   }
24.
25.   body {
26.     margin: 0 200px 0 200px;
27.   }
28.
29.   header h1 {
30.     font-size: 2em;
31.     text-transform: uppercase;
32.     text-align: center;
33.   }
34.
35.   #miCanvas {
36.     width: 100%;
37.     height: 80vh;
38.     background-color: #dddddd;
39.   }
40. </style>
41. </head>
42.
43. <!-- Comenzamos nuestra pagina -->
44.
45. <body>
46.   <!-- Introducimos un titulo -->
47.   <header>
48.     <h1>
49.       Juego 2D desarrollado en JavaScript puro: Introducción a la Informática
50.     </h1>
51.   </header>
52.
53.   <!-- Introducimos nuestro canvas con un identificador -->
54.   <canvas id="miCanvas"> </canvas>
55.
56.   <!-- Introducimos nuestro codigo JavaScript -->
57.   <script>
58.     /* Llamamos al canvas y le decimos que queremos dibujar en 2d */
59.     var canvas = document.getElementById("miCanvas");
60.     var ctx = canvas.getContext("2d");
61.
62.     //Variables para la bola
63.     var ballRadius = 3; //Tamaño de la bola
64.
65.     var x = canvas.width / 2; // Coloca la bola en la mitad del ancho del CANVAS
66.
67.     var y = canvas.height - 30; // Coloca la bola en la parte inferior del CANVAS
68.
69.     /* DEFINE LOS INCREMENTOS EN X y en Y. EL valor dy es negativo
70.     para que inicialmente el movimiento de la bola sea hacia arriba */
71.     var dx = 2;
72.     var dy = -2;

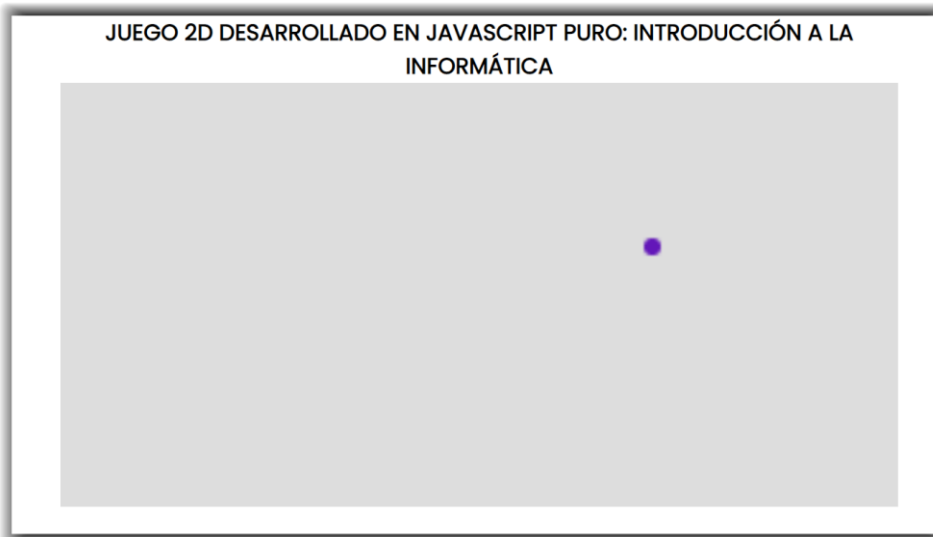
```

```

73.
74.      //Bola 2 el valor dx es negativo para que inicialmente se mueva a la izquierda
75.      var x2 = x;
76.      var y2 = y;
77.      var dx2 = -0.8;
78.      var dy2 = -0.8;
79.
80.      function dibujarBola(cordenaX, cordenaY, tamaño, color) {
81.          // Inicia el dibujo
82.          ctx.beginPath();
83.          ctx.arc(cordenaX, cordenaY, tamaño, 0, Math.PI * 2);
84.          // Color de llenado
85.          ctx.fillStyle = color;
86.          // Se rellena el círculo con el color indicado
87.          ctx.fill();
88.          // Finaliza el dibujo
89.          ctx.closePath();
90.      }
91.
92.      var azar = Math.random() * 100; // Nos dara un numero al azar entre 0 y 99
93.
94.      function dibujar() {
95.          // Limpia el CANVAS
96.          ctx.clearRect(0, 0, canvas.width, canvas.height);
97.
98.          // Dibuja la bola.
99.          dibujarBola(x, y, ballRadius, "#6317B9");
100.
101.          /* Si la variable azar que almacena un numero aleatorio
102.             es mayor a 50 entonces dibujara una segunda bola,
103.             no sabemos si es la primera vez que cargamos la pagina o n veces */
104.          if (azar > 50) {
105.              dibujarBola(x2, y2, ballRadius - 1, "#2196f3");
106.          }
107.
108.          // Se incrementa Los valores para poder hacer el movimiento de las bolas
109.          x += dx;
110.          y += dy;
111.          x2 += dx2;
112.          y2 += dy2;
113.      }
114.
115.      /* EJECUTA LA FUNCIÓN dibujar CADA 10 MILISEGUNDOS
116.         Este es el mecanismo utilizado para construir un sistema que
117.         ejecuta acciones de manera permanente y periódica */
118.      setInterval(dibujar, 20);
119.  </script>
120. </body>
121.
122. </html>

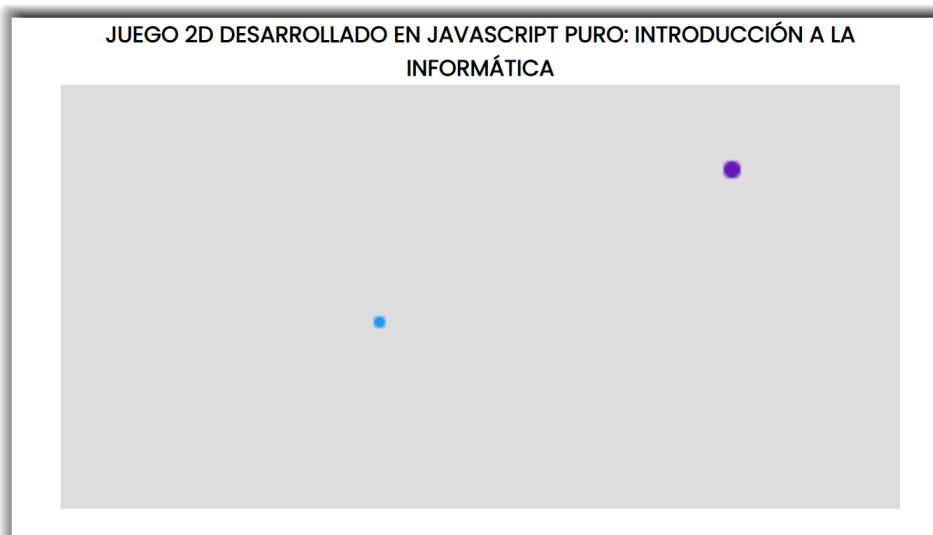
```

Al ejecutar este código se obtiene la siguiente interfaz visual:



Gráfica 2. La interfaz inicial si la condición de dibujar una segunda bola es falsa.

En la gráfica 2 podemos apreciar el dibujo de la bola, y la secuencia de movimiento a partir de los incrementos en X y Y que fueron definidos los cuales corresponden a un movimiento diagonal, solo hay una bola ya que el numero almacenado en la variable azar es menor a 50, lo cual la condición que está en la función dibujar() no se cumple.



Gráfica 3. La interfaz inicial si la condición de dibujar una segunda bola es verdadera.

## 4 FASE 2: REBOTANDO EN LAS PAREDES

El segundo paso consiste en elaborar los límites permitidos a los que la bola puede llegar, en los cuales rebotará y así mantenerse dentro del cuadro asignado.

En este paso se crean dos condicionales por cada bola, ya que al cargar la página nos puede salir una o dos bolas, esto al azar. Las condiciones generan los límites permitidos a los que la bola puede llegar y posteriormente rebotará:

La primera condición es `if(x + dx > canvas.width-ballRadius || x + dx < ballRadius) { dx = -dx;}`, esta condición crea el rango horizontal al que la pelota se puede desplazar. Y el valor de dx pasará a valer negativo para poder que el movimiento se invierta, o sea, si se movía a la derecha, cuando llegue al límite se mueva a la izquierda.

La segunda condición es `if(y + dy > canvas.height-ballRadius || y + dy < ballRadius) {dy = -dy;}`, esta condición crea el rango vertical al que la pelota se puede desplazar, la cual si se movía hacia arriba cambie su movimiento hacia abajo. En esta condición también hay un evento, cada vez que se cumpla esta condición activará este evento y le agregará al elemento CANVAS en HTML una clase (identificador) que será usada para cambiar el color del mismo.

Las condiciones de la segunda bola son iguales, solo que en estas se cambian los valores que corresponden a su posición, ya que no golpea a la misma vez que la primera bola. Además en la condición correspondiente al golpe vertical, no hay un evento, lo cual no cambiará el color del CANVAS.

```

1. <!DOCTYPE html>
2. <html>
3. <head>
4.   <meta charset="UTF-8" />
5.   <title>Juego 2D</title>
6.
7.   <style>
8.     @import url("https://fonts.googleapis.com/css2?family=Poppins&display=swap");
9.     * {
10.      font-family: "Poppins", sans-serif;
11.      padding: 0;
12.      margin: 0;
13.    }
14.    body {
15.      margin: 0 200px 0 200px;
16.    }
17.    h1 {
18.      font-size: 2em;
19.      text-transform: uppercase;
20.      text-align: center;
21.    }
22.    canvas {
23.      width: 100%;
24.      height: 80vh;
25.      background-color: #dddddd;
26.      transition: 0.5s; /* El retardo para cambiar de un estado a otro */
27.    }

```



```

28.      /* Se crea un nuevo estilo el cual sera activado cuando la bola principal toca el limite en la pa
rte superior del canvas */
29.      canvas.activado{
30.          background-color: #222222;
31.          display: block;
32.          margin: 0 auto;
33.      }
34.  </style>
35. </head>
36.
37. <body>
38.     <h1>Juego 2D - Fase 2</h1>
39.
40.     <canvas id="miCanvas"> </canvas>
41.
42.     <script>
43.         var canvas = document.getElementById("miCanvas");
44.         var ctx = canvas.getContext("2d");
45.         var ballRadius = 3;
46.         var x = canvas.width / 2;
47.         var y = canvas.height - 30;
48.         var dx = 2;
49.         var dy = -2;
50.         var x2 = x;
51.         var y2 = y;
52.         var dx2 = -0.8;
53.         var dy2 = -0.8;
54.
55.         function dibujarBola(cordenaX, cordenaY, tamaño, color) {
56.             ctx.beginPath();
57.             ctx.arc(cordenaX, cordenaY, tamaño, 0, Math.PI * 2);
58.             ctx.fillStyle = color;
59.             ctx.fill();
60.             ctx.closePath();
61.         }
62.
63.         var azar = Math.random() * 100;
64.
65.         function dibujar() {
66.             ctx.clearRect(0, 0, canvas.width, canvas.height);
67.             dibujarBola(x, y, ballRadius, "#6317B9");
68.
69.             if (azar > 50) {
70.                 dibujarBola(x2, y2, ballRadius - 1, "#2196f3");
71.             }
72.             //Las condiciones nos permitira saber si la bola a llegado al limete de x o y, si es asi ento
nces cambiara la direccíon, se logra cambiando el signo de las variable dx y dy, tenemos posiblemente dos
bolas las cuales el movimiento de una de ellas es mas lento, por lo cual no golpeará a la misma vez enton
ces utilizamos diferentes condiciones.
73.
74.             //CONDICIONES PARA LA PRIMERA BOLA
75.             //Cuando el valor de x + dx es mayor al ancho del canvas o menor al radio de la bola entonces
se cambia la direccíon horizontal
76.             if(x + dx > canvas.width-ballRadius || x + dx < ballRadius){
77.                 dx = -dx;
78.             }
79.             //Cuando el valor de y + dy es mayor a la altura del canvas o menor al radio de la bola enton
ces se cambia la direccíon vertical.
80.             if(y + dy < ballRadius){
81.                 dy = -dy;
82.             }
83.             //Si llega al limite en Y entonces rebotará y habrá un evento, el cual consiste que al el
emento CANVAS en HTML le agregue o le quite una clase(identificador) la cual permitira en CSS cambiar el
color del CANVAS
84.             var cambia = document.querySelector('canvas')
85.             cambia.classList.toggle('activado')
86.         }
87.         //CONDICIONES PARA LA SEGUNDA BOLA similares a la bola 1 solo que diferentes coordenadas X y Y
88.         if(x2 + dx2 > canvas.width-ballRadius || x2 + dx2 < ballRadius){
89.             dx2 = -dx2;
90.         }
91.         if(y2 + dy2 < ballRadius){
92.             dy2 = -dy2;
93.         }
94.     </script>

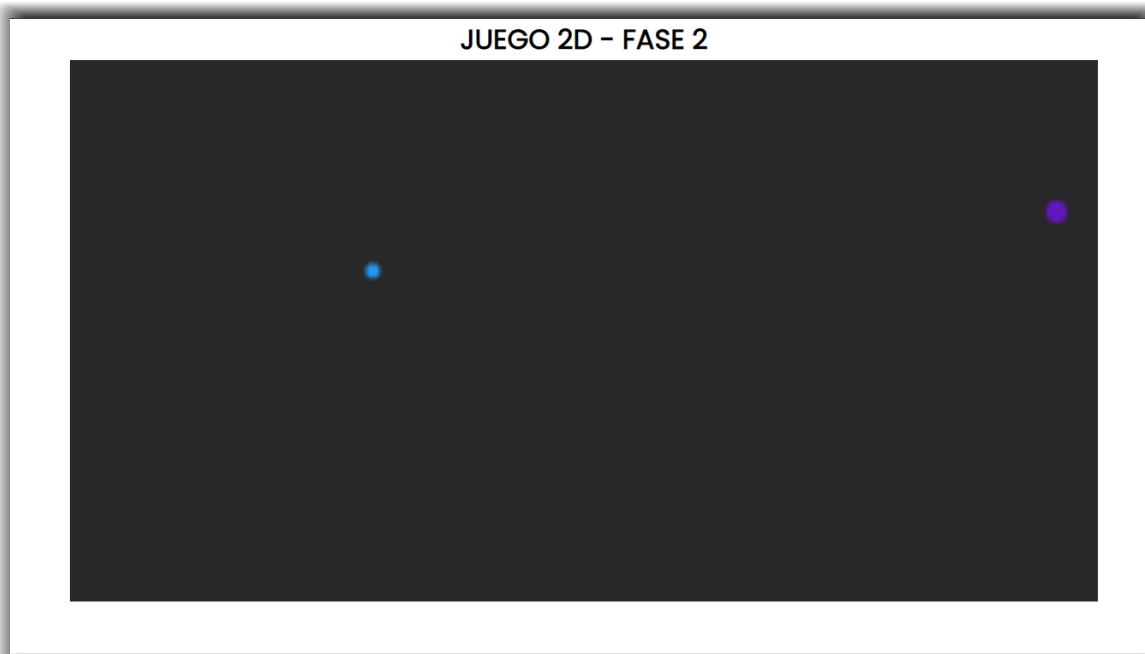
```

```

95.         x += dx;
96.         y += dy;
97.         x2 += dx2;
98.         y2 += dy2;
99.     }
100.
101.     setInterval(dibujar, 20);
102. </script>
103. </body>
104. </html>

```

El resultado será el siguiente:



**Gráfica 4.** Interfaz en donde se cumple la condición de dibujar dos bolas y el rebote de aquellas.

Antes que rebote en Y tenemos una interfaz visual similar a la gráfica 3. En la gráfica 4 podemos observar a la bola rebotando y cumpliendo con los límites anteriormente definidos en las condiciones. Podemos observar dos bolas ya que si recordamos tenemos un número al azar que lo obtenemos luego de cargar la página, y este si es mayor a 50 dibujara la segunda bola que en este caso es la azul, la bola morada es la principal y al rebotar en Y cambiara el color del CANVAS a negro.

En el siguiente apartado se explicará la siguiente fase del juego. En caso de ser necesario, se agregarán todas las explicaciones que sean necesarias para que el juego quede debidamente explicado.



## 5 FASE 3: CONTROL DE LA PALETA CON EL TECLADO

---

El paso número 3, consiste en crear una paleta, esta estará sujeta a unas variables la cuales serán utilizadas para su características, como el taño la anchura y el posicionamiento. Para la paleta tenemos dos variables que almacenan un dato booleano (true, false), esto nos permitirá saber si el jugador a presionado una de las dos teclas que corresponden al movimiento de la paleta, las variables tienen el nombre de flechaDerechaPulsada y flechalzquierdaPulsada, estas dos comienzan con el valor de false ya que no se han pulsado ninguna flecha.

Más adelante tenemos un par de eventos, estos nos permitirá llamar a una función dependiendo de lo que el usuario haga, en este caso pulsar o liberar una tecla, las funciones correspondiente tienen condicionales para saber que tecla se ha pulsado o liberado, y dependiendo de esto las variable que definimos antes cambiaran su valor a true o false.

Tenemos la función correspondiente para dibujar la paleta he igualmente en la función dibujar() se llama a esta, en la función dibujar tenemos un par de condicionales, en donde si alguna de las dos variables flechaDerechaPulsada o flechalzquierdaPulsada tienen el valor true, entonces se encargara de desplazar la paleta.

```
1. <!DOCTYPE html>
2. <html>
3. <head>
4.   <meta charset="UTF-8" />
5.   <title>Juego 2D</title>
6.
7.   <style>
8.     @import url("https://fonts.googleapis.com/css2?family=Poppins&display=swap");
9.     * {
10.       font-family: "Poppins", sans-serif;
11.       padding: 0;
12.       margin: 0;
13.     }
14.     body {
15.       margin: 0 200px 0 200px;
16.     }
17.     h1 {
18.       font-size: 2em;
19.       text-transform: uppercase;
20.       text-align: center;
21.     }
22.     canvas {
23.       width: 100%;
24.       height: 80vh;
25.       background-color: #dddddd;
26.       transition: 0.5s;
27.     }
28.     canvas.activado{
29.       background-color: #222222;
30.       display: block;
31.       margin: 0 auto;
32.     }
33.   </style>
34. </head>
35.
```

```

36. <body>
37.   <h1>Juego 2D - Fase 2</h1>
38.
39.   <canvas id="miCanvas"> </canvas>
40.
41.   <script>
42.     var canvas = document.getElementById("miCanvas");
43.     var ctx = canvas.getContext("2d");
44.     var ballRadius = 3;
45.     var x = canvas.width / 2; //x: columna en la que se encuentra situada la bola
46.     var y = canvas.height - 30; //y: fila en la que se encuentra situada la bola
47.     var dx = 2; //dx: desplazamiento horizontal de la bola
48.     var dy = -2; //dy: desplazamiento vertical de la bola
49.
50.     var x2 = x;
51.     var y2 = y;
52.     var dx2 = -0.8;
53.     var dy2 = -0.8;
54.
55.     /* Las siguiente variables corresponden a las características de la paleta */
56.     var alturaPaleta = 4; // define la altura de la paleta en píxeles
57.     var anchuraPaleta = 40; // define el ancho de la paleta
58.
59.     //Posición donde se encontrará inicialmente nuestra paleta en este caso en el centro del canvas y
    en la base de la pantalla, Para calcular la posición en X de la paleta, se debe tomar el ancho del CANVA
    S, restarle la anchura de la paleta, y el espacio que sobre debe dividirse entre dos. Esto garantiza que or
    iginalmente la paleta estará centrada en la base de la pantalla
60.     var paletaPostX = (canvas.width-anchuraPaleta) / 2;
61.
62.     /* Se define dos variables con tipos booleanos, para saber que tecla se preciona y permitir el mo
    vimiento, estas comienzan con el valor de false ya que al iniciar el juego no se ha precionado ninguna tec
    la, cuando se pulse cualquiera de las dos teclas el estado correspondiente a la tecla cambiara a true, y
    permitira mover la paleta en su respectiva dirección */
63.
64.     var flechaDerechaPulsada = false; // corresponde a la flecha derecha si es pulsada cambiara a true
65.     var flechaIzquierdaPulsada = false; // corresponde a la flecha izquierda si es pulsada cambiara a
    true
66.
67.     /* La instruccion addEventListener, es un mecanismo para dar respuesta a los eventos que se prod
    ucen en el juego, por ejemplo al precionar una de las dos teclas queremos que se mueva.
68.
69.     Este recibe tres parametros.
70.     1) El evento que se va a detectar en este caso precionar una tecla
71.     2) El nombre que le asignamos a la función que responde ante el evento
72.     3) Valor true o false que determina la reacción ante el evento, o sea la propagación de este, cua
    ndo le damos a la tecla el evento sera asignado a toda la pagina, lo cual utilizamos el valor de false, e
    ste valor permite que el flujo del evento comience desde lo inferior hasta lo superior, o sea que al puls
    ar la tecla el evento se le asignara al CANVAS y luego al Body, en cambio si el valor es true el evento s
    e le asignara al Body y luego al CANVAS.
73.     */
74.
75.     // Evento si se pulsa una tecla
76.     document.addEventListener("keydown", teclaPresionada, false);
77.
78.     // Evento si se suelta una tecla
79.     document.addEventListener("keyup", teclaLiberada, false);
80.
81.     // Función que maneja la tecla presionada
82.     function teclaPresionada(e){
83.       if(e.keyCode == 39){
84.         /* e: es el evento que se produce, keyCode permitir saber que código correspondiente a la
            tecla se ha presionado, 39 equivale a la tecla flecha derecha por lo cual el valor de la variable la fl
            echaDerechaPulsada cambia a true*/
85.         flechaDerechaPulsada = true;
86.       }
87.       else if(e.keyCode == 37){
88.         /* e: es el evento que se produce, keyCode permitir saber que código correspondiente a la
            tecla se ha presionado, 37 equivale a la tecla flecha izquierda por lo cual el valor de la variable la
            flechaIzquierdaPulsada cambia a true*/
89.         flechaIzquierdaPulsada = true;
90.       }
91.     }
92.
93.     // Función que maneja la tecla liberada

```

```

94.     function teclaLiberada(e){
95.         if(e.keyCode == 39){
96.             /* Si la tecla liberada es La 39, se ha dejado de precionar La flecha derecha, La variabl
e cambia a false*/
97.             flechaDerechaPulsada = false;
98.         }
99.         else if(e.keyCode == 37){
100.            /* Si la tecla liberada es La 37, se ha dejado de precionar La flecha izquierda, La varia
ble cambia a false*/
101.            flechaIzquierdaPulsada = false;
102.        }
103.    }
104.
105.    function dibujarBola(cordernaX, cordenaY, tamaño, color) {
106.        ctx.beginPath();
107.        ctx.arc(cordernaX, cordenaY, tamaño, 0, Math.PI * 2);
108.        ctx.fillStyle = color;
109.        ctx.fill();
110.        ctx.closePath();
111.    }
112.
113.    //Función para dibujar la paleta
114.    function dibujarPaleta(){
115.
116.        ctx.beginPath(); // Se incia el dibujo
117.
118.        //Se crea un rectangulo utilizando Los valores de Las variable correspondientes a La paleta, u
tilizamos La posición de x, y Y menos La altura de La paleta parasituarla en La base del CANVAS ,La altur
a, anchura
119.        ctx.rect(paletaPostX, canvas.height-alturaPaleta, anchuraPaleta, alturaPaleta);
120.
121.        ctx.fillStyle = "#f5d109" // Color de ala paleta
122.        ctx.fill(); // Se rellena la paleta
123.        ctx.closePath(); // Finaliza el dibujo
124.    }
125.
126.    var azar = Math.random() * 100;
127.
128.    // Función principal para generar Los procesos
129.    function dibujar() {
130.
131.        ctx.clearRect(0, 0, canvas.width, canvas.height);
132.
133.        dibujarBola(x, y, ballRadius, "#6317B9"); // dibuja La bola
134.
135.        dibujarPaleta(); // Dibuja La paleta
136.
137.        // Dibuja una segunda bola
138.        if (azar > 50) {
139.            dibujarBola(x2, y2, ballRadius - 1, "#2196f3");
140.        }
141.
142.        // Rebote de La bola
143.        if(x + dx > canvas.width-ballRadius || x + dx < ballRadius){
144.            dx = -dx;
145.        }
146.
147.        if(y + dy < ballRadius){
148.            dy = -dy;
149.
150.            var cambia = document.querySelector('canvas')
151.            cambia.classList.toggle('activado')
152.        }
153.
154.        if(x2 + dx2 > canvas.width-ballRadius || x2 + dx2 < ballRadius){
155.            dx2 = -dx2;
156.        }
157.        if(y2 + dy2 < ballRadius){
158.            dy2 = -dy2;
159.        }
160.
161.        /* Movimiento de La paleta
162.        si se pulsa La flecha derecha y aun no ha Llegado al borde de La anchura del CANVAS se puede
cambiar su posición, La función dibujarPaleta() redibuja La paleta de manera ciclica en La nueva posición
*/

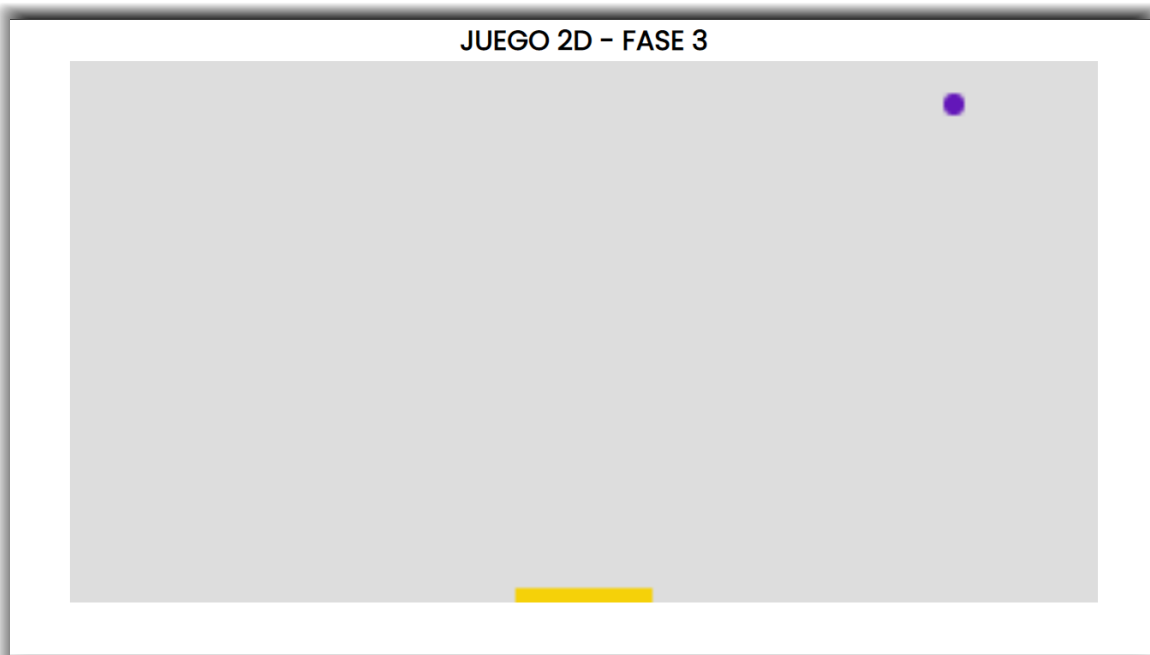
```

```

163.         if(flechaDerechaPulsada && paletaPostX < canvas.width-anchuraPaleta){
164.             // Desplaza la paleta hacia la derecha se logra sumando la poscion actual con la nueva
165.             paletaPostX += 7;
166.         }
167.         else if(flechaIzquierdaPulsada && paletaPostX > 0){
168.             // Desplaza la paleta hacia la izquierda se logra restando la poscion actual con la nueva
169.             paletaPostX -= 7;
170.         }
171.
172.         // Se incrementan los valores de la pocion de las bolas
173.         x += dx;
174.         y += dy;
175.         x2 += dx2;
176.         y2 += dy2;
177.     }
178.
179.     // Redibuja, actualiza las posiciones en el juego cada 25 milisegundos,
180.     setInterval(dibujar, 25);
181. </script>
182. </body>
183. </html>

```

Al ejecutar el anterior código se obtiene la siguiente interfaz visual:



Gráfica 5. Interfaz con la correspondiente paleta dibujada.

En el siguiente apartado se explicara la siguiente fase del juego.

## 6 FASE 4: FIN DE PARTES ESENCIALES DEL JUEGO

En esta fase del programa, vamos a programar el rebote de las bolas en la paleta ya que si recordamos nos puede salir una o dos bolas esto al azar, lo cual debemos evaluar cuando estas lleguen a la base del CANVAS si está en posición de la paleta o si toca directamente la base.

Para lo anterior debemos poner más condicionales en la función dibujar, con base a la posición de la bola cuando llegue a la base ( $y + dy > \text{canvas.height} - \text{ballRadius}$ ), evaluaremos si esta toca la parte superior de la paleta ( $x > \text{paletaPosX} \ \&\& \ x < \text{paletaPosX} + \text{anchuraPaleta}$ ), si es así, se cambiara el valor de  $dy$  que corresponde al movimiento de la bola, de lo contrario el juego se detendrá, se mostrara un mensaje diciendo “Game Over” y se cargara la página nuevamente. Estas condicionales se definen dos veces ya que corresponden a la primera y segunda bola.

```

1. <!DOCTYPE html>
2. <html>
3. <head>
4.   <meta charset="UTF-8" />
5.   <title>Juego 2D</title>
6.
7.   <style>
8.     @import url("https://fonts.googleapis.com/css2?family=Poppins&display=swap");
9.
10.    * { font-family: "Poppins", sans-serif; padding: 0; margin: 0; }
11.    body { margin: 0 200px 0 200px; }
12.    h1 { font-size: 2em; text-transform: uppercase; text-align: center; }
13.    canvas { width: 100%; height: 80vh; background-color: #dddddd; transition: 0.5s; }
14.    canvas.activado { background-color: #222222; display: block; margin: 0 auto; }
15.  </style>
16. </head>
17. <body>
18.   <h1>Juego 2D - Fase 4</h1>
19.   <canvas id="miCanvas"> </canvas>
20.
21.   <script>
22.     var canvas = document.getElementById("miCanvas");
23.     var ctx = canvas.getContext("2d");
24.     var ballRadius = 3;
25.     var x = canvas.width / 2;
26.     var y = canvas.height - 30;
27.     var dx = 2;
28.     var dy = -2;
29.
30.     var x2 = x;
31.     var y2 = y;
32.     var dx2 = -0.8;
33.     var dy2 = -0.8;
34.
35.     // Paleta
36.     var alturaPaleta = 4;
37.     var anchuraPaleta = 40;
38.     var paletaPosX = (canvas.width - anchuraPaleta) / 2;
39.     var flechaDerechaPulsada = false;
40.     var flechaIzquierdaPulsada = false;
41.
42.     document.addEventListener("keydown", teclaPresionada, false);
43.     document.addEventListener("keyup", teclaLiberada, false);

```

```

44.
45.     function teclaPresionada(e) {
46.         if (e.keyCode == 39) {
47.             flechaDerechaPulsada = true;
48.         } else if (e.keyCode == 37) {
49.             flechaIzquierdaPulsada = true;
50.         }
51.     }
52.     function teclaLiberada(e) {
53.         if (e.keyCode == 39) {
54.             flechaDerechaPulsada = false;
55.         } else if (e.keyCode == 37) {
56.             flechaIzquierdaPulsada = false;
57.         }
58.     }
59.
60.     function dibujarBola(cordenaX, cordenaY, tamaño, color) {
61.         ctx.beginPath();
62.         ctx.arc(cordenaX, cordenaY, tamaño, 0, Math.PI * 2);
63.         ctx.fillStyle = color;
64.         ctx.fill();
65.         ctx.closePath();
66.     }
67.
68.     function dibujarPaleta() {
69.         ctx.beginPath();
70.         ctx.rect(paletaPosX, canvas.height-alturaPaleta, anchuraPaleta, alturaPaleta);
71.         ctx.fillStyle = "#f5d109";
72.         ctx.fill();
73.         ctx.closePath();
74.     }
75.
76.     var azar = Math.random()*100;
77.
78.     // Función principal para generar los procesos
79.     function dibujar() {
80.         ctx.clearRect(0, 0, canvas.width, canvas.height);
81.
82.         dibujarBola(x, y, ballRadius, "#6317B9"); // dibuja la bola
83.
84.         // Dibuja una segunda bola
85.         if (azar > 50) {
86.             dibujarBola(x2, y2, ballRadius - 1, "#2196f3");
87.         }
88.
89.         dibujarPaleta(); // Dibuja la paleta
90.
91.         // Rebote de la bola
92.         if (x + dx > canvas.width-ballRadius || x + dx < ballRadius) {
93.             dx = -dx;
94.         }
95.         if (y + dy < ballRadius) {
96.             dy = -dy;
97.
98.             var cambia = document.querySelector("canvas");
99.             cambia.classList.toggle("activado");
100.        }
101.
102.        if (x2 + dx2 > canvas.width-ballRadius || x2 + dx2 < ballRadius) {
103.            dx2 = -dx2;
104.        }
105.        if (y2 + dy2 < ballRadius) {
106.            dy2 = -dy2;
107.        }
108.
109.        // Si la bola alcanza la base del CANVAS, se debe evaluar si la posición de la bola está dentro
        // o de la paleta, si esta entonces se incrementa y si no es así, cuando la bola llegue a la base de la pantalla
        // se detiene el ciclo del juego y se da a entender por medio de un mensaje con la función alert que
        // el juego ha terminado y posteriormente se recarga la página. Se definen dos veces estas condiciones ya que
        // se nos puede salir dos bolas y estas no tocan la paleta a la misma vez.
110.
111.        // Rebotar en Paleta o Game Over
112.        if (y + dy > canvas.height-ballRadius) {
113.            if (x > paletaPosX && x < paletaPosX + anchuraPaleta) {
114.                dy = -dy;

```

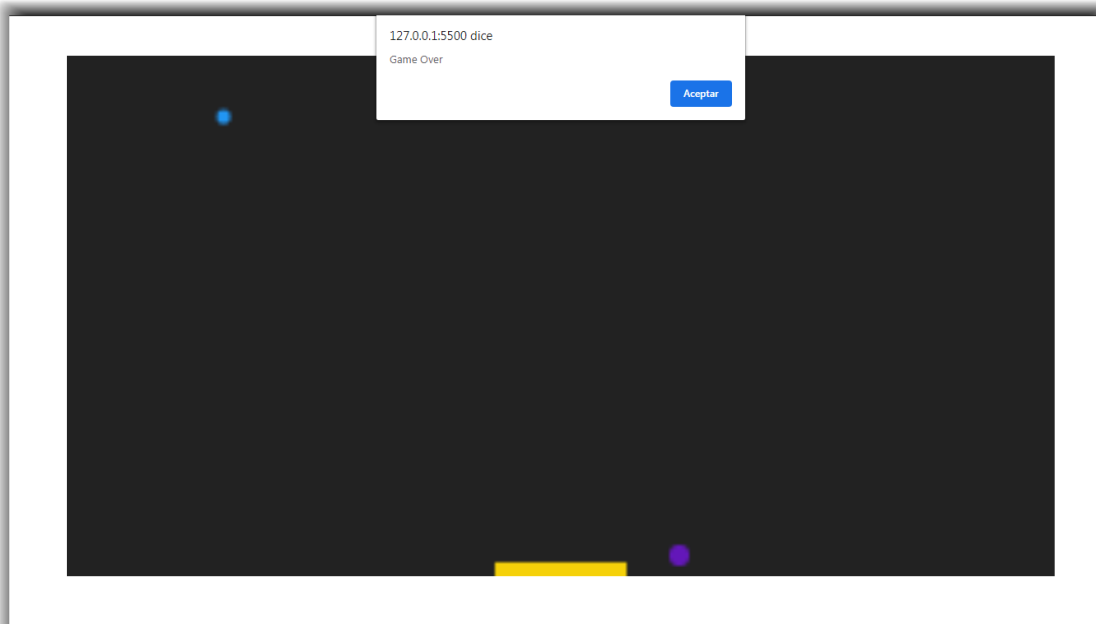


```

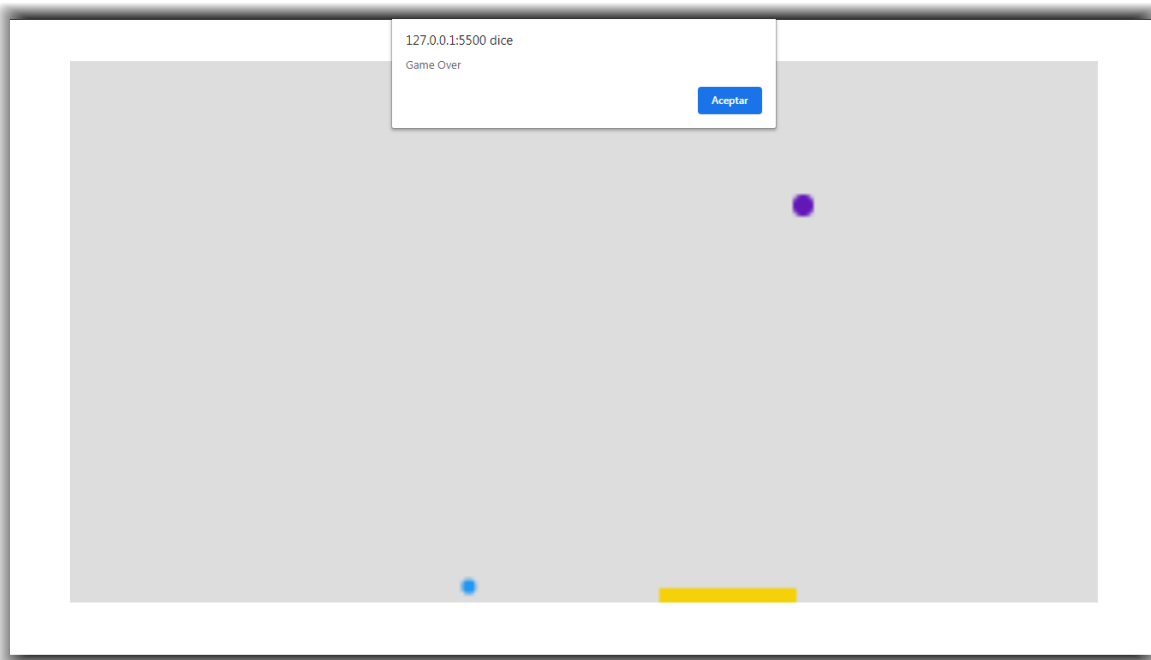
115.         } else {
116.             clearInterval(juego);
117.             alert("Game Over");
118.             document.location.reload();
119.         }
120.     }
121.     // Por si hay una segunda bola entonces esta también debe rebotar en la paleta o si no perder
122.     if (azar > 50) {
123.         if (y2 + dy2 > canvas.height - ballRadius) {
124.             if (x2 > paletaPosX && x2 < paletaPosX + anchuraPaleta) {
125.                 dy2 = -dy2;
126.             } else {
127.                 clearInterval(juego);
128.                 alert("Game Over");
129.                 document.location.reload();
130.             }
131.         }
132.     }
133.
134.     // Movimiento Paleta
135.     if (flechaDerechaPulsada && paletaPosX < canvas.width - anchuraPaleta) {
136.         paletaPosX += 7;
137.     } else if (flechaIzquierdaPulsada && paletaPosX > 0) {
138.         paletaPosX -= 7;
139.     }
140.     // Se incrementan los valores de la posición de las bolas
141.     x += dx;
142.     y += dy;
143.     x2 += dx2;
144.     y2 += dy2;
145. }
146. /* Se asigna una variable al proceso del ciclo, si se quiere eliminar el ciclo se utiliza esta va
    riante */
147. var juego = setInterval(dibujar, 25);
148. </script>
149. </body>
150. </html>

```

El resultado del anterior código es:



Gráfica 6. Interfaz cuando la bola llega al a la base del Canvas.



**Gráfica 7. Interfaz cuando la segunda bola llega al a la base del Canvas.**

En la gráfica 6, vemos que la bola morada ha llegado a la base del canvas y no toca la paleta por lo cual se ejecuta el código para finalizar el juego, mostrar un mensaje y recargar la página. No importa si la segunda bola está en la parte superior o si es viceversa, esto lo podemos ver en la gráfica 7.

En el siguiente apartado se explicara la siguiente fase del juego.

## 7 FASE 5: MURO DE LADRILLOS

En esta parte del juego crearemos los ladrillos, los cuales estarán sujetos a unas propiedades definidas gracias a variables, utilizaremos bucles anidados para dibujar los ladrillos tanto vertical como horizontalmente, también listas o comúnmente llamadas array para almacenarlos.

Las variables a utilizar para las características de estos son: el ancho de cada ladrillo, altura, espacio entre ellos, número de filas y columnas, estas dos últimas variables no serán constantes cada vez que se carga la página, ya que gracias al método `Math.random()`, anteriormente utilizado para dibujar una segunda bola o no, esta vez lo utilizamos para obtener la cantidad de filas y columnas que se dibujaran, esto obviamente al azar.

Creamos una nueva función llamada `dibujarLadrillos()`, esta nos servirá para dibujar los ladrillos, también tiene dos bucles para recorrer las columnas y filas, dentro tenemos variables las cuales nos servirá para el posicionamiento de cada uno, tanto en el eje X como en el Y, utilizamos variables anteriormente creadas altura, ancho etc. También un par de condicionales para pintar los ladrillos dependiendo si una variable anteriormente inicializada `colorLadrillo` es true o false, pero también, dependiendo si el número donde se encuentra el ladrillo en el array es par o impar, el valor de `colorLadrillo` cambiara si la bola principal (bola morada), rebota en X o Y.

```

1. <!DOCTYPE html>
2. <html>
3.
4. <head>
5.   <meta charset="UTF-8" />
6.   <title>Juego 2D</title>
7.
8.   <style>
9.     @import url("https://fonts.googleapis.com/css2?family=Poppins&display=swap");
10.
11.     * {font-family: "Poppins", sans-serif; padding: 0; margin: 0; }
12.     body {margin: 0 200px 0 200px;}
13.     h1 {font-size: 2em; text-transform: uppercase; text-align: center;}
14.     canvas {width: 100%; height: 80vh; background-color: #dddddd;transition: 0.5s;}
15.     canvas.activado {background-color: #222222; display: block;margin: 0 auto;}
16.   </style>
17. </head>
18.
19. <body>
20.   <h1>Juego 2D - Fase 5</h1>
21.   <canvas id="miCanvas"> </canvas>
22.
23.   <script>
24.     var canvas = document.getElementById("miCanvas");
25.     var ctx = canvas.getContext("2d");
26.     var ballRadius = 3;
27.     var x = canvas.width / 2;
28.     var y = canvas.height - 30;
29.     var dx = 2;
30.     var dy = -2;
31.

```

```

32.     var x2 = x;
33.     var y2 = y;
34.     var dx2 = -0.8;
35.     var dy2 = -0.8;
36.
37.     // Paleta
38.     var alturaPaleta = 4;
39.     var anchuraPaleta = 40;
40.     var paletaPosX = (canvas.width - anchuraPaleta) / 2;
41.     var flechaDerechaPulsada = false;
42.     var flechaIzquierdaPulsada = false;
43.
44.     // VARIABLES ASOCIADAS A LOS LADRILLOS
45.     var numFilasLadrillos = 0;
46.     var numColumnasLadrillos = 0;
47.
48.     // Se crea un bucle para poner un numero al zar de ladrillos y columnas
49.     do {
50.         numFilasLadrillos = Math.round(Math.random() * 100);
51.         numColumnasLadrillos = Math.round(Math.random() * 10);
52.     } while (numFilasLadrillos < 1 || numFilasLadrillos > 19 || numColumnasLadrillos < 1 || numColumnasLadrillos > 6);
53.
54.     var anchoLadrillo = 12;
55.     var alturaLadrillo = 5;
56.     var rellenoLadrillos = 2;
57.     var vacioSuperiorLadrillos = 10;
58.     var vacioIzquierdoLadrillos = 15;
59.
60.     // Crea un conjunto de ladrillos, Inicialmente vacio
61.     var ladrillos = [];
62.
63.     // Se recorren el numero de columnas
64.     for (var columna = 0; columna < numColumnasLadrillos; columna++) {
65.         // Se define la primera columna. es una lista vertical
66.         ladrillos[columna] = [];
67.
68.         // Cada columna, recorre el numero de filas
69.         for (var fila = 0; fila < numFilasLadrillos; fila++) {
70.             // para cada columna y fila se define un ladrillo
71.
72.             // A cada ladrillo se le asignan la posicon de la cordena X y Y estos valores valen 0 est
73.             // o cambia cuando se dibujan en la funcion dibujarladrillos()
74.             ladrillos[columna][fila] = { x: 0, y: 0 };
75.         }
76.     }
77.
78.     document.addEventListener("keydown", teclaPresionada, false);
79.     document.addEventListener("keyup", teclaLiberada, false);
80.
81.     function teclaPresionada(e) {
82.         if (e.keyCode == 39) {
83.             flechaDerechaPulsada = true;
84.         } else if (e.keyCode == 37) {
85.             flechaIzquierdaPulsada = true;
86.         }
87.     }
88.
89.     function teclaLiberada(e) {
90.         if (e.keyCode == 39) {
91.             flechaDerechaPulsada = false;
92.         } else if (e.keyCode == 37) {
93.             flechaIzquierdaPulsada = false;
94.         }
95.     }
96.
97.     function dibujarBola(cordenaX, cordenaY, tamaño, color) {
98.         ctx.beginPath();
99.         ctx.arc(cordenaX, cordenaY, tamaño, 0, Math.PI * 2);
100.        ctx.fillStyle = color;
101.        ctx.fill();
102.        ctx.closePath();
103.    }
104.
105.    function dibujarPaleta() {
106.        ctx.beginPath();

```

```

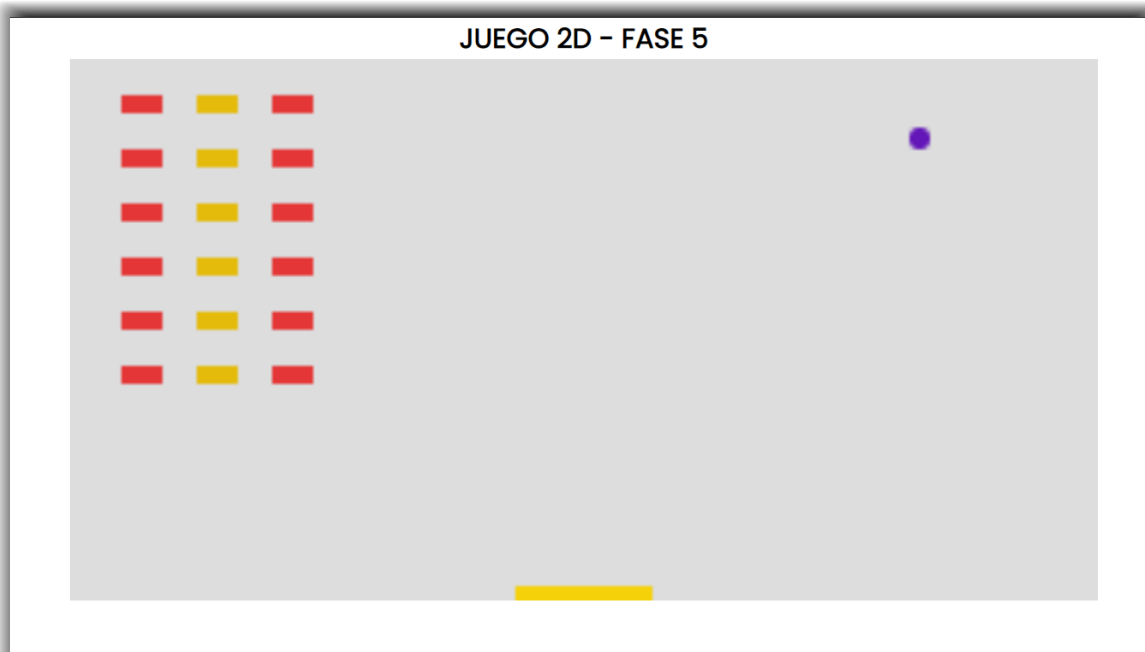
105.         ctx.rect(
106.             paletaPosX,
107.             canvas.height - alturaPaleta,
108.             anchuraPaleta,
109.             alturaPaleta
110.         );
111.         ctx.fillStyle = "#f5d109";
112.         ctx.fill();
113.         ctx.closePath();
114.     }
115.
116.     var colorLadrillos = false; // La utilizaremos para cambiar el color de Los Ladrillos
117.
118.     // DIBUJAR LADRILLOS
119.     function dibujarLadrillos() {
120.         // Recorrer todas Las columnas
121.         for (var columna = 0; columna < numColumnasLadrillos; columna++) {
122.             // Para cada columna, recorre sus filas
123.             for (var fila = 0; fila < numFilasLadrillos; fila++) {
124.                 // Para calcular la posición X que le corresponde a cada ladrillo, lo calculamos segun
125.                 // el ancho de ladrillo, el relleno, el espacio hacia a la izquierda
126.                 var brickX =
127.                     fila * (anchoLadrillo + rellenoLadrillos) +
128.                     vacioIzquierdoLadrillos;
129.
130.                 // Se repite pero para la posición en Y
131.                 var brickY =
132.                     columna * (alturaLadrillo + rellenoLadrillos) +
133.                     vacioSuperiorLadrillos;
134.
135.                 // Se le asignan los valores anteriores a cada ladrillo
136.                 ladrillos[columna][fila].x = brickX;
137.                 ladrillos[columna][fila].y = brickY;
138.
139.                 // Dibuja el ladrillo con los valores asociados
140.                 // Coordenada: (brickX, brickY)
141.                 // Anchura: anchoLadrillo
142.                 // Altura: alturaLadrillo
143.
144.                 ctx.beginPath();
145.                 ctx.rect(brickX, brickY, anchoLadrillo, alturaLadrillo);
146.
147.                 // Dependiendo de la variable colorLadrillos si es true los ladrillos tendrán un color
148.                 // diferente si es false, pero también si el número correspondiente al ladrillo es par o impar cambiará de
149.                 // color
150.
151.                 if (colorLadrillos) {
152.                     if (fila % 2) {
153.                         ctx.fillStyle = "#e61c1cdc";
154.                     } else {
155.                         ctx.fillStyle = "#e4bb06fa";
156.                     }
157.                 } else {
158.                     if (fila % 2) {
159.                         ctx.fillStyle = "#e4bb06fa";
160.                     } else {
161.                         ctx.fillStyle = "#e61c1cdc";
162.                     }
163.                 }
164.
165.                 ctx.fill();
166.                 ctx.closePath();
167.
168.                 // Se recorre todo el ciclo se dibuja todos los ladrillos
169.             }
170.         }
171.     }
172.
173.     var azar = Math.random() * 100;
174.
175.     // Función principal para generar los procesos
176.     function dibujar() {
177.         ctx.clearRect(0, 0, canvas.width, canvas.height);
178.
179.         dibujarBola(x, y, ballRadius, "#6317B9"); // dibuja la bola

```

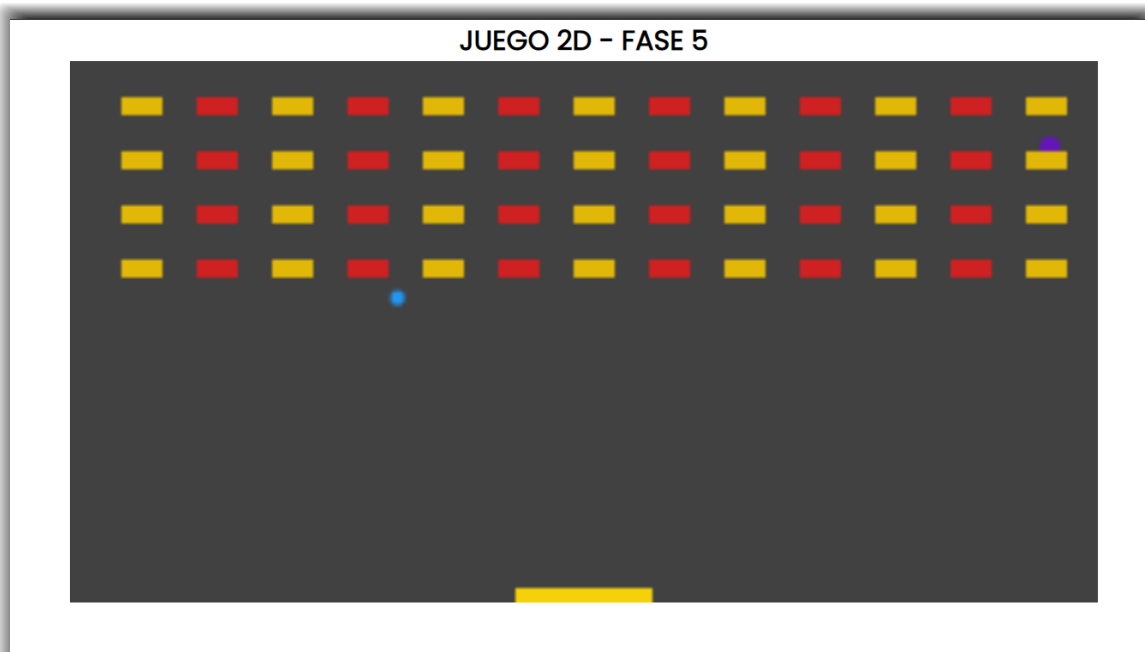
```

177.
178.      // Dibuja una segunda bola
179.      if (azar > 50) {
180.          dibujarBola(x2, y2, ballRadius - 1, "#2196f3");
181.      }
182.
183.      dibujarPaleta(); // Dibuja la paleta
184.
185.      dibujarLadrillos(); // Dibuja el conjunto de los ladrillos
186.
187.      // Rebote de la bola
188.      if (x + dx > canvas.width - ballRadius || x + dx < ballRadius) {
189.          dx = -dx;
190.
191.          // Cambia el color de los ladrillos si la bola principal rebota en x
192.          colorLadrillos = false;
193.      }
194.      if (y + dy < ballRadius) {
195.          dy = -dy;
196.
197.          var cambia = document.querySelector("canvas");
198.          cambia.classList.toggle("activado");
199.
200.          // Cambiara el valor y color de los ladrillos si rebota la bola principal en Y
201.          colorLadrillos = true;
202.      }
203.
204.      if (x2 + dx2 > canvas.width - ballRadius || x2 + dx2 < ballRadius) {
205.          dx2 = -dx2;
206.      }
207.      if (y2 + dy2 < ballRadius) {
208.          dy2 = -dy2;
209.      }
210.
211.      // Rebotar en Paleta o Game Over
212.      if (y + dy > canvas.height - ballRadius) {
213.          if (x > paletaPosX && x < paletaPosX + anchuraPaleta) {
214.              dy = -dy;
215.          } else {
216.              clearInterval(juego);
217.              alert("Game Over");
218.              document.location.reload();
219.          }
220.      }
221.      if (azar > 50) {
222.          if (y2 + dy2 > canvas.height - ballRadius) {
223.              if (x2 > paletaPosX && x2 < paletaPosX + anchuraPaleta) {
224.                  dy2 = -dy2;
225.              } else {
226.                  clearInterval(juego);
227.                  alert("Game Over");
228.                  document.location.reload();
229.              }
230.          }
231.      }
232.
233.      // Movimiento Paleta
234.      if (flechaDerechaPulsada && paletaPosX < canvas.width - anchuraPaleta) {
235.          paletaPosX += 7;
236.      } else if (flechaIzquierdaPulsada && paletaPosX > 0) {
237.          paletaPosX -= 7;
238.      }
239.      // Se incrementan los valores de la poscion de las bolas
240.      x += dx;
241.      y += dy;
242.      x2 += dx2;
243.      y2 += dy2;
244.  }
245.  /* Se asigna una variable al proceso del ciclo, si se quiere eliminar el ciclo se utiliza esta va
riable */
246.  var juego = setInterval(dibujar, 25);
247.  </script>
248. </body>
249. </html>

```



Gráfica 8. Interfaz con los ladrillos dibujados antes de que la bola rebote en cualquier eje.



Gráfica 9. Interfaz cuando la bola morada rebota en el eje Y.

En las anteriores gráficas vimos que el número de ladrillos es al azar nos puede entre uno hasta 114 ladrillo, y el cambio de color de estos cuando la bola morada toca el eje Y.

## 8 FASE 6: DETECCIÓN DE COLISIONES

En la fase 6, crearemos la función que nos permita detectar la colisión entre la bola con los ladrillos, al ocurrir eso el ladrillo desaparecerá y la bola cambiara su dirección.

Procederemos a analizar la función que hace esto posible, la función será la siguiente: function deteccionColision(), cuando creamos los ladrillos se le asigna un status, este nos permitirá saber si el ladrillo es visible o no, en deteccionColision() recorreremos las filas y columnas de los ladrillos para verificar si el estatus vale 1, si es así procedemos a calcular la posición de la bola con respecto al ladrillos, y si la bola toca algún borde superior o inferior del ladrillo esta cambiar su dirección y el status de cuyo ladrillo valdrá 0, lo cual no se dibujara. Lo hacemos dos veces ya que si recordamos nos puede salir dos bolas.

```

1.  <!DOCTYPE html>
2.  <html>
3.
4.  <head>
5.    <meta charset="UTF-8" />
6.    <title>Juego 2D</title>
7.
8.    <style>
9.      @import url("https://fonts.googleapis.com/css2?family=Poppins&display=swap");
10.
11.      * {font-family: "Poppins", sans-serif; padding: 0; margin: 0; }
12.      body {margin: 0 200px 0 200px;}
13.      h1 {font-size: 2em; text-transform: uppercase; text-align: center;}
14.      canvas {width: 100%; height: 80vh; background-color: #dddddd;transition: 0.5s;}
15.      canvas.activado {background-color: #222222; display: block;margin: 0 auto;}
16.    </style>
17.  </head>
18.
19.  <body>
20.    <h1>Juego 2D - Fase 6</h1>
21.    <canvas id="miCanvas"> </canvas>
22.
23.    <script>
24.      var canvas = document.getElementById("miCanvas");
25.      var ctx = canvas.getContext("2d");
26.      var ballRadius = 3;
27.      var x = canvas.width / 2;
28.      var y = canvas.height - 30;
29.      var dx = 2;
30.      var dy = -2;
31.
32.      var x2 = x;
33.      var y2 = y;
34.      var dx2 = -0.8;
35.      var dy2 = -0.8;
36.
37.      // Paleta
38.      var alturaPaleta = 4;
39.      var anchuraPaleta = 40;
40.      var paletaPosX = (canvas.width - anchuraPaleta) / 2;
41.      var flechaDerechaPulsada = false;
42.      var flechaIzquierdaPulsada = false;
43.
44.      // VARIABLES ASOCIADAS A LOS LADRILLOS
45.      var numFilasLadrillos = 0;
46.      var numColumnasLadrillos = 0;
47.      // Se crea un bucle para poner un numero al zar de ladrillos y columnas
48.      do {
49.        numFilasLadrillos = Math.round(Math.random() * 100);

```



```

50.         numColumnasLadrillos = Math.round(Math.random() * 10);
51.     } while (numFilasLadrillos < 2 || numFilasLadrillos > 13 || numColumnasLadrillos < 2 || numColumna
sLadrillos > 6);
52.
53.     var anchoLadrillo = 12;
54.     var alturaLadrillo = 5;
55.     var rellenoLadrillos = 5;
56.     var vacioSuperiorLadrillos = 10;
57.     var vacioIzquierdoLadrillos = 10;
58.
59.     var ladrillos = [];
60.
61.     for (var columna = 0; columna < numColumnasLadrillos; columna++) {
62.         ladrillos[columna] = [];
63.
64.         for (var fila = 0; fila < numFilasLadrillos; fila++) {
65.             // Status definira si el ladrillo es visible o no en 1 sera visible
66.             ladrillos[columna][fila] = { x: 0, y: 0, status: 1 };
67.         }
68.     }
69.
70.     document.addEventListener("keydown", teclaPresionada, false);
71.     document.addEventListener("keyup", teclaLiberada, false);
72.
73.     function teclaPresionada(e) {
74.         if (e.keyCode == 39) {
75.             flechaDerechaPulsada = true;
76.         } else if (e.keyCode == 37) {
77.             flechaIzquierdaPulsada = true;
78.         }
79.     }
80.     function teclaLiberada(e) {
81.         if (e.keyCode == 39) {
82.             flechaDerechaPulsada = false;
83.         } else if (e.keyCode == 37) {
84.             flechaIzquierdaPulsada = false;
85.         }
86.     }
87.
88.     function dibujarBola(cordenaX, cordenaY, tamaño, color) {
89.         ctx.beginPath();
90.         ctx.arc(cordenaX, cordenaY, tamaño, 0, Math.PI * 2);
91.         ctx.fillStyle = color;
92.         ctx.fill();
93.         ctx.closePath();
94.     }
95.
96.     function dibujarPaleta() {
97.         ctx.beginPath();
98.         ctx.rect(paletaPosX, canvas.height - alturaPaleta, anchuraPaleta, alturaPaleta);
99.         ctx.fillStyle = "#f5d109";
100.        ctx.fill();
101.        ctx.closePath();
102.    }
103.
104.    var colorLadrillos = false; // La utilizaremos para cambiar el color de Los Ladrillos
105.
106.    function dibujarLadrillos() {
107.        // Recorrer todas las columnas
108.        for (var columna = 0; columna < numColumnasLadrillos; columna++) {
109.            for (var fila = 0; fila < numFilasLadrillos; fila++) {
110.                var brickX = fila * (anchoLadrillo + rellenoLadrillos) + vacioIzquierdoLadrillos;
111.
112.                var brickY = columna * (alturaLadrillo + rellenoLadrillos) + vacioSuperiorLadrillos;
113.
114.                ladrillos[columna][fila].x = brickX;
115.                ladrillos[columna][fila].y = brickY;
116.
117.                // Se dibujan los ladrillos cuyo status vale 1
118.
119.                if(ladrillos[columna][fila].status == 1){
120.                    ctx.beginPath();
121.                    ctx.rect(brickX, brickY, anchoLadrillo, alturaLadrillo);
122.
123.                    if (colorLadrillos) {

```

```

124.         if (fila % 2) {
125.             ctx.fillStyle = "#e61c1cdc";
126.         } else {
127.             ctx.fillStyle = "#e4bb06fa";
128.         }
129.     } else {
130.         if (fila % 2) {
131.             ctx.fillStyle = "#e4bb06fa";
132.         } else {
133.             ctx.fillStyle = "#e61c1cdc";
134.         }
135.     }
136. }
137. ctx.fill();
138. ctx.closePath();
139. }
140. }
141. }
142.
143. // Detectar La colision
144. function deteccionColision(){
145.     // Los ciclos recorre todos Los Ladrillos
146.     for(var c = 0; c < numColumnasLadrillos; c++){
147.         for(var f = 0; f < numFilasLadrillos; f++){
148.             // Se define una variable para guardar el Ladrillo en donde nos encontramos
149.             var b = ladrillos[c][f];
150.
151.             // Si el Ladrillo es visible, se debe verificar si entra en contacto con las bolas
152.             if(b.status == 1){
153.                 // si hay contacto entre el ladrillo y las bolas entonces se modifica las cordena
s y el valor de status valdra (0) para que no se muestre
154.
155.                 if(x > b.x && x < b.x + anchoLadrillo && y > b.y && y < b.y + alturaLadrillo){
156.                     dy = -dy;
157.                     b.status = 0;
158.                 }
159.                 // Debemos tener en cuenta la segunda bola
160.                 if(azarSegundaBola > 50){
161.                     if(x2 > b.x && x2 < b.x + anchoLadrillo && y2 > b.y && y2 < b.y + alturaLadri
llo){
162.                         dy2 = -dy2;
163.                         b.status = 0;
164.                     }
165.                 }
166.             }
167.         }
168.     }
169. }
170.
171. var azarSegundaBola = Math.random() * 100;
172.
173. // Función principal para generar los procesos
174. function dibujar() {
175.     ctx.clearRect(0, 0, canvas.width, canvas.height);
176.
177.     dibujarBola(x, y, ballRadius, "#6317B9"); // dibuja la bola
178.
179.     // Dibuja una segunda bola
180.     if (azarSegundaBola > 50) {
181.         dibujarBola(x2, y2, ballRadius - 1, "#2196f3");
182.     }
183.
184.     dibujarPaleta(); // Dibuja la paleta
185.
186.     dibujarLadrillos(); // Dibuja el conjunto de los ladrillos
187.
188.     // SE LLAMA A LA FUNCION DETECTAR COLISION
189.     deteccionColision();
190.
191.     // Rebote de la bola
192.     if (x + dx > canvas.width - ballRadius || x + dx < ballRadius) {
193.         dx = -dx;
194.
195.         // Cambia el color de los ladrillos si la bola principal rebota en x
196.         colorLadrillos = false;

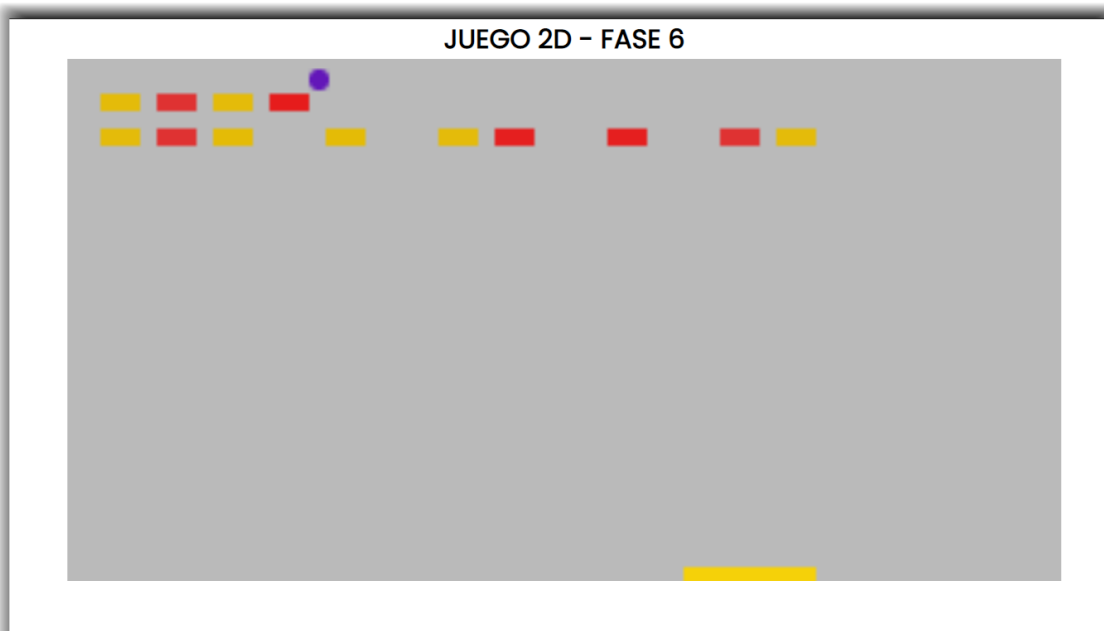
```

```

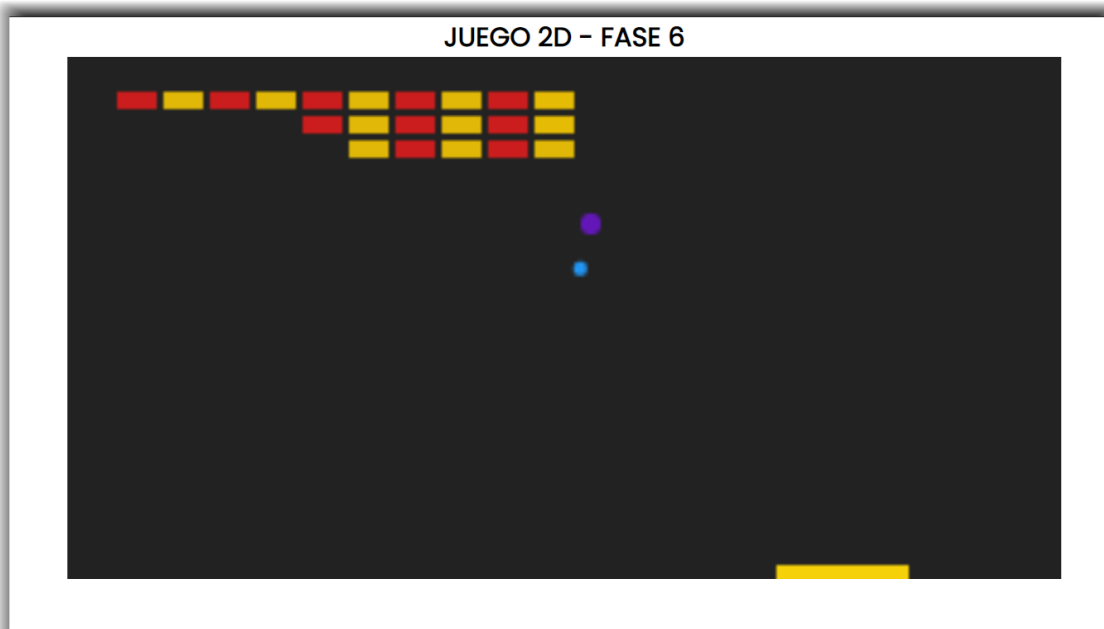
197.     }
198.     if (y + dy < ballRadius) {
199.         dy = -dy;
200.         var cambia = document.querySelector("canvas");
201.         cambia.classList.toggle("activado");
202.         colorLadrillos = true;
203.     }
204.
205.     if (x2 + dx2 > canvas.width - ballRadius || x2 + dx2 < ballRadius) {
206.         dx2 = -dx2;
207.     }
208.     if (y2 + dy2 < ballRadius) {
209.         dy2 = -dy2;
210.     }
211.
212.     // Rebotar en Paleta o Game Over
213.     if (y + dy > canvas.height - ballRadius) {
214.         if (x > paletaPosX && x < paletaPosX + anchuraPaleta) {
215.             dy = -dy;
216.         } else {
217.             clearInterval(juego);
218.             alert("Game Over");
219.             document.location.reload();
220.         }
221.     }
222.     if (azarSegundaBola > 50) {
223.         if (y2 + dy2 > canvas.height - ballRadius) {
224.             if (x2 > paletaPosX && x2 < paletaPosX + anchuraPaleta) {
225.                 dy2 = -dy2;
226.             } else {
227.                 clearInterval(juego);
228.                 alert("Game Over");
229.                 document.location.reload();
230.             }
231.         }
232.     }
233.     // Movimiento Paleta
234.     if (flechaDerechaPulsada && paletaPosX < canvas.width - anchuraPaleta) {
235.         paletaPosX += 7;
236.     } else if (flechaIzquierdaPulsada && paletaPosX > 0) {
237.         paletaPosX -= 7;
238.     }
239.     // Se incrementan los valores de la posición de las bolas
240.     x += dx;
241.     y += dy;
242.     x2 += dx2;
243.     y2 += dy2;
244. }
245.
246.     var juego = setInterval(dibujar, 25);
247. </script>
248. </body>
249. </html>

```

El resultado tras ejecutar el anterior código es:



Gráfica 10. Interfaz cuando la bola morada rebota en los ladrillos.



Gráfica 11. Interfaz cuando una de las dos bolas rebota en los ladrillos.

En la gráfica anterior vemos como la bola morada al chocar con los ladrillos estos desaparecen, hay que tener en cuenta que nos puede salir dos bolas lo cual la segunda también tendrá la misma propiedad.

En los siguientes apartados se explicaran las fases finales.

## 9 FASE 7: CONTAR PUNTOS, GANAR, GOLPE Y DEGRADADO DE LADRILLOS

---

En esta fase del juego haremos unos cambios para la mejora de este, también agregaremos una variable que nos permitirá contar los puntos.

En el código HTML hemos introducido un elemento div con su respectivo identificador, el cual nos servirá para poder escribir la variable puntaje correspondiente, este elemento cuenta con sus respectivos estilos, como el posicionamiento etc.

Uno de los cambios es la anchura de la paleta, esta dependerá si nuestro juego cuenta con dos bolas, si recordamos esto es al azar, si es así la paleta podrá ser más grande.

En la parte de los ladrillos cambiamos el status que nos permitía saber si el ladrillo es visible (1) o no, está la cambiamos por un estado inicialmente en 10 para posteriormente hacer un degradado con los ladrillos.

Uno de los grandes cambios se encuentra en la función dibujarLadrillos(), el contenido es similar a la fase anterior, lo que cambia es que al golpear la bola un ladrillo, este comenzara a restarle uno al estado correspondiente, lo cual el estado en vez de valer 10 pasa a valer 9 y luego a valer 8 y así sucesivamente hasta 1, cada vez que cambie el estado también cambiara el color del ladrillo y cuando este llegue a 1 el ladrillo correspondiente desaparecerá.

El cambio más grande se da en la función deteccionColision(), en donde creamos una función interna para disminuir el estado y el ciclo, además para sumarle uno al puntaje y ver si se ha ganado, hacemos esta función para evitar más adelante copiar el mismo código dos veces. También creamos una variable con un número al azar cada vez que se ejecuta esta función, este nos servirá cuando la bola golpee un ladrillo, ya que si el número es par el ladrillo desaparecerá, la bola rebotara y el puntaje se sumara, pero si es impar la bola rebotara pero no eliminara el ladrillo ni sumara puntos, como sabemos esto es al azar entonces no sabemos si al golpear el ladrillo por primera vez o si n veces.

Creamos una nueva función para imprimir la variable puntaje, esta suma cuando la bola golpea un ladrillo y lo elimina, dentro de las instrucciones para el tipo de letra, el tamaño y el color del texto.

El código es el siguiente:

```

1. <!DOCTYPE html>
2. <html>
3.
4. <head>
5.   <meta charset="UTF-8" />
6.   <title>Juego 2D</title>
7.
8.   <style>
9.     @import url("https://fonts.googleapis.com/css2?family=Poppins&display=swap");
10.
11.     * {
12.       font-family: "Poppins", sans-serif;
13.       padding: 0;
14.       margin: 0;
15.     }
16.
17.     body {
18.       margin: 0 200px 0 200px;
19.     }
20.
21.     h1 {
22.       font-size: 2em;
23.       text-transform: uppercase;
24.       text-align: center;
25.     }
26.
27.     canvas {
28.       width: 100%;
29.       height: 80vh;
30.       background-color: #dddddd;
31.       transition: 0.5s;
32.       display: block;
33.       margin: 50px auto;
34.     }
35.
36.     canvas.activado {
37.       background-color: #222222;
38.     }
39.   </style>
40. </head>
41.
42. <body>
43.   <h1>Juego 2D - Fase 7</h1>
44.   <canvas id="miCanvas"> </canvas>
45.
46.   <script>
47.     var canvas = document.getElementById("miCanvas");
48.     var ctx = canvas.getContext("2d");
49.
50.     var ballRadius = 3;
51.     var x = canvas.width / 2;
52.     var y = canvas.height - 30;
53.     var dx = 2;
54.     var dy = -2;
55.
56.     var x2 = x;
57.     var y2 = y;
58.     var dx2 = -0.8;
59.     var dy2 = -0.8;
60.
61.     // Variable para saber si se dibuja la segunda bola depende del numero arrojado al azar
62.     var segundaBola = false;
63.     if(Math.random() * 100 > 50){
64.       segundaBola = true;
65.     }
66.
67.     // Paleta
68.     var alturaPaleta = 4;
69.     var anchuraPaleta = 40;
70.
71.     // Si el valor de la variable segundaBola es true, hara que la paleta sea mas grand

```

```

72.
73.     if(segundaBola){
74.         anchuraPaleta = 60;
75.     }
76.
77.     var paletaPosX = (canvas.width - anchuraPaleta) / 2;
78.     var flechaDerechaPulsada = false;
79.     var flechaIzquierdaPulsada = false;
80.
81.     // VARIABLES ASOCIADAS A LOS LADRILLOS
82.
83.     var numFilasLadrillos = 0;
84.     var numColumnasLadrillos = 0;
85.
86.     // Se crea un bucle para poner un numero al azar de ladrillos y columnas, esto para que nos arroje
    e un numeros entre 1 y 12 en filas y entre 1 y 8 en columnas
87.
88.     do {
89.         numFilasLadrillos = Math.round(Math.random() * 100);
90.         numColumnasLadrillos = Math.round(Math.random() * 10);
91.     } while (numFilasLadrillos < 1 || numFilasLadrillos > 12 || numColumnasLadrillos < 1 || numColumn
asLadrillos > 8);
92.
93.     var anchoLadrillo = 20;
94.     var alturaLadrillo = 5;
95.     var rellenoLadrillos = 2;
96.     var vacioSuperiorLadrillos = 10;
97.     var vacioIzquierdoLadrillos = 17;
98.
99.     var puntaje = 0;
100.
101.     var ladrillos = [];
102.
103.     for (var columna = 0; columna < numColumnasLadrillos; columna++) {
104.         ladrillos[columna] = [];
105.
106.         for (var fila = 0; fila < numFilasLadrillos; fila++) {
107.
108.             // estado definira si el ladrillo es visible o no en 10 sera visible, mas adelante permit
            e hacer un degradado
109.             ladrillos[columna][fila] = { x: 0, y: 0, estado: 10, ciclo: 10 };
110.         }
111.     }
112.
113.     document.addEventListener("keydown", teclaPresionada, false);
114.     document.addEventListener("keyup", teclaLiberada, false);
115.
116.     function teclaPresionada(e) {
117.         if (e.keyCode == 39) {
118.             flechaDerechaPulsada = true;
119.         } else if (e.keyCode == 37) {
120.             flechaIzquierdaPulsada = true;
121.         }
122.     }
123.
124.     function teclaLiberada(e) {
125.         if (e.keyCode == 39) {
126.             flechaDerechaPulsada = false;
127.         } else if (e.keyCode == 37) {
128.             flechaIzquierdaPulsada = false;
129.         }
130.     }
131.
132.     function dibujarBola(cordernaX, cordenaY, tamaño, color) {
133.         ctx.beginPath();
134.         ctx.arc(cordernaX, cordenaY, tamaño, 0, Math.PI * 2);
135.         ctx.fillStyle = color;
136.         ctx.fill();
137.         ctx.closePath();
138.     }
139.
140.     function dibujarPaleta() {
141.         ctx.beginPath();
142.         ctx.rect(paletaPosX, canvas.height - alturaPaleta, anchuraPaleta, alturaPaleta);
143.         ctx.fillStyle = "#f5d109";
144.         ctx.fill();

```

```

144.         ctx.closePath();
145.     }
146.
147.     var colorLadrillos = false; // La utilizaremos para cambiar el color de Los Ladrillos
148.
149.     function dibujarLadrillos() {
150.         for (var columnas = 0; columnas < numColumnasLadrillos; columnas++) {
151.             for (var filas = 0; filas < numFilasLadrillos; filas++) {
152.
153.                 if (ladrillos[columnas][filas].estado > 0) {
154.
155.                     var b = ladrillos[columnas][filas];
156.
157.                     var brickX = filas * (anchoLadrillo + rellenoLadrillos) + vacioIzquierdoLadrillos
158.                 ;
159.                     var brickY = columnas * (alturaLadrillo + rellenoLadrillos) + vacioSuperiorLadril
160.                 los;
161.
162.                     // Se reduce el estado y el ciclo del ladrillo para posteriormente hacer un degra
163.                 dado y eliminar el ladrillo
164.
165.                     if (b.estado < 10 && b.estado > 1) {
166.                         b.ciclo--;
167.                         if (b.ciclo == 0) {
168.                             if (b.estado > 1) {
169.                                 b.estado--;
170.                                 b.ciclo = 10;
171.                             }
172.                         }
173.                     }
174.
175.                     ladrillos[columnas][filas].x = brickX;
176.                     ladrillos[columnas][filas].y = brickY;
177.
178.                     ctx.beginPath();
179.                     ctx.rect(brickX, brickY, anchoLadrillo, alturaLadrillo);
180.
181.                     // Las filas cuyo numero es par, se pintaran diferentes a Las impares
182.
183.                     if (b.estado == 10) {
184.                         if (colorLadrillos) {
185.                             if (filas % 2) {
186.                                 ctx.fillStyle = "#e61c1cdc";
187.                             } else {
188.                                 ctx.fillStyle = "#e4bb06fa";
189.                             }
190.                         } else {
191.                             if (filas % 2) {
192.                                 ctx.fillStyle = "#e4bb06fa";
193.                             } else {
194.                                 ctx.fillStyle = "#e61c1cdc";
195.                             }
196.                         }
197.
198.                     }
199.
200.                     // Como anteriormente el estado se estara reduciendo Los Ladrillos que hayan sido
201.                 golpeados, cambian de color hasta desaparecer.
202.
203.                     } else if (b.estado == 9) {
204.                         ctx.fillStyle = "#fad638";
205.                     } else if (b.estado == 8) {
206.                         ctx.fillStyle = "#fadb51";
207.                     } else if (b.estado == 7) {
208.                         ctx.fillStyle = "#fbe16a";
209.                     } else if (b.estado == 6) {
210.                         ctx.fillStyle = "#fce683";
211.                     } else if (b.estado == 5) {
212.                         ctx.fillStyle = "#fceb9c";
213.                     } else if (b.estado == 4) {
214.                         ctx.fillStyle = "#fdf0b4";
215.                     } else if (b.estado == 3) {
216.                         ctx.fillStyle = "#fef5cd";
217.                     } else if (b.estado == 2) {
218.                         ctx.fillStyle = "#fefae6";
219.                     } else if (b.estado == 1) {
220.                         ctx.fillStyle = "transparent";
221.                     }
222.                 }
223.             }
224.         }
225.     }

```



```

215.         }
216.         ctx.fill();
217.         ctx.closePath();
218.     }
219. }
220. }
221. }
222.
223. // Detectar la colision
224. function deteccionColision() {
225.     // Los ciclos recorren todos los ladrillos
226.     for (var c = 0; c < numColumnasLadrillos; c++) {
227.         for (var f = 0; f < numFilasLadrillos; f++) {
228.
229.             var b = ladrillos[c][f];
230.
231.             // Funcion para reducir el estado, el ciclo, sumar el puntaje y ver si se ha ganado,
232.             // esto es para evitar copiar el mismo código mas adelante
233.
234.             function sumarPuntaje(){
235.                 b.estado--;
236.                 b.ciclo--;
237.                 puntaje++;
238.                 if (puntaje == numFilasLadrillos * numColumnasLadrillos) {
239.                     alert("FELICITACIONES: HAS GANADO");
240.                     document.location.reload();
241.                 }
242.             }
243.
244.             /*
245.             En la variable a continuacion nos permitira obtener un numero al azar
246.
247.             Este si es par, y si el ladrillo con el cual ha chocado es visible entonces rebota
248.             ara la bola, llamara a la funcion anteriormente hecha, la cual reducirá el estado y hará que el ladrillo
249.             se degrade además sumará 1 al puntaje,
250.
251.             De lo contrario si el numero es impar la bola rebotará pero no llamará a la funci
252.             on anteriormente hecha, lo cual el estado no se reduce, o sea el ladrillo no desaparece y también el punt
253.             aje no se suma
254.
255.             Recordemos que podemos tener dos bolas, lo cual es el mismo proceso solo que camb
256.             iando los valores correspondientes a su posición
257.             */
258.
259.             var num = (Math.round(Math.random() * 100));
260.
261.             if (b.estado == 10) {
262.
263.                 // se modifica si la bola llega al ladrillo
264.                 if (x > b.x && x < b.x + anchoLadrillo && y > b.y && y < b.y + alturaLadrillo
265. ) {
266.                     dy = -dy;
267.
268.                     if(num % 2){
269.                         sumarPuntaje();// Llama a la funcion para reducir el ciclo, el estado
270.                         y sumar el puntaje
271.                     }
272.                 }
273.
274.                 if (segundaBola) {
275.                     if (x2 > b.x && x2 < b.x + anchoLadrillo && y2 > b.y && y2 < b.y + altura
276. Ladrillo) {
277.                         dy2 = -dy2;
278.
279.                         if(num % 2){
280.                             sumarPuntaje();// Llama a la funcion para reducir el ciclo, el es
281. tado y sumar el puntaje
282.                         }
283.                     }
284.                 }
285.             }
286.         }
287.     }
288. }
289.

```

```

280. // Esta funcion permitira que la variable puntaje se pueda visualizar en la pantalla, dentro hay
    una variable que almacena de manera de texto la palabra puntaje con el valor del puntaje correspondientes
    , ejemplo "Puntaje: 5". Y gracias al elemento div en HTML se mostrara lo que devuelve esta funcion, esta
    es tambien llamada en la funcion principal dibujar()
281.
282.     function dibujarPuntos(){
283.         ctx.font = "24px Poppins"
284.         ctx.fillStyle = #2496f3;
285.         ctx.fillText("puntaje: " + puntaje, 8, 20);
286.     }
287.
288. // Función principal para generar los procesos
289. function dibujar() {
290.     ctx.clearRect(0, 0, canvas.width, canvas.height);
291.     dibujarBola(x, y, ballRadius, "#631789"); // dibuja la bola
292.
293.     // Dibuja una segunda bola
294.     if (segundaBola) {
295.         dibujarBola(x2, y2, ballRadius - 1, "#2196f3");
296.     }
297.
298.     dibujarPaleta(); // Dibuja la paleta
299.     dibujarLadrillos(); // Dibuja el conjunto de los ladrillos
300.
301.     // SE LLAMA A LA FUNCION DETECTAR COLISION
302.     deteccionColision();
303.
304.     // SE LLAMA A LA FUNCION DIBUJAR PUNTOS
305.     dibujarPuntos();
306.
307.     // Rebote de la bola
308.     if (x + dx > canvas.width - ballRadius || x + dx < ballRadius) {
309.         dx = -dx;
310.         colorLadrillos = false; // Cambia el color de los ladrillos si la bola principal rebota en
x
311.     }
312.     if (y + dy < ballRadius) {
313.         dy = -dy;
314.         var cambia = document.querySelector("canvas");
315.         cambia.classList.toggle("activado");
316.         colorLadrillos = true;
317.     }
318.
319.     if (x2 + dx2 > canvas.width - ballRadius || x2 + dx2 < ballRadius) {
320.         dx2 = -dx2;
321.     }
322.     if (y2 + dy2 < ballRadius) {
323.         dy2 = -dy2;
324.     }
325.
326.     // Rebotar en Paleta o Game Over
327.     if (y + dy > canvas.height - ballRadius) {
328.         if (x > paletaPosX && x < paletaPosX + anchuraPaleta) {
329.             dy = -dy;
330.         } else {
331.             clearInterval(juego);
332.             alert("Game Over");
333.             document.location.reload();
334.         }
335.     }
336.     if (segundaBola) {
337.         if (y2 + dy2 > canvas.height - ballRadius) {
338.             if (x2 > paletaPosX && x2 < paletaPosX + anchuraPaleta) {
339.                 dy2 = -dy2;
340.             } else {
341.                 clearInterval(juego);
342.                 alert("Game Over");
343.                 document.location.reload();
344.             }
345.         }
346.     }
347.
348.     // Movimiento Paleta
349.     if (flechaDerechaPulsada && paletaPosX < canvas.width - anchuraPaleta) {
350.         paletaPosX += 7;

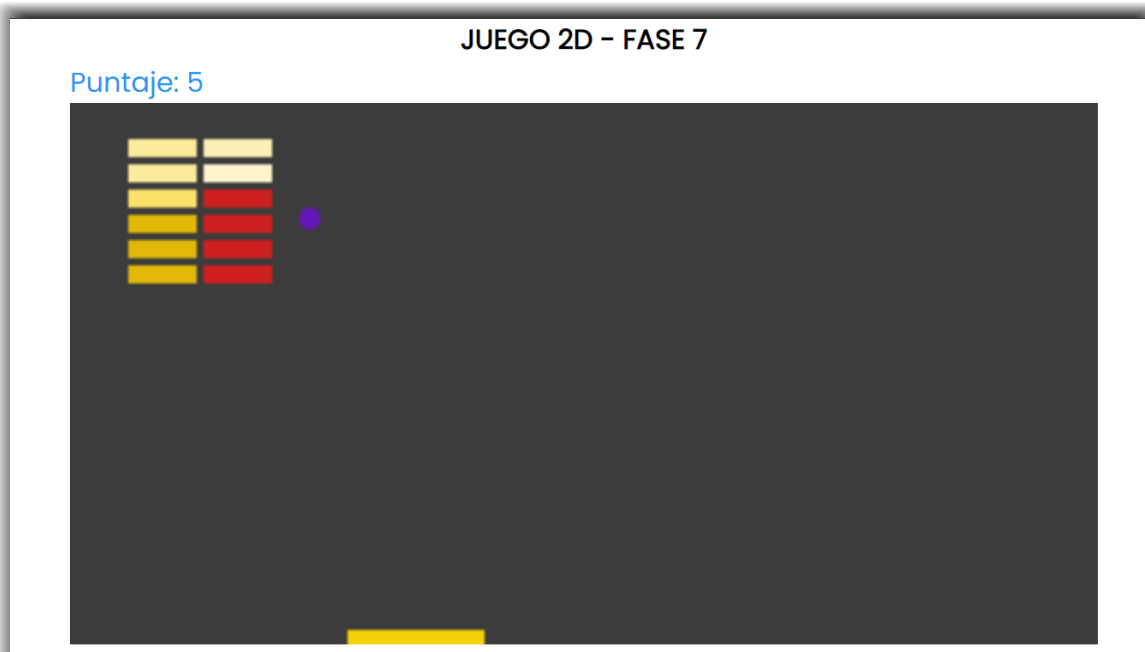
```

```

351.         } else if (flechaIzquierdaPulsada && paletaPosX > 0) {
352.             paletaPosX -= 7;
353.         }
354.
355.         // Se incrementan los valores de la posición de las bolas
356.         x += dx;
357.         y += dy;
358.         x2 += dx2;
359.         y2 += dy2;
360.     }
361.
362.     var juego = setInterval(dibujar, 20);
363. </script>
364. </body>
365. </html>

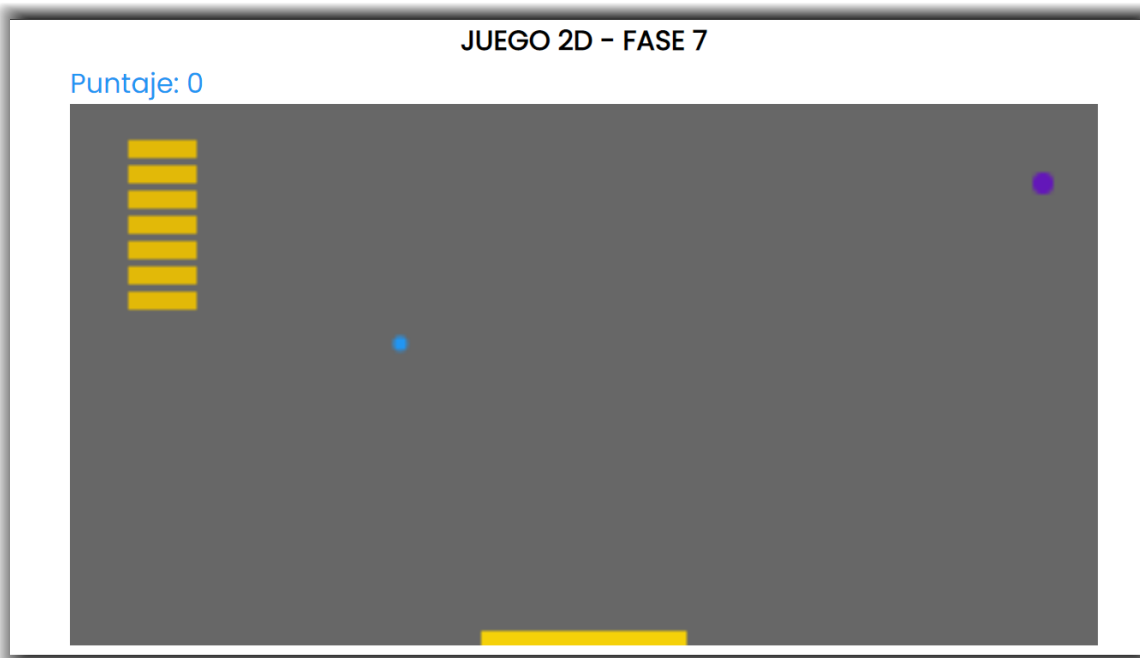
```

El resultado es el siguiente:



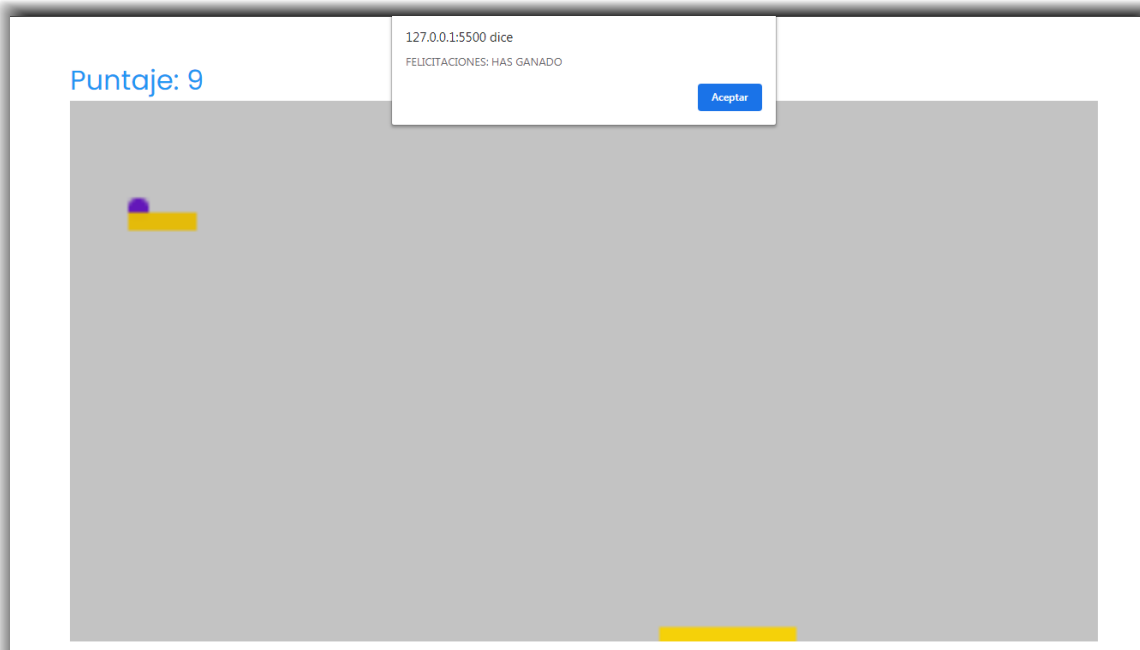
Gráfica 12. Interfaz cuando la bola golpea los ladrillos.

En la gráfica anterior podemos observar como la bola golpea los ladrillos, los cuales comenzaron a cambiar el color para luego desaparecer, además el puntaje se incrementa a un costado de la pantalla.



Gráfica 13. Interfaz cuando la condición de dibujar dos bolas se cumple.

Como podemos observar en la gráfica 13, al salir dos bolas la paleta cambia su tamaño, en este caso es más grande.



Gráfica 14. Interfaz cuando se han eliminado todos los ladrillos.



## 10 FASE 8: CONTROLANDO EL RATÓN

---

En esta parte del programa haremos que la paleta en lugar de ser movida por las flechas sea movida por el mouse.

Esto se obtiene creando una función llamada `function manejadorRaton(e)` a la cual se le da una variable que permitirá almacenar la posición del ratón en la pantalla, y gracias a una condicional se puede deducir si este está dentro del canvas, al cumplirla hace que la paleta pueda ser desplazada mediante el mouse.

```
1. <!DOCTYPE html>
2. <html>
3.
4. <head>
5.   <meta charset="UTF-8" />
6.   <title>Juego 2D</title>
7.
8.   <style>
9.     @import url("https://fonts.googleapis.com/css2?family=Poppins&display=swap");
10.
11.     * {
12.       font-family: "Poppins", sans-serif;
13.       padding: 0;
14.       margin: 0;
15.     }
16.
17.     body {
18.       margin: 0 200px 0 200px;
19.     }
20.
21.     h1 {
22.       font-size: 2em;
23.       text-transform: uppercase;
24.       text-align: center;
25.     }
26.
27.     canvas {
28.       background-color: #dddddd;
29.       transition: 0.5s;
30.       display: block;
31.       margin: 50px auto;
32.     }
33.
34.     canvas.activado {
35.       background-color: #222222;
36.     }
37.
38.     #puntos {
39.       font-size: 2em;
40.       position: fixed;
41.       top: 50px;
42.       color: #2196f3;
43.     }
44.   </style>
45. </head>
46.
47. <body>
48.   <h1>Juego 2D - Fase 8</h1>
49.   <canvas id="miCanvas" width="1200" height="640"> </canvas>
50.
51.   <div id="puntos">
52.   </div>
53.
54.   <script>
55.     var canvas = document.getElementById("miCanvas");
```

```

56.     var ctx = canvas.getContext("2d");
57.
58.     var ballRadius = 10;
59.     var x = canvas.width / 2;
60.     var y = canvas.height - 30;
61.     var dx = 5;
62.     var dy = -5;
63.
64.     var x2 = x;
65.     var y2 = y;
66.     var dx2 = -1.5;
67.     var dy2 = -1.5;
68.
69.     var segundaBola = false;
70.     if (Math.random() * 100 > 50) {
71.         segundaBola = true;
72.     }
73.
74.     // Paleta
75.     var alturaPaleta = 15;
76.     var anchuraPaleta = 140;
77.
78.     if (segundaBola) {
79.         anchuraPaleta = 180;
80.     }
81.
82.     var paletaPosX = (canvas.width - anchuraPaleta) / 2;
83.     var flechaDerechaPulsada = false;
84.     var flechaIzquierdaPulsada = false;
85.
86.     var numFilasLadrillos = 0;
87.     var numColumnasLadrillos = 0;
88.
89.     do {
90.         numFilasLadrillos = Math.round(Math.random() * 100);
91.         numColumnasLadrillos = Math.round(Math.random() * 10);
92.     } while (numFilasLadrillos < 1 || numFilasLadrillos > 14 || numColumnasLadrillos < 1 || numColumnasLadrillos > 8);
93.
94.     var anchoLadrillo = 70;
95.     var alturaLadrillo = 20;
96.     var rellenoLadrillos = 10;
97.     var vacioSuperiorLadrillos = 30;
98.     var vacioIzquierdoLadrillos = 45;
99.
100.    var puntaje = 0;
101.
102.    var ladrillos = [];
103.
104.    for (var columna = 0; columna < numColumnasLadrillos; columna++) {
105.        ladrillos[columna] = [];
106.        for (var fila = 0; fila < numFilasLadrillos; fila++) {
107.            ladrillos[columna][fila] = { x: 0, y: 0, estado: 10, ciclo: 10 };
108.        }
109.    }
110.
111.    document.addEventListener("keydown", teclaPresionada, false);
112.    document.addEventListener("keyup", teclaLiberada, false);
113.
114.    document.addEventListener("mousemove", manejadorRaton, false);
115.
116.    function teclaPresionada(e) {
117.        if (e.keyCode == 39) {
118.            flechaDerechaPulsada = true;
119.        } else if (e.keyCode == 37) {
120.            flechaIzquierdaPulsada = true;
121.        }
122.    }
123.
124.    function teclaLiberada(e) {
125.        if (e.keyCode == 39) {
126.            flechaDerechaPulsada = false;
127.        } else if (e.keyCode == 37) {
128.            flechaIzquierdaPulsada = false;
129.        }
130.    }

```

```

130.
131. // ESTE ES EL MANEJADOR DEL RATÓN
132. // -----
133. // La instrucción: "offsetLeft" calcula la distancia desde el borde izquierdo
134. // de la pantalla hasta un componente html
135. // -----
136. // Por tanto, la instrucción: "canvas.offsetLeft" calcula el espacio a la izquierda
137. // del objeto CANVAS
138. // -----
139. // Dentro del manejador del ratón, la instrucción: "e.clientX" calcula la posición
140. // del ratón en la pantalla. Para calcular la posición del ratón DENTRO del CANVAS
141. // debemos RESTAR a la posición X del ratón, el valor izquierdo del CANVAS
142. // -----
143. // Es decir: "e.clientX - canvas.offsetLeft"
144. // -----
145.
146. function manejadorRaton(e){
147.     var posXRatonDentroDeCanvas = e.clientX - canvas.offsetLeft;
148.
149.     // La siguiente condicional determinara si el raton esta dentro del CANVAS
150.
151.     if (posXRatonDentroDeCanvas > anchuraPaleta / 2 && posXRatonDentroDeCanvas < canvas.width -
152.         anchuraPaleta / 2) {
153.         // Si el raton esta dentro del canvas. Se recalcula la posicion de la paleta, la posicon
154.         // de la paleta es ahora la misma posicion del ratón. Se resta ya que la paleta tiene un ancho, gracias a est
155.         // o la paleta se mueve a basa del ratón
156.
157.         paletaPosX = posXRatonDentroDeCanvas - anchuraPaleta / 2;
158.     }
159.
160.     function dibujarBola(cordernaX, cordenaY, tamaño, color) {
161.         ctx.beginPath();
162.         ctx.arc(cordernaX, cordenaY, tamaño, 0, Math.PI * 2);
163.         ctx.fillStyle = color;
164.         ctx.fill();
165.         ctx.closePath();
166.     }
167.
168.     function dibujarPaleta() {
169.         ctx.beginPath();
170.         ctx.rect(paletaPosX, canvas.height - alturaPaleta, anchuraPaleta, alturaPaleta);
171.         ctx.fillStyle = "#f5d109";
172.         ctx.fill();
173.         ctx.closePath();
174.     }
175.
176.     var colorLadrillos = false; // La utilizaremos para cambiar el color de los ladrillos
177.
178.     function dibujarLadrillos() {
179.         for (var columnas = 0; columnas < numColumnasLadrillos; columnas++) {
180.             for (var filas = 0; filas < numFilasLadrillos; filas++) {
181.
182.                 if (ladrillos[columnas][filas].estado > 0) {
183.
184.                     var b = ladrillos[columnas][filas];
185.
186.                     var brickX = filas * (anchoLadrillo + rellenoLadrillos) + vacioIzquierdoLadrillos
187.
188.                     var brickY = columnas * (alturaLadrillo + rellenoLadrillos) + vacioSuperiorLadril
189.
190.                     los;
191.
192.                     if (b.estado < 10 && b.estado > 1) {
193.                         b.ciclo--;
194.                         if (b.ciclo == 0) {
195.                             if (b.estado > 1) {
196.                                 b.estado--;
197.                                 b.ciclo = 10;
198.                             }
199.                         }
200.                     }
201.
202.                     ladrillos[columnas][filas].x = brickX;
203.                     ladrillos[columnas][filas].y = brickY;

```

```

200.
201.         ctx.beginPath();
202.         ctx.rect(brickX, brickY, anchoLadrillo, alturaLadrillo);
203.
204.         if (b.estado == 10) {
205.             if (colorLadrillos) {
206.                 if (filas % 2) {
207.                     ctx.fillStyle = "#e61c1cdc";
208.                 } else {
209.                     ctx.fillStyle = "#e4bb06fa";
210.                 }
211.             } else {
212.                 if (filas % 2) {
213.                     ctx.fillStyle = "#e4bb06fa";
214.                 } else {
215.                     ctx.fillStyle = "#e61c1cdc";
216.                 }
217.             }
218.         }
219.         } else if (b.estado == 9) {
220.             ctx.fillStyle = "#fad638";
221.         } else if (b.estado == 8) {
222.             ctx.fillStyle = "#fbd51";
223.         } else if (b.estado == 7) {
224.             ctx.fillStyle = "#f8e16a";
225.         } else if (b.estado == 6) {
226.             ctx.fillStyle = "#fce683";
227.         } else if (b.estado == 5) {
228.             ctx.fillStyle = "#fceb9c";
229.         } else if (b.estado == 4) {
230.             ctx.fillStyle = "#fdf0b4";
231.         } else if (b.estado == 3) {
232.             ctx.fillStyle = "#fef5cd";
233.         } else if (b.estado == 2) {
234.             ctx.fillStyle = "#fefae6";
235.         } else if (b.estado == 1) {
236.             ctx.fillStyle = "transparent";
237.         }
238.         ctx.fill();
239.         ctx.closePath();
240.     }
241. }
242. }
243. }
244.
245. // Detectar la colision
246. function deteccionColision() {
247.     for (var c = 0; c < numColumnasLadrillos; c++) {
248.         for (var f = 0; f < numFilasLadrillos; f++) {
249.             var b = ladrillos[c][f];
250.             function sumarPuntaje() {
251.                 b.estado--;
252.                 b.ciclo--;
253.                 puntaje++;
254.                 if (puntaje == numFilasLadrillos * numColumnasLadrillos) {
255.                     alert("FELICITACIONES: HAS GANADO");
256.                     document.location.reload();
257.                 }
258.             }
259.             if (b.estado == 10) {
260.                 // se modifica si la bola llega al ladrillo
261.                 if (x > b.x && x < b.x + anchoLadrillo + ballRadius/2 && y > b.y && y < b.y + alturaLadrillo + ballRadius/2) {
262.                     dy = -dy;
263.                     if ((Math.round(Math.random() * 100)) % 2) {
264.                         sumarPuntaje(); // Llama a la funcion para reducir el ciclo, el estado y sumar el puntaje
265.                     }
266.                 }
267.             }
268.         }
269.     }
270.     if (segundaBola) {
271.         if (x2 > b.x && x2 < b.x + anchoLadrillo && y2 > b.y && y2 < b.y + alturaLadrillo) {

```



```

272.             dy2 = -dy2;
273.
274.             if ((Math.round(Math.random() * 100)) % 2) {
275.                 sumarPuntaje(); // Llama a la función para reducir el ciclo, el estado
y sumar el puntaje
276.             }
277.         }
278.     }
279. }
280. }
281. }
282. }
283.
284. function dibujarPuntos() {
285.     var str = 'Puntaje: ' + puntaje.toString();
286.     document.getElementById('puntos').innerText = str;
287. }
288.
289. function dibujar() {
290.     ctx.clearRect(0, 0, canvas.width, canvas.height);
291.
292.     // SE LLAMA A LA FUNCION Dibujar puntos
293.     dibujarPuntos();
294.
295.     dibujarLadrillos();
296.     dibujarBola(x, y, ballRadius, "#6317B9");
297.     if (segundaBola) {
298.         dibujarBola(x2, y2, ballRadius - 2, "#2196f3");
299.     }
300.     dibujarPaleta();
301.     deteccionColision();
302.
303.     // Rebote de la bola
304.     if (x + dx > canvas.width - ballRadius || x + dx < ballRadius) {
305.         dx = -dx;
306.         colorLadrillos = false; // Cambia el color de los ladrillos si la bola principal rebota en
x
307.     }
308.     if (y + dy < ballRadius) {
309.         dy = -dy;
310.         var cambia = document.querySelector("canvas");
311.         cambia.classList.toggle("activado");
312.         colorLadrillos = true;
313.     }
314.
315.     if (x2 + dx2 > canvas.width - ballRadius || x2 + dx2 < ballRadius) {
316.         dx2 = -dx2;
317.     }
318.     if (y2 + dy2 < ballRadius) {
319.         dy2 = -dy2;
320.     }
321.
322.     // Rebotar en Paleta o Game Over
323.     if (y + dy > canvas.height - ballRadius) {
324.         if (x > paletaPosX && x < paletaPosX + anchuraPaleta) {
325.             dy = -dy;
326.         } else {
327.             clearInterval(juego);
328.             alert("Game Over");
329.             document.location.reload();
330.         }
331.     }
332.     if (segundaBola) {
333.         if (y2 + dy2 > canvas.height - ballRadius) {
334.             if (x2 > paletaPosX && x2 < paletaPosX + anchuraPaleta) {
335.                 dy2 = -dy2;
336.             } else {
337.                 clearInterval(juego);
338.                 alert("Game Over");
339.                 document.location.reload();
340.             }
341.         }
342.     }
343.
344.     // Movimiento Paleta

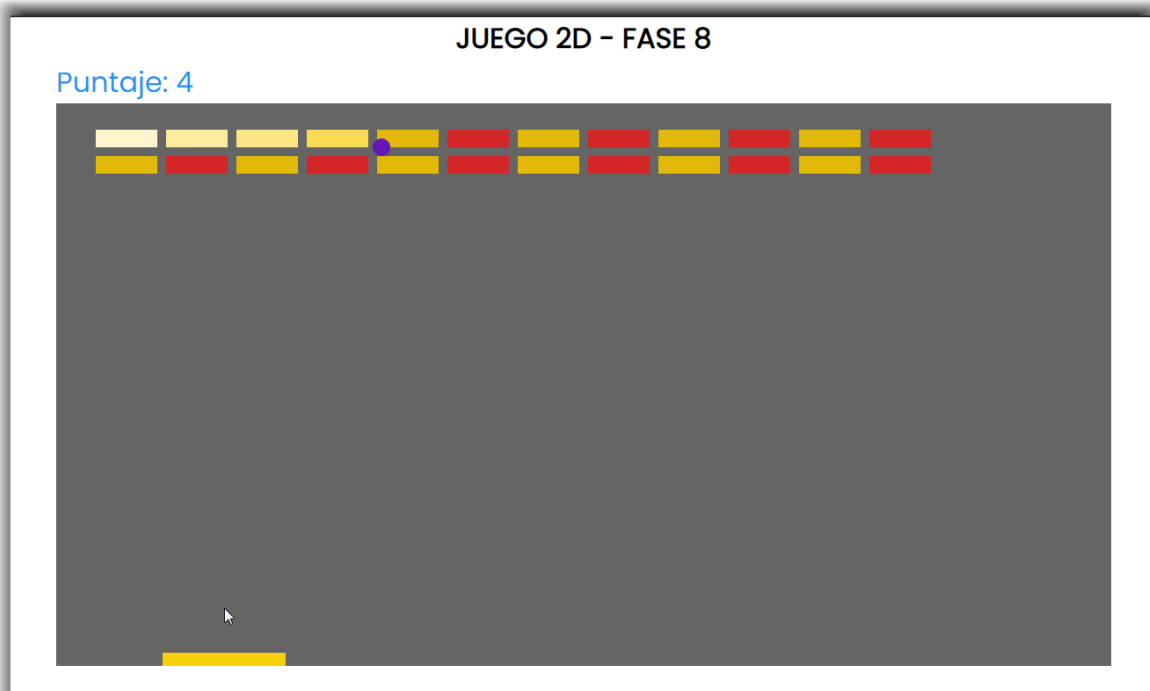
```

```

345.         if (flechaDerechaPulsada && paletaPosX < canvas.width - anchuraPaleta) {
346.             paletaPosX += 15;
347.         } else if (flechaIzquierdaPulsada && paletaPosX > 0) {
348.             paletaPosX -= 15;
349.         }
350.
351.         // Se incrementan los valores de la poción de las bolas
352.         x += dx;
353.         y += dy;
354.         x2 += dx2;
355.         y2 += dy2;
356.     }
357.
358.     var juego = setInterval(dibujar, 15);
359. </script>
360. </body>
361.
362. </html>

```

El resultado es:



Gráfica 14. Interfaz cuando se mueve el ratón dentro del canvas.

En la gráfica anterior podemos observar que el cursor del mouse está dentro del canvas, y como se muestra este se mueve en la mitad de la paleta

En el siguiente apartado se explicará la siguiente fase del juego. En caso de ser necesario, se agregarán todas las explicaciones que sean necesarias para que el juego quede debidamente explicado.

## 11 FASE 9: FINALIZANDO EL JUEGO

En esta parte del programa ya se agregan los toques finales tales como vidas para el jugador, ocultar el mouse entre otras.

Se crea una variable `var vidas = 3` con la instrucción de controlar las vidas que tiene dentro del juego cada participante y se crea otra variable `canvas.style.cursor = 'none'` para ocultar el mouse dentro del campo del juego.

Se crea una nueva función `quitarVidas()`, esta permitirá reducir las vidas en uno cuando se llamada, además evaluara si se han agotado las vidas para terminar el juego, de lo contrario el juego debe continuar por lo tanto se debe colocar las bolas y paleta en la posición inicial.

```

1. <!DOCTYPE html>
2. <html>
3.
4. <head>
5.   <meta charset="UTF-8" />
6.   <title>Juego 2D</title>
7.
8.   <style>
9.     @import url("https://fonts.googleapis.com/css2?family=Poppins&display=swap");
10.
11.     * {
12.       font-family: "Poppins", sans-serif;
13.       padding: 0;
14.       margin: 0;
15.     }
16.
17.     h1 {
18.       font-size: 2em;
19.       text-transform: uppercase;
20.       text-align: center;
21.     }
22.
23.     canvas {
24.       background-color: #dddddd;
25.       transition: 0.5s;
26.       display: block;
27.       margin: auto;
28.
29.       /* OCULTA EL CURSOR EN EL CANVAS */
30.       cursor: none;
31.     }
32.
33.     canvas.activado {
34.       background-color: #222222;
35.     }
36.
37.   </style>
38. </head>
39.
40. <body>
41.   <h1>Juego 2D - Fase 9</h1>
42.
43.   <canvas id="miCanvas" width="1200" height="640"> </canvas>
44.
45.   <script>
46.     var canvas = document.getElementById("miCanvas");
47.     var ctx = canvas.getContext("2d");

```



```
48.
49.     var ballRadius = 10;
50.     var x = canvas.width / 2;
51.     var y = canvas.height - 30;
52.     var dx = 5;
53.     var dy = -5;
54.
55.     var x2 = x;
56.     var y2 = y;
57.     var dx2 = -1.5;
58.     var dy2 = -1.5;
59.
60.     var segundaBola = false;
61.     if (Math.random() * 100 > 50) {
62.         segundaBola = true;
63.     }
64.
65.     // Paleta
66.     var alturaPaleta = 15;
67.     var anchuraPaleta = 140;
68.
69.     if (segundaBola) {
70.         anchuraPaleta = 180;
71.     }
72.
73.     var paletaPosX = (canvas.width - anchuraPaleta) / 2;
74.     var flechaDerechaPulsada = false;
75.     var flechaIzquierdaPulsada = false;
76.
77.     var numFilasLadrillos = 0;
78.     var numColumnasLadrillos = 0;
79.
80.     do {
81.         numFilasLadrillos = Math.round(Math.random() * 100);
82.         numColumnasLadrillos = Math.round(Math.random() * 10);
83.     } while (numFilasLadrillos < 1 || numFilasLadrillos > 14 || numColumnasLadrillos < 1 || numColumnasLadrillos > 8);
84.
85.     var anchoLadrillo = 70;
86.     var alturaLadrillo = 20;
87.     var rellenoLadrillos = 10;
88.     var vacioSuperiorLadrillos = 50;
89.     var vacioIzquierdoLadrillos = 45;
90.
91.     var puntaje = 0;
92.
93.     // ESTA INSTRUCCIÓN CONTROLA EL NÚMERO DE VIDAS DEL JUGADOR
94.     // CUANDO LA INSTRUCCIÓN vidas DISMINUYE A CERO, EL JUGADOR PIERDE,
95.     // PUESTO QUE HA PERDIDO TRES VECES
96.
97.     // ESTA VARIABLE DEFINE UN COLOR
98.     // Se pueden utilizar otros colores para los diferentes elementos del juego
99.
100.    var colorFigura = "#ff0000";
101.    var colorBola = "#6317B9";
102.    var colorBola2 = "#2196f3";
103.    var colorPaleta = "#f5d109";
104.    var colorLadrillo = "#e61c1cdc";
105.    var colorLadrillo2 = "#e4bb06fa";
106.    var colorTexto = "#2196f3";
107.    var vidas = 3;
108.
109.    // ESTA INSTRUCCIÓN OCULTA EL CURSOR DEL RATON (DENTRO DEL CANVAS)
110.    canvas.style.cursor = 'none';
111.
112.    var ladrillos = [];
113.
114.    for (var columna = 0; columna < numColumnasLadrillos; columna++) {
115.        ladrillos[columna] = [];
116.        for (var fila = 0; fila < numFilasLadrillos; fila++) {
117.            ladrillos[columna][fila] = { x: 0, y: 0, estado: 10, ciclo: 10 };
118.        }
119.    }
120.
121.    document.addEventListener("keydown", teclaPresionada, false);
```

```

122.     document.addEventListener("keyup", teclaLiberada, false);
123.
124.     document.addEventListener("mousemove", manejadorRaton, false);
125.
126.     function teclaPresionada(e) {
127.         if (e.keyCode == 39) {
128.             flechaDerechaPulsada = true;
129.         } else if (e.keyCode == 37) {
130.             flechaIzquierdaPulsada = true;
131.         }
132.     }
133.     function teclaLiberada(e) {
134.         if (e.keyCode == 39) {
135.             flechaDerechaPulsada = false;
136.         } else if (e.keyCode == 37) {
137.             flechaIzquierdaPulsada = false;
138.         }
139.     }
140.
141.     function manejadorRaton(e) {
142.         var posXRatonDentroDeCanvas = e.clientX - canvas.offsetLeft;
143.         if (posXRatonDentroDeCanvas > anchuraPaleta / 2 && posXRatonDentroDeCanvas < canvas.width -
144. anchuraPaleta / 2) {
145.             paletaPosX = posXRatonDentroDeCanvas - anchuraPaleta / 2;
146.         }
147.
148.         function dibujarBola(cordernaX, cordenaY, tamaño, color) {
149.             ctx.beginPath();
150.             ctx.arc(cordernaX, cordenaY, tamaño, 0, Math.PI * 2);
151.             ctx.fillStyle = color;
152.             ctx.fill();
153.             ctx.closePath();
154.         }
155.
156.         function dibujarPaleta() {
157.             ctx.beginPath();
158.             ctx.rect(paletaPosX, canvas.height - alturaPaleta, anchuraPaleta, alturaPaleta);
159.             ctx.fillStyle = colorPaleta;
160.             ctx.fill();
161.             ctx.closePath();
162.         }
163.
164.         var colorLadrillos = false; // La utilizaremos para cambiar el color de los ladrillos
165.
166.         function dibujarLadrillos() {
167.             for (var columnas = 0; columnas < numColumnasLadrillos; columnas++) {
168.                 for (var filas = 0; filas < numFilasLadrillos; filas++) {
169.
170.                     if (ladrillos[columnas][filas].estado > 0) {
171.
172.                         var b = ladrillos[columnas][filas];
173.
174.                         var brickX = filas * (anchoLadrillo + rellenoLadrillos) + vacioIzquierdoLadrillos
175. ;
176.                         var brickY = columnas * (alturaLadrillo + rellenoLadrillos) + vacioSuperiorLadril
177. los;
178.
179.                         if (b.estado < 10 && b.estado > 1) {
180.                             b.ciclo--;
181.                             if (b.ciclo == 0) {
182.                                 if (b.estado > 1) {
183.                                     b.estado--;
184.                                     b.ciclo = 10;
185.                                 }
186.                             }
187.                         }
188.
189.                         ladrillos[columnas][filas].x = brickX;
190.                         ladrillos[columnas][filas].y = brickY;
191.
192.                         ctx.beginPath();
193.                         ctx.rect(brickX, brickY, anchoLadrillo, alturaLadrillo);
194.
195.                         if (b.estado == 10) {

```

```

194.         if (colorLadrillos) {
195.             if (filas % 2) {
196.                 ctx.fillStyle = colorLadrillo;
197.             } else {
198.                 ctx.fillStyle = colorLadrillo2;
199.             }
200.         } else {
201.             if (filas % 2) {
202.                 ctx.fillStyle = colorLadrillo2;
203.             } else {
204.                 ctx.fillStyle = colorLadrillo;
205.             }
206.         }
207.
208.     } else if (b.estado == 9) {
209.         ctx.fillStyle = "#fad638";
210.     } else if (b.estado == 8) {
211.         ctx.fillStyle = "#fbdb51";
212.     } else if (b.estado == 7) {
213.         ctx.fillStyle = "#fbe16a";
214.     } else if (b.estado == 6) {
215.         ctx.fillStyle = "#fce683";
216.     } else if (b.estado == 5) {
217.         ctx.fillStyle = "#fceb9c";
218.     } else if (b.estado == 4) {
219.         ctx.fillStyle = "#fdf0b4";
220.     } else if (b.estado == 3) {
221.         ctx.fillStyle = "#fef5cd";
222.     } else if (b.estado == 2) {
223.         ctx.fillStyle = "#fefae6";
224.     } else if (b.estado == 1) {
225.         ctx.fillStyle = "transparent";
226.     }
227.     ctx.fill();
228.     ctx.closePath();
229. }
230. }
231. }
232. }
233.
234. // Detectar la colision
235. function deteccionColision() {
236.     for (var c = 0; c < numColumnasLadrillos; c++) {
237.         for (var f = 0; f < numFilasLadrillos; f++) {
238.             var b = ladrillos[c][f];
239.             function sumarPuntaje() {
240.                 b.estado--;
241.                 b.ciclo--;
242.                 puntaje++;
243.                 if (puntaje == numFilasLadrillos * numColumnasLadrillos) {
244.                     alert("FELICITACIONES: HAS GANADO");
245.                     document.location.reload();
246.                 }
247.             }
248.             if (b.estado == 10) {
249.
250.                 // se modifica si la bola llega al ladrillo
251.                 if (x > b.x && x < b.x + anchoLadrillo + ballRadius / 2 && y > b.y && y < b.y + alturaLadrillo + ballRadius / 2) {
252.                     dy = -dy;
253.
254.                     if ((Math.round(Math.random() * 100)) % 2) {
255.                         sumarPuntaje(); // Llama a la función para reducir el ciclo, el estado y sumar el puntaje
256.                     }
257.                 }
258.
259.                 if (segundaBola) {
260.                     if (x2 > b.x && x2 < b.x + anchoLadrillo && y2 > b.y && y2 < b.y + alturaLadrillo) {
261.                         dy2 = -dy2;
262.
263.                         if ((Math.round(Math.random() * 100)) % 2) {
264.                             sumarPuntaje(); // Llama a la función para reducir el ciclo, el estado y sumar el puntaje

```

```

265.         }
266.     }
267. }
268. }
269. }
270. }
271. }
272.
273. function dibujarPuntos() {
274.     ctx.font = "24px Poppins";
275.     ctx.fillStyle = colorTexto;
276.     ctx.fillText("puntaje: " + puntaje, 8, 20);
277. }
278.
279. function dibujarVidas() {
280.     ctx.font = "24px Poppins";
281.     ctx.fillStyle = colorTexto;
282.     // SE MUESTRA EL NÚMERO DE VIDAS DISPONIBLES
283.     ctx.fillText("vidas: " + vidas, canvas.width - 100, 20);
284. }
285.
286. /* La siguiente funcion permitira evitar escribir el codigo dos veces ya que tenemos posiblemente
dos bolas */
287.
288. function quitarVidas() {
289.     // SI SE PRODUCE UN CONTACTO DE LA BOLA CON LA BASE DEL CANVAS
290.     // SE PIERDE UNA VIDA. PARA ELLO, LA INSTRUCCIÓN vidas--;
291.     // LO CUAL EQUIVALE A: vidas = vidas - 1
292.     vidas--;
293.     if (!vidas) {
294.         // SI vidas == 0 (Lo cual también puede escribir: !vidas)
295.         // EL JUGADOR HA PERDIDO
296.         alert("GAME OVER");
297.         document.location.reload();
298.     } else {
299.         // SI vidas > 0 (diferente de CERO) EL JUEGO CONTINUA
300.         x = canvas.width / 2;
301.         y = canvas.height - 30;
302.         x2 = canvas.width / 2;
303.         y2 = canvas.height - 30;
304.         dx = 5;
305.         dy = -5;
306.         dx2 = -1.5;
307.         dy2 = -1.5;
308.         paletaPosX = (canvas.width - anchuraPaleta) / 2;
309.     }
310. }
311.
312. function dibujar() {
313.     ctx.clearRect(0, 0, canvas.width, canvas.height);
314.
315.     // SE LLAMA A LA FUNCION Dibujar puntos
316.     dibujarPuntos();
317.
318.     // SE LLAMA A LA FUNCION Dibujar puntos
319.     dibujarVidas();
320.
321.     dibujarLadrillos();
322.     dibujarBola(x, y, ballRadius, colorBola);
323.     if (segundaBola) {
324.         dibujarBola(x2, y2, ballRadius - 2, colorBola2);
325.     }
326.     dibujarPaleta();
327.     deteccionColision();
328.
329.     // Rebote de la bola
330.     if (x + dx > canvas.width - ballRadius || x + dx < ballRadius) {
331.         dx = -dx;
332.         colorLadrillos = false; // Cambia el color de los Ladrillos si la bola principal rebota en
333.         x
334.     }
335.     if (y + dy < ballRadius) {
336.         dy = -dy;
337.         var cambia = document.querySelector("canvas");
338.         cambia.classList.toggle("activado");

```

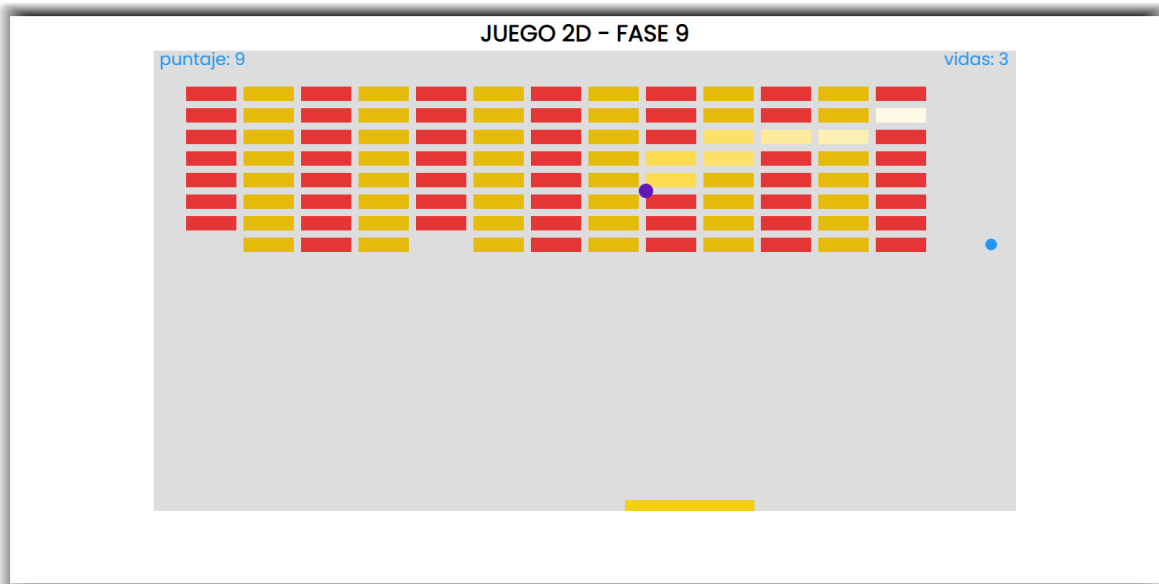
```

338.         colorLadrillos = true;
339.     }
340.
341.     if (x2 + dx2 > canvas.width - ballRadius || x2 + dx2 < ballRadius) {
342.         dx2 = -dx2;
343.     }
344.     if (y2 + dy2 < ballRadius) {
345.         dy2 = -dy2;
346.     }
347.
348.     // Rebotar en Paleta o Game Over
349.     if (y + dy > canvas.height - ballRadius) {
350.         if (x > paletaPosX && x < paletaPosX + anchuraPaleta) {
351.             dy = -dy;
352.         } else {
353.             quitarVidas();
354.         }
355.     }
356.     if (segundaBola) {
357.         if (y2 + dy2 > canvas.height - ballRadius) {
358.             if (x2 > paletaPosX && x2 < paletaPosX + anchuraPaleta) {
359.                 dy2 = -dy2;
360.             } else {
361.                 quitarVidas();
362.             }
363.         }
364.     }
365.
366.     // Movimiento Paleta
367.     if (flechaDerechaPulsada && paletaPosX < canvas.width - anchuraPaleta) {
368.         paletaPosX += 15;
369.     } else if (flechaIzquierdaPulsada && paletaPosX > 0) {
370.         paletaPosX -= 15;
371.     }
372.
373.     // Se incrementan los valores de la poción de las bolas
374.     x += dx;
375.     y += dy;
376.     x2 += dx2;
377.     y2 += dy2;
378.
379.     // ESTE ES UN SEGUNDO MÉTODO PARA REALIZAR LA ANIMACIÓN DEL JUEGO
380.     // LA INSTRUCCIÓN: requestAnimationFrame SE EJECUTA 60 VECES POR SEGUNDO
381.     // Y AL EJECUTARSE LLAMA A LA FUNCIÓN ENTRE PARÉNTESIS
382.     // POR TANTO, dibujar SE EJECUTA 60 VECES POR SEGUNDO
383.     // GENERANDO EL CICLO DEL JUEGO
384.     requestAnimationFrame(dibujar);
385. }
386.
387.     dibujar();
388. </script>
389. </body>
390.
391. </html>

```

Al ejecutar este código se obtiene la siguiente interfaz visual:





**Gráfica 15.** Interfaz donde se muestran los componentes del canvas.

En la gráfica anterior podemos observar, todos los componentes vistos hasta ahora, podemos observar en los costados superiores del CANVAS, las vidas y el puntaje, como las dos bolas, y si observamos bien el cursor no está presente.

Hasta ahora tenemos lo esencial del juego, pero más adelante se pondrán más funcionalidades al juego las cuales eran explicadas.

## 12 FASE 10: MOVIMIENTO DE LADRILLOS, MISIL Y EXPLOSIÓN

---

En esta fase crearemos el código para que los ladrillos aparezcan después en el margen derecho del canvas y que en cuestión de tiempo se muevan hacia la parte izquierda del mismo, además podemos usar un misil para destruir un ladrillo.

Para el movimiento de los ladrillos se debe color la posición o el vacío izquierdo de los ladrillos al máximo del ancho del canvas, si observamos la función `dibujarLadrillos()`; al final de esta, hay unas condicionales en donde se evalúan un par de variables y dependiendo de estas los ladrillos se dibujaran en diferentes posiciones, haciendo el efecto de movimiento desde un lado a otro, esto lo logramos aumentando y reduciendo el vacío izquierdo de los ladrillos.

Para la cuestión del misil se debe tener una imagen en la carpeta del programa, una variable almacenara esta imagen, además se cran las coordenadas de esta, se crea una segunda variable para almacenar la imagen de la explosión pero esta no tiene coordenadas, también se crea una función que dibujara el misil esta recibe como parámetro la coordenada X donde debe aparecer respecto al CANVAS.

Se crea una variable booleana para saber si el misil ha sido disparado, esto para evitar dos misiles a la misma vez, también la cantidad de misiles, si recordamos anteriormente creamos un evento que llama una función, `teclaPresionada()`, la utilizaremos en nuestro caso para saber si se ha pulsado la tecla espacio, de ser así se evalúa si `dispararMisil` vale `false` y si cantidad de misiles es mayor a cero, si es así entonces la posición en X de la imagen será la misma que la posición de la mitad de la paleta, en la función `dibujar()` se evalúa si `dispararMisil` es `true`, lo cual le restara la posición en el eje Y de la imagen para que el movimiento sea hacia arriba.

En la función `detectarColision()`, se evalúa si la imagen del misil toca un ladrillo, si esto es verdadero entonces se muestra la imagen de la explosión en la coordenada del ladrillo además el misil desaparece y se restablecen sus coodenadas, también se evalúa si este llega al borde superior del canvas donde sucede el mismo proceso pero obviamente no se elimina ningún ladrillo. También se dibuja la cantidad de misiles disponibles.

El código es el siguiente:

```

1. <!DOCTYPE html>
2. <html>
3.
4. <head>
5.   <meta charset="UTF-8" />
6.   <title>Juego 2D</title>
7.
8.   <style>
9.     @import url("https://fonts.googleapis.com/css2?family=Poppins&display=swap");
10.
11.     * {font-family: "Poppins", sans-serif; padding: 0;margin: 0;}
12.     h1 {font-size: 2em;text-transform: uppercase;text-align: center;}canvas {background-
13.       color: #dddddd;transition: 0.5s;display: block;margin: auto;cursor: none;}
14.     canvas.activado {background-color: #222222;}
15.   </style>
16. </head>
17. <body>
18.   <h1>Juego 2D - Fase 10</h1>
19.   <canvas id="miCanvas" width="1200" height="640"> </canvas>
20.   <script>
21.     var canvas = document.getElementById("miCanvas");
22.     var ctx = canvas.getContext("2d");
23.     var ballRadius = 10;
24.     var x = canvas.width / 2;
25.     var y = canvas.height - 30;
26.     var dx = 5;
27.     var dy = -5;
28.
29.     var x2 = x;
30.     var y2 = y;
31.     var dx2 = -1.5;
32.     var dy2 = -1.5;
33.
34.     var segundaBola = false;
35.     if (Math.random() * 100 > 50) {
36.       segundaBola = true;
37.     }
38.
39.     // Paleta
40.     var alturaPaleta = 15;
41.     var anchuraPaleta = 140;
42.     if (segundaBola) {
43.       anchuraPaleta = 180;
44.     }
45.     var paletaPosX = (canvas.width - anchuraPaleta) / 2;
46.     var flechaDerechaPulsada = false;
47.     var flechaIzquierdaPulsada = false;
48.
49.     var numFilasLadrillos = 0;
50.     var numColumnasLadrillos = 0;
51.
52.     do {
53.       numFilasLadrillos = Math.round(Math.random() * 100);
54.       numColumnasLadrillos = Math.round(Math.random() * 10);
55.     } while (numFilasLadrillos < 1 || numFilasLadrillos > 14 || numColumnasLadrillos < 1 || numColumnasLadrillos > 8);
56.
57.     var anchoLadrillo = 70;
58.     var alturaLadrillo = 20;
59.     var rellenoLadrillos = 10;
60.     var vacioSuperiorLadrillos = 50;
61.     var vacioIzquierdoLadrillos = canvas.width;
62.
63.     var puntaje = 0;
64.
65.     // Misil
66.     /* Se crean las variables para las características del misil
67.
68.     La variable imgMisil nos permitira almacenar una imagen que se encuentra en la carpeta principal
69.     creamos la posicon y del misil que sera la altura del canvas esto hace que el misil no se vea
70.     la posiocion x la inicilizamos en (0) para que mas adelante le podamos asignar un valor
71.
72.     Creamos una segunda imagen para la explosion no tiene coordenadas
73.

```

```

74.      se crea una variable booleana para saber cuando se debe visualizar el misil
75.
76.      La cantidad de misilis nos permitira ejecutarlo tres veces
77.
78.      Se crea una funcion para dibujar el misil que sera llamada luego
79.      */
80.
81.      var imgMisil = new Image();
82.      imgMisil.src = 'misil.png';
83.
84.      var posY_Misil = canvas.height;
85.      var posX_Misil = 0;
86.
87.      var explosion = new Image();
88.      explosion.src = 'explosion.png'
89.
90.      function misil(x) {
91.          ctx.drawImage(imgMisil, x, posY_Misil);
92.      }
93.
94.      var dispararMisil = false;
95.      var cantidadMisiles = 3;
96.
97.      // Caracteristicas de Los elementos
98.
99.      var colorFigura = "#ff0000";
100.     var colorBola = "#6317B9";
101.     var colorBola2 = "#2196f3";
102.     var colorPaleta = "#f5d109";
103.     var colorLadrillo = "#e61c1cdc";
104.     var colorLadrillo2 = "#e4bb06fa";
105.     var colorTexto = "#2196f3";
106.     var vidas = 3;
107.
108.     var ladrillos = [];
109.
110.     for (var columna = 0; columna < numColumnasLadrillos; columna++) {
111.         ladrillos[columna] = [];
112.         for (var fila = 0; fila < numFilasLadrillos; fila++) {
113.             ladrillos[columna][fila] = { x: 0, y: 0, estado: 10, ciclo: 10 };
114.         }
115.     }
116.
117.     document.addEventListener("keydown", teclaPresionada, false);
118.     document.addEventListener("keyup", teclaLiberada, false);
119.     document.addEventListener("mousemove", manejadorRaton, false);
120.
121.     function teclaPresionada(e) {
122.         if (e.keyCode == 39) {
123.             flechaDerechaPulsada = true;
124.         } else if (e.keyCode == 37) {
125.             flechaIzquierdaPulsada = true;
126.         } else if (e.keyCode == 32) {
127.
128.             /* Si se pulsa la tecla numero 32 que corresponde al la tecla espacio, entonces se evalua
129.             ra si disparaMisil es falso ademas y cantidad de misiles es mayor a 1, de ser asi la coordenada x del mis
130.             il sera la cordena de la paleta sumandole la mitad de la anchura de esta, esto para que el misil salga de
131.             sde la mitad de la paleta no importa donde esta este situada en el CANVAS, ademas se restan cantidadMisil
132.             es en (1) */
133.
134.             if (!dispararMisil && cantidadMisiles > 0) {
135.                 posX_Misil = paletaPosX + anchuraPaleta/2 - 20;
136.                 dispararMisil = true;
137.                 cantidadMisiles--;
138.             }
139.         }
140.     }
141.
142.     function teclaLiberada(e) {
143.         if (e.keyCode == 39) {
144.             flechaDerechaPulsada = false;
145.         } else if (e.keyCode == 37) {
146.             flechaIzquierdaPulsada = false;
147.         }
148.     }
149.
150.
151.
152.
153.
154.
155.
156.
157.
158.
159.
160.
161.
162.
163.
164.
165.
166.
167.
168.
169.
170.
171.
172.
173.
174.
175.
176.
177.
178.
179.
180.
181.
182.
183.
184.
185.
186.
187.
188.
189.
190.
191.
192.
193.
194.
195.
196.
197.
198.
199.
200.

```

```

145.     function manejarRaton(e) {
146.         var posXRatonDentroDeCanvas = e.clientX - canvas.offsetLeft;
147.         if (posXRatonDentroDeCanvas > anchuraPaleta / 2 && posXRatonDentroDeCanvas < canvas.width -
anchuraPaleta / 2) {
148.             paletaPosX = posXRatonDentroDeCanvas - anchuraPaleta / 2;
149.         }
150.     }
151.
152.     function dibujarBola(cordernaX, cordenaY, tamaño, color) {
153.         ctx.beginPath();
154.         ctx.arc(cordernaX, cordenaY, tamaño, 0, Math.PI * 2);
155.         ctx.fillStyle = color;
156.         ctx.fill();
157.         ctx.closePath();
158.     }
159.
160.     function dibujarPaleta() {
161.         ctx.beginPath();
162.         ctx.rect(paletaPosX, canvas.height - alturaPaleta, anchuraPaleta, alturaPaleta);
163.         ctx.fillStyle = colorPaleta;
164.         ctx.fill();
165.         ctx.closePath();
166.     }
167.
168.     var colorLadrillos = false;
169.
170.     /* Nos permitira mover los ladrillos mas adelante */
171.
172.     var moverDerecha = false;
173.
174.     function dibujarLadrillos() {
175.         for (var columnas = 0; columnas < numColumnasLadrillos; columnas++) {
176.             for (var filas = 0; filas < numFilasLadrillos; filas++) {
177.
178.                 if (ladrillos[columnas][filas].estado > 0) {
179.
180.                     var b = ladrillos[columnas][filas];
181.
182.                     var brickX = filas * (anchoLadrillo + rellenoLadrillos) + vacioIzquierdoLadrillos
;
183.                     var brickY = columnas * (alturaLadrillo + rellenoLadrillos) + vacioSuperiorLadril
los;
184.
185.                     if (b.estado < 10 && b.estado > 1) {
186.                         b.ciclo--;
187.                         if (b.ciclo == 0) {
188.                             if (b.estado > 1) {
189.                                 b.estado--;
190.                                 b.ciclo = 10;
191.                             }
192.                         }
193.                     }
194.
195.                     ladrillos[columnas][filas].x = brickX;
196.                     ladrillos[columnas][filas].y = brickY;
197.
198.                     ctx.beginPath();
199.                     ctx.rect(brickX, brickY, anchoLadrillo, alturaLadrillo);
200.
201.                     if (b.estado == 10) {
202.                         if (colorLadrillos) {
203.                             if (filas % 2) {
204.                                 ctx.fillStyle = colorLadrillo;
205.                             } else {
206.                                 ctx.fillStyle = colorLadrillo2;
207.                             }
208.                         } else {
209.                             if (filas % 2) {
210.                                 ctx.fillStyle = colorLadrillo2;
211.                             } else {
212.                                 ctx.fillStyle = colorLadrillo;
213.                             }
214.                         }
215.                     }
216.                     } else if (b.estado == 9) {

```

```

217.         ctx.fillStyle = "#fad638";
218.     } else if (b.estado == 8) {
219.         ctx.fillStyle = "#fbd51";
220.     } else if (b.estado == 7) {
221.         ctx.fillStyle = "#fbd51";
222.     } else if (b.estado == 6) {
223.         ctx.fillStyle = "#fce683";
224.     } else if (b.estado == 5) {
225.         ctx.fillStyle = "#fceb9c";
226.     } else if (b.estado == 4) {
227.         ctx.fillStyle = "#fdf0b4";
228.     } else if (b.estado == 3) {
229.         ctx.fillStyle = "#fef5cd";
230.     } else if (b.estado == 2) {
231.         ctx.fillStyle = "#fefa66";
232.     } else if (b.estado == 1) {
233.         ctx.fillStyle = "transparent";
234.     }
235.     ctx.fill();
236.     ctx.closePath();
237. }
238. }
239. }
240.
241.     /* Si mover deracha es falso o Las vidas son impares entonces el vacio izquierdo de Los Ladrillos se disminui
    ra hasta que esta variable tenga un valor de 50, cuando llegue a 50 la variable booleana valdra true, cuando las vidas sean pares entonces Los Ladrillos se moveran a la derecha hasta que tengan una distancia de 50 del margen derecho del CANVAS*/
242.
243.     if (!moverDerecha || vidas % 2 !== 0) {
244.         if (vacioIzquierdoLadrillos >= 50) {
245.             vacioIzquierdoLadrillos -= 10;
246.         }
247.         else {
248.             moverDerecha = true;
249.         }
250.     } else {
251.         if (vacioIzquierdoLadrillos + filas * (anchoLadrillo + rellenoLadrillos) - 50 <= canvas.width - 100) {
252.             vacioIzquierdoLadrillos += 10;
253.         }
254.     }
255. }
256.
257. // Detectar la colision
258. function detectarColision() {
259.     for (var c = 0; c < numColumnasLadrillos; c++) {
260.         for (var f = 0; f < numFilasLadrillos; f++) {
261.             var b = ladrillos[c][f];
262.             function sumarPuntaje() {
263.                 b.estado--;
264.                 b.ciclo--;
265.                 puntaje++;
266.                 if (puntaje == numFilasLadrillos * numColumnasLadrillos) {
267.                     alert("FELICITACIONES: HAS GANADO");
268.                     document.location.reload();
269.                 }
270.             }
271.             if (b.estado == 10) {
272.
273.                 // se modifica si la bola llega al ladrillo
274.                 if (x > b.x && x < b.x + anchoLadrillo + ballRadius / 2 && y > b.y && y < b.y + alturaLadrillo + ballRadius / 2) {
275.                     dy = -dy;
276.
277.                     if ((Math.round(Math.random() * 100)) % 2) {
278.                         sumarPuntaje(); // Llama a la funcion para reducir el ciclo, el estado y sumar el puntaje
279.                     }
280.                 }
281.
282.                 if (segundaBola) {
283.                     if (x2 > b.x && x2 < b.x + anchoLadrillo && y2 > b.y && y2 < b.y + alturaLadrillo) {
284.                         dy2 = -dy2;

```

```

285.
286.             if ((Math.round(Math.random() * 100)) % 2) {
287.                 sumarPuntaje(); // Llama a la función para reducir el ciclo, el estado
                y sumar el puntaje
288.             }
289.         }
290.     }
291.
292.     /* Si se dispara el misil entonces se debe evaluar si este toca un ladrillo, de s
er así entonces se mostrara en la pantalla la imagen de la explosión con las coordenadas del ladrillo, adem
as la posición y la variable bool correspondiente al misil volverán a sus estados originales para que el m
isil pueda ser lanzado de nuevo lanzado de nuevo */
293.
294.     if (posX_Misil+20 > b.x && posX_Misil+20 < b.x + anchoLadrillo && posY_Misil > b.y
&& posY_Misil < b.y + alturaLadrillo){
295.         ctx.drawImage(explosion, b.x, b.y);
296.         dispararMisil = false;
297.         posY_Misil = canvas.width/2;
298.         sumarPuntaje();
299.     }
300. }
301. }
302. }
303. }
304.
305. /* La función permitirá dibujar el puntaje y la cantidad de misiles */
306.
307. function dibujarPuntos_Misiles() {
308.     ctx.font = "24px Poppins";
309.     ctx.fillStyle = colorTexto;
310.     ctx.fillText("Puntaje: " + puntaje + "    |    Misiles: " + cantidadMisiles, 8, 20);
311. }
312.
313. function dibujarVidas() {
314.     ctx.font = "24px Poppins";
315.     ctx.fillStyle = colorTexto;
316.     // SE MUESTRA EL NÚMERO DE VIDAS DISPONIBLES
317.     ctx.fillText("Vidas: " + vidas, canvas.width - 100, 20);
318. }
319.
320. function quitarVidas() {
321.     vidas--;
322.     if (!vidas) {
323.         alert("GAME OVER");
324.         document.location.reload();
325.     } else {
326.         x = canvas.width / 2;
327.         y = canvas.height - 30;
328.         x2 = canvas.width / 2;
329.         y2 = canvas.height - 30;
330.         dx = 5;
331.         dy = -5;
332.         dx2 = -1.5;
333.         dy2 = -1.5;
334.         paletaPosX = (canvas.width - anchuraPaleta) / 2;
335.     }
336. }
337.
338. function dibujar() {
339.     ctx.clearRect(0, 0, canvas.width, canvas.height);
340.
341.     dibujarPuntos_Misiles();
342.     dibujarVidas();
343.     dibujarLadrillos();
344.     dibujarBola(x, y, ballRadius, colorBola);
345.     if (segundaBola) {
346.         dibujarBola(x2, y2, ballRadius - 2, colorBola2);
347.     }
348.     dibujarPaleta();
349.     detectarColision();
350.
351.     // Rebote de la bola
352.     if (x + dx > canvas.width - ballRadius || x + dx < ballRadius) {
353.         dx = -dx;

```

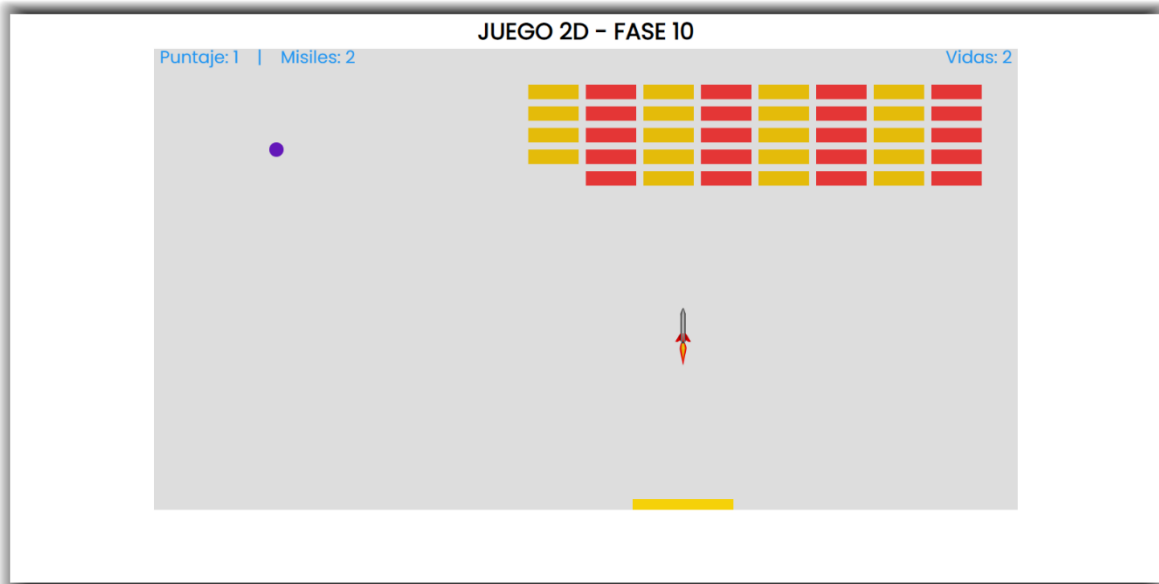
```

354.         colorLadrillos = false; // Cambia el color de los ladrillos si la bola principal rebota en
355.         x
356.         }
357.         if (y + dy < ballRadius) {
358.             dy = -dy;
359.             var cambia = document.querySelector("canvas");
360.             cambia.classList.toggle("activado");
361.             colorLadrillos = true;
362.         }
363.         if (x2 + dx2 > canvas.width - ballRadius || x2 + dx2 < ballRadius) {
364.             dx2 = -dx2;
365.         }
366.         if (y2 + dy2 < ballRadius) {
367.             dy2 = -dy2;
368.         }
369.
370.         // Rebotar en Paleta o Game Over
371.         if (y + dy > canvas.height - ballRadius) {
372.             if (x > paletaPosX && x < paletaPosX + anchuraPaleta + ballRadius / 2) {
373.                 dy = -dy;
374.             } else {
375.                 quitarVidas();
376.             }
377.         }
378.         if (segundaBola) {
379.             if (y2 + dy2 > canvas.height - ballRadius) {
380.                 if (x2 > paletaPosX && x2 < paletaPosX + anchuraPaleta) {
381.                     dy2 = -dy2;
382.                 } else {
383.                     quitarVidas();
384.                 }
385.             }
386.         }
387.
388.         // Movimiento Paleta
389.         if (flechaDerechaPulsada && paletaPosX < canvas.width - anchuraPaleta) {
390.             paletaPosX += 15;
391.         } else if (flechaIzquierdaPulsada && paletaPosX > 0) {
392.             paletaPosX -= 15;
393.         }
394.
395.         // Movimiento del misil
396.         /* Si se pulsa la tecla espacio entonces la dispararMisil vale true, por lo tanto se llama a
397.         la funcion que dibuja el misil y como parametro que le dejamos su coordena x, si recordamos cuando se pul
398.         sa la tecla espacio la posicion de x del misil varlra a la posicion de la mitad del canvas
399.
400.         Restamos Y para que la imagen se mueva hacia arriba
401.
402.         si la imagen alcanza la parte superior del CANVAS entonces se dibuja la explosion y se actual
403.         izan las variables del misil para que pueda ser disparado de nuevo
404.         */
405.
406.         if(dispararMisil){
407.             misil(posX_Misil);
408.             posY_Misil -= 5;
409.         }
410.         if(posY_Misil == 0){
411.             ctx.drawImage(explosion, posX_Misil-50, posY_Misil)
412.             dispararMisil = false;
413.             posY_Misil = canvas.height;
414.         }
415.
416.         // Se incrementan los valores de la pocion de las bolas
417.         x += dx;
418.         y += dy;
419.         x2 += dx2;
420.         y2 += dy2;
421.
422.         requestAnimationFrame(dibujar);
423.     }
424.     dibujar();
425. </script>
426. </body>
427. </html>

```



El resultado es el siguiente:



Gráfica 16. Interfaz cuando se presiona la tecla espacio.



Gráfica 17. Interfaz cuando el misil golpea un ladrillo.



## 13 FASE 11: FIN, MENÚ, GANAR Y PERDER

---

En este apartado finalizaremos nuestro juego, si recordamos nuestro juego es al azar, o sea no sabemos cuantos ladrillos nos van a salir en el principio, como también dos bolas o solo una, pero en esta fase haremos un menu donde el jugador pueda escoger las caracteristicas del juego, como cambiar el color del canvas, el tamaño de las bolas, número de coetes y vidas etc.

Para esto agregamos nuevo elementos en el código HTML como el `<input type="text">` que permitirá al usuario introducir datos por medio del teclado como también el uso de botones, si nos fijamos en esto elementos podemos observar que cada uno tiene un identificador lo cual lo utilizaremos en CSS y en JavaScript para poder tener los datos introducidos. Podemos ver que hay tres etiquetas div que nos permite almacenar estos elementos, el primero es la interfaz inicial este tendrá una imagen y el menú para las condiguraciones, sin embargo el canvas no se vera, también tenemos otros dos div que los utilizaremos cuando se gane o se pierda el juego, estos contiene un archivo gif.

El botón JUGAR llamara una función llamada `jugar()`, que mas adelante explicaremos.

En el código CSS hacemos todas las configuraciones y como podemos observar el canvas no se ve ya que lo movemos a la izquierda esto se consigue con `left:-100%`, esto permite no verse en la página, también lo hacemos con los otros elementos. En el código JavaScript veremos como se mueven los lementos esto combinado con CSS.

En el código JavaScript creamos tres función, la primera `valor(documeto)`; nos permitira obtener el valor del elemento cuyo id concida con el id de un elemento en el código HTML, esto para obtener el código mas limpio y tener mas facilidad a la hora de obtener el valor de los diferentes elementos, la segunda `comprobarColor()`; nos permitirá saber si el texto introducido esta en el campo de los colores RGB ya que este tipo de colores comienza con un (#) y debe tener 4 o 7 caracteres como (`#fff`) o (`#ffffff`) que son el mismo color, el par de condicionales es para saber a que variable le pertence lo introducido por el usuario, la tercera `jugar()`; nos permitirá evitar cargar la pagina cuando pulsemos en el botón JUGAR, además llamamos a las funciones anteriores, también tenemos las condiconales para obtener el valor de las otras caracteristicas estas son el número de ladrillos, el tamaño de las bolas etc. Por ultimo le damos valor a una variable que permite ejecutar la animación como también llamamos a la función `dibujar`, esto para que el canvas no se dibuje en la sección de las configuraciones, como también cuando se gana o se pierde.

El primer código es el de HTML, el segundo es el de CSS y tercero es el de JavaScript.

```

1. <!DOCTYPE html>
2. <html>
3.
4. <head>
5.   <meta charset="UTF-8" />
6.   <title>Juego 2D</title>
7.
8.   <link rel="stylesheet" href="style.css">
9. </head>
10.
11. <body>
12.   <div class="inicio">
13.     <div class="menu">
14.       <h2>Configuraciones</h2>
15.       <form action="">
16.         <ul>
17.           <li>
18.             Color del Texto:
19.             <input id="cT" type="text" placeholder="#2196f3" />
20.           </li>
21.         </ul>
22.         <ul>
23.           <li>
24.             Color del canvas No.1:
25.             <input id="cC1" type="text" placeholder="#b2b2b2" />
26.           </li>
27.           <li>
28.             Color del canvas No.2:
29.             <input id="cC2" type="text" placeholder="#222222" />
30.           </li>
31.         </ul>
32.         <ul>
33.           <li>
34.             Color de la bola No.1:
35.             <input id="cB1" type="text" placeholder="#6317B9" />
36.           </li>
37.           <li>
38.             Color de la bola No.2:
39.             <input id="cB2" type="text" placeholder="#2196f3" />
40.           </li>
41.         </ul>
42.         <ul>
43.           <li>
44.             Tamaño de la bola No.1:
45.             <input id="tB1" type="text" placeholder="10" />
46.           </li>
47.           <li>
48.             Tamaño de la bola No.2:
49.             <input id="tB2" type="text" placeholder="8" />
50.           </li>
51.         </ul>
52.         <ul>
53.           <li>
54.             Número de filas:
55.             <input id="numFilas" type="text" placeholder="Menor a 15" />
56.           </li>
57.           <li>
58.             Número de columnas:
59.             <input id="numColumnas" type="text" placeholder="Menor a 9" />
60.           </li>
61.         </ul>
62.         <ul>
63.           <li>
64.             Color No.1 de ladrillos:
65.             <input id="cL1" type="text" placeholder="#e61c1cdc" />
66.           </li>
67.           <li>
68.             Color No.2 de ladrillos:
69.             <input id="cL2" type="text" placeholder="#e4bb06fa" />
70.           </li>
71.         </ul>
72.         <ul>

```

```

73.         <li>
74.             Jugar con 2 bolas:
75.             <input id="check" type="checkbox" value="2" />
76.         </li>
77.         <li>
78.             Cantidad de vidas:
79.             <input id="vidas" type="text" placeholder="3" />
80.         </li>
81.     </ul>
82.     <ul>
83.         <li>
84.             Cantidad de misiles:
85.             <input id="misiles" type="text" placeholder="3" />
86.         </li>
87.         <li>
88.             Tamaño de la Paleta:
89.             <input id="paleta" type="text" placeholder="100" />
90.         </li>
91.     </ul>
92.     <ul>
93.         <li>
94.             <input id="boton" type="submit" value="JUGAR" onclick="jugar()" />
95.         </li>
96.     </ul>
97. </form>
98. </div>
99. <div class="presentacion">
100.     <h2>Juego 2D JavaScript</h2>
101.     
102.     <h3>Usar Misil: tecla Espacio</h3>
103.     <p>
104.         Al comenzar el juego debe dar click, de lo contrario al darle espacio
105.         se mostrara esta sección
106.     </p>
107. </div>
108. </div>
109.
110. <canvas id="miCanvas" width="1200" height="640"> </canvas>
111.
112. <div class="ganar">
113.     
114.     <input class="volVer" type="submit" value="VOLVER" onclick="document.Location.reload()"/>
115. </div>
116.
117. <div class="perder">
118.     
119.     <input class="volVer" type="submit" value="VOLVER" onclick="document.Location.reload()"/>
120. </div>
121.
122. <script src="script.js"></script>
123. </body>
124. </html>

```

```

105. @import url("https://fonts.googleapis.com/css2?family=Poppins&display=swap");
106.
107. * {
108.     font-family: "Poppins", sans-serif;
109.     padding: 0;
110.     margin: 0;
111. }
112.
113. canvas {
114.     transition: 0.5s;
115.     display: block;
116.     cursor: none;
117.     position: fixed;
118.     right: -100%;
119.     transition: 1s;
120. }
121.
122. canvas.moverCanvas {
123.     position: relative;
124.     right: 0;

```

```

125.   margin: 50px auto 0 auto;
126. }
127.
128. /* Flex nos permite configurar el tipo de contenedor del elemento
129.
130. justify-content: space-between; nos permite poner un elemento junto al otro
131.
132. align-items: center; nos permite centrar el contenido
133.
134. */
135.
136. .inicio {
137.   width: 100%;
138.   height: 100vh;
139.   display: flex;
140.   position: fixed;
141.   justify-content: space-between;
142.   align-items: center;
143.   text-align: center;
144.   left: 0;
145.   transition: 1s;
146. }
147.
148. .inicio .menu {
149.   display: block;
150.   width: 50%;
151.   height: 100vh;
152. }
153.
154. .inicio .menu form {
155.   display: block;
156.   margin: 0 auto;
157. }
158.
159. .inicio form ul {
160.   list-style: none;
161.   width: 50%;
162.   margin: auto;
163. }
164.
165. .inicio form ul li {
166.   padding: 7px;
167.   justify-content: space-between;
168.   display: flex;
169. }
170.
171. .inicio form ul li input[type="text"] {
172.   border: none;
173.   padding: 0 5px;
174.   border-bottom: solid 2px #111111;
175. }
176.
177. #boton, .volver{
178.   background-color: #2196f3;
179.   color: #ffffff;
180.   display: block;
181.   font-weight: 600;
182.   border: none;
183.   padding: 10px;
184.   margin: auto;
185.   width: 150px;
186. }
187.
188. /* Hover nos permite modificar un elemento cuando el curso se encuentra encima del elemento */
189.
190. #boton:hover, .volver:hover {
191.   background-color: #f32f21d2;
192.   cursor: pointer;
193. }
194.
195. .inicio .presentacion {
196.   display: block;
197.   margin: auto;
198.   width: 50%;
199.   height: 100vh;

```



```
200.   text-align: center;
201.   color: #111111;
202. }
203.
204. .inicio .presentacion h3 {
205.   border-bottom: solid 2px #111111;
206.   display: inline-block;
207. }
208.
209. .inicio .presentacion img {
210.   width: 80%;
211.   padding: 100px 0;
212. }
213.
214. /* .mover es una clase que nos permitira mover el elemento esto lo logramos con javascript */
215.
216. .inicio.mover {
217.   left: -100%;
218. }
219.
220. .ganar, .perder {
221.   width: 100%;
222.   height: 100vh;
223.   background-color: #ffffff;
224.   display: block;
225.   margin: auto;
226.   position: fixed;
227.   left: -100%;
228.   transition: 1s;
229. }
230.
231. .ganar.mover, .perder.mover {
232.   left: 0;
233. }
234.
235. .ganar img, .perder img{
236.   display: block;
237.   margin: 100px auto 0 auto;
238.   box-shadow: 20px 20px 10px #333333;
239. }
240.
241. .volver{
242.   margin-top: 30px;
243. }
```

```
244. var canvas = document.getElementById("miCanvas");
245. var ctx = canvas.getContext("2d");
246. var ballRadius = 10;
247. var x = canvas.width / 2;
248. var y = canvas.height - 30;
249. var dx = 5;
250. var dy = -5;
251.
252. var ballRadius2 = 8;
253. var x2 = x;
254. var y2 = y;
255. var dx2 = -1.5;
256. var dy2 = -1.5;
257.
258. var segundaBola = false;
259. if (Math.random() * 100 > 50) {
260.   segundaBola = true;
261. }
262.
263. // Paleta
264. var alturaPaleta = 15;
265. var anchuraPaleta = 140;
266. if (segundaBola) {
267.   anchuraPaleta = 180;
268. }
269. var paletaPosX = (canvas.width - anchuraPaleta) / 2;
270. var flechaDerechaPulsada = false;
271. var flechaIzquierdaPulsada = false;
```

```

272.
273. var numFilasLadrillos = 0;
274. var numColumnasLadrillos = 0;
275.
276. do {
277.     numFilasLadrillos = Math.round(Math.random() * 100);
278.     numColumnasLadrillos = Math.round(Math.random() * 10);
279. } while (numFilasLadrillos < 1 || numFilasLadrillos > 14 || numColumnasLadrillos < 1 || numColumnasLadrillos > 8);
280.
281. var anchoLadrillo = 70;
282. var alturaLadrillo = 20;
283. var rellenoLadrillos = 10;
284. var vacioSuperiorLadrillos = 50;
285. var vacioIzquierdoLadrillos = canvas.width;
286.
287. var puntaje = 0;
288.
289. // Misil
290. var imgMisil = new Image();
291. imgMisil.src = "img/misil.png";
292.
293. var posY_Misil = canvas.height;
294. var posX_Misil = 0;
295.
296. var explosion = new Image();
297. explosion.src = "img/explosion.png";
298.
299. function misil(x) {
300.     ctx.drawImage(imgMisil, x, posY_Misil);
301. }
302.
303. // Características de los elementos
304. var dispararMisil = false;
305. var cantidadMisiles = 3;
306.
307. var colorCanvas1 = "#f2f2f2";
308. canvas.style.backgroundColor = colorCanvas1;
309. var colorCanvas2 = "#222222";
310. var colorFigura = "#ff0000";
311. var colorBola = "#631789";
312. var colorBola2 = "#2196f3";
313. var colorPaleta = "#f5d109";
314. var colorLadrillo = "#e61c1cdc";
315. var colorLadrillo2 = "#e4bb06fa";
316. var colorTexto = "#2196f3";
317. var vidas = 3;
318.
319. var ladrillos = [];
320.
321. /* Convertimos poner ladrillos en una función ya que si el usuario introduce una cantidad de filas y columnas estas se puedan dibujar ya que si dejamos solo los bucles estos ponen un número de filas y columnas al azar, lo cual al usuario al poner un gusto de estos el programa no podrá ponerlos */
322.
323. function ponerLadrillos(){
324.     for (var columna = 0; columna < numColumnasLadrillos; columna++) {
325.         ladrillos[columna] = [];
326.         for (var fila = 0; fila < numFilasLadrillos; fila++) {
327.             ladrillos[columna][fila] = { x: 0, y: 0, estado: 10, ciclo: 10 };
328.         }
329.     }
330. }
331.
332. document.addEventListener("keydown", teclaPresionada, false);
333. document.addEventListener("keyup", teclaLiberada, false);
334. document.addEventListener("mousemove", manejadorRaton, false);
335.
336. function teclaPresionada(e) {
337.     if (e.keyCode == 39) {
338.         flechaDerechaPulsada = true;
339.     } else if (e.keyCode == 37) {
340.         flechaIzquierdaPulsada = true;
341.     } else if (e.keyCode == 32) {
342.         if (!dispararMisil && cantidadMisiles > 0) {
343.             posX_Misil = paletaPosX + anchuraPaleta / 2 - 20;

```

```

344.         dispararMisil = true;
345.         cantidadMisiles--;
346.     }
347. }
348. }
349. function teclaLiberada(e) {
350.     if (e.keyCode == 39) {
351.         flechaDerechaPulsada = false;
352.     } else if (e.keyCode == 37) {
353.         flechaIzquierdaPulsada = false;
354.     }
355. }
356.
357. function manejadorRaton(e) {
358.     var posXRatonDentroDeCanvas = e.clientX - canvas.offsetLeft;
359.     if (posXRatonDentroDeCanvas > anchuraPaleta / 2 && posXRatonDentroDeCanvas < canvas.width -
        anchuraPaleta / 2) {
360.         paletaPosX = posXRatonDentroDeCanvas - anchuraPaleta / 2;
361.     }
362. }
363.
364. function dibujarBola(cordenaX, cordenaY, tamaño, color) {
365.     ctx.beginPath();
366.     ctx.arc(cordenaX, cordenaY, tamaño, 0, Math.PI * 2);
367.     ctx.fillStyle = color;
368.     ctx.fill();
369.     ctx.closePath();
370. }
371.
372. function dibujarPaleta() {
373.     ctx.beginPath();
374.     ctx.rect(paletaPosX, canvas.height - alturaPaleta, anchuraPaleta, alturaPaleta);
375.     ctx.fillStyle = colorPaleta;
376.     ctx.fill();
377.     ctx.closePath();
378. }
379.
380. var colorLadrillos = false;
381.
382. var moverDerecha = false;
383.
384. function dibujarLadrillos() {
385.     for (var columnas = 0; columnas < numColumnasLadrillos; columnas++) {
386.         for (var filas = 0; filas < numFilasLadrillos; filas++) {
387.             if (ladrillos[columnas][filas].estado > 0) {
388.                 var b = ladrillos[columnas][filas];
389.
390.                 var brickX = filas * (ancholLadrillo + rellenoLadrillos) + vacioIzquierdoLadrillos;
391.                 var brickY = columnas * (alturaLadrillo + rellenoLadrillos) + vacioSuperiorLadrillos;
392.
393.                 if (b.estado < 10 && b.estado > 1) {
394.                     b.ciclo--;
395.                     if (b.ciclo == 0) {
396.                         if (b.estado > 1) {
397.                             b.estado--;
398.                             b.ciclo = 10;
399.                         }
400.                     }
401.                 }
402.
403.                 ladrillos[columnas][filas].x = brickX;
404.                 ladrillos[columnas][filas].y = brickY;
405.
406.                 ctx.beginPath();
407.                 ctx.rect(brickX, brickY, ancholLadrillo, alturaLadrillo);
408.
409.                 if (b.estado == 10) {
410.                     if (colorLadrillos) {
411.                         if (filas % 2) {
412.                             ctx.fillStyle = colorLadrillo;
413.                         } else {
414.                             ctx.fillStyle = colorLadrillo2;
415.                         }
416.                     } else {
417.                         if (filas % 2) {

```



```

418.         ctx.fillStyle = colorLadrillo2;
419.     } else {
420.         ctx.fillStyle = colorLadrillo;
421.     }
422. }
423. } else if (b.estado == 9) {
424.     ctx.fillStyle = "#fad638";
425. } else if (b.estado == 8) {
426.     ctx.fillStyle = "#fbd551";
427. } else if (b.estado == 7) {
428.     ctx.fillStyle = "#fbd551";
429. } else if (b.estado == 6) {
430.     ctx.fillStyle = "#fce683";
431. } else if (b.estado == 5) {
432.     ctx.fillStyle = "#fce69c";
433. } else if (b.estado == 4) {
434.     ctx.fillStyle = "#fdf0b4";
435. } else if (b.estado == 3) {
436.     ctx.fillStyle = "#fef5cd";
437. } else if (b.estado == 2) {
438.     ctx.fillStyle = "#fe6ae6";
439. } else if (b.estado == 1) {
440.     ctx.fillStyle = "transparent";
441. }
442. ctx.fill();
443. ctx.closePath();
444. }
445. }
446. }
447.
448. if (!moverDerecha || vidas % 2 !== 0) {
449.     if (vacioIzquierdoLadrillos >= 50) {
450.         vacioIzquierdoLadrillos -= 10;
451.     } else {
452.         moverDerecha = true;
453.     }
454. } else {
455.     if (vacioIzquierdoLadrillos + filas * (anchoLadrillo + rellenoLadrillos) - 50 <= canvas.width -
100) {
456.         vacioIzquierdoLadrillos += 10;
457.     }
458. }
459. }
460.
461. // Detectar la colision
462. function detectarColision() {
463.     for (var c = 0; c < numColumnasLadrillos; c++) {
464.         for (var f = 0; f < numFilasLadrillos; f++) {
465.             var b = ladrillos[c][f];
466.             function sumarPuntaje() {
467.                 b.estado--;
468.                 b.ciclo--;
469.                 puntaje++;
470.                 if (puntaje == numFilasLadrillos * numColumnasLadrillos) {
471.                     /* Si se gana entonces agregamos una clase al canvas esto para moverlo, o desaparece
a de la pagina, pero agregamos una clase llamada (mover) al elemento que tiene una otra clase definida ll
amada "ganar" esto para que cuando se active la mover permita mostrar este elemento en la pagina y clickJ
ugar se vuelve false para que no se siga dibujando el canvas */
472.                     document.getElementById("miCanvas").classList.toggle("moverCanvas");
473.                     document.querySelector(".ganar").classList.toggle("mover");
474.                     clickJugar = false;
475.                 }
476.             }
477.         }
478.     }
479.     if (b.estado == 10) {
480.         // se modifica si la bola llega al ladrillo
481.         if (x > b.x && x < b.x + anchoLadrillo + ballRadius / 2 && y > b.y && y < b.y + alturaLad
rillo + ballRadius / 2) {
482.             dy = -dy;
483.             if (Math.round(Math.random() * 100) % 2) {
484.                 sumarPuntaje();
485.             }
486.         }
487.     }

```

```

488.         if (segundaBola) {
489.             if (x2 > b.x && x2 < b.x + anchoLadrillo && y2 > b.y && y2 < b.y + alturaLadrillo) {
490.                 dy2 = -dy2;
491.
492.                 if (Math.round(Math.random() * 100) % 2) {
493.                     sumarPuntaje();
494.                 }
495.             }
496.         }
497.         if (posX_Misil + 20 > b.x && posX_Misil + 20 < b.x + anchoLadrillo && posY_Misil > b.y &&
posY_Misil < b.y + alturaLadrillo) {
498.             ctx.drawImage(explosion, b.x, b.y);
499.             dispararMisil = false;
500.             posY_Misil = canvas.width / 2;
501.             sumarPuntaje();
502.         }
503.     }
504. }
505. }
506. }
507.
508. function dibujarPuntos_Misiles() {
509.     ctx.font = "24px Poppins";
510.     ctx.fillStyle = colorTexto;
511.     ctx.fillText("Puntaje: " + puntaje + "      /      Misiles: " + cantidadMisiles, 8, 20 );
512. }
513.
514. function dibujarVidas() {
515.     ctx.font = "24px Poppins";
516.     ctx.fillStyle = colorTexto;
517.     ctx.fillText("Vidas: " + vidas, canvas.width - 100, 20);
518. }
519.
520. function quitarVidas() {
521.     vidas--;
522.     if (!vidas) {
523.
524.         /* Si el numero de vidas llega a (0) cero, entonces debemos mostrar elemento que corresponde a pe
rde, ocultamos nuestro canvas con ayuda de CSS y mostramos esta seccion ademas detenemos el dibujo del c
anvas */
525.
526.         document.getElementById("miCanvas").classList.toggle("moverCanvas");
527.         document.querySelector(".perder").classList.toggle("mover");
528.         clickJugar = false;
529.     } else {
530.         x = canvas.width / 2;
531.         y = canvas.height - 30;
532.         x2 = canvas.width / 2;
533.         y2 = canvas.height - 30;
534.         dx = 5;
535.         dy = -5;
536.         dx2 = -1.5;
537.         dy2 = -1.5;
538.         paletaPosX = (canvas.width - anchuraPaleta) / 2;
539.     }
540. }
541.
542. /* Como tenemos varios elementos tipo texto los cuales debemos conocer su valor y estos tiene como identi
ficador un id entonces creamos una funcion para no escribir mas de diez veces document.getElementById(do
cumento).value; */
543.
544. function valor(documento) {
545.     var entregarValor = document.getElementById(documento);
546.     return entregarValor.value;
547. }
548.
549. /* como debemos comprobar el color o el texto que el usuario introduzca o si no generara un error, entonc
es se debe verificar lo que el usuario a introducido.
550. 547.
551. 548.     Tenemos que verificar si el color es RGB lo cual este tipo de colores tiene como predeterminado c
omenzar con un numeral (#) lo cual es obligatorio que sea el primer caracter de lo introducido tambien es
te debe tener 4 o 7 caracteres ejemplo (#111) o (#111111) son el mismo color lo cual uno tiene 4 y el otr
o tiene 7 caracteres
552. 549.

```

```

553. 550.    si el color es verdadero, debemos ver el id de este del texto que corresponde a un elemento HTML
           y verificamos dependiendo de esto almacenamos lo introducido por el usuario en uno de los tipos de elemnt
           os en el canvas, o mejor dicho lo almacenamos en una variable, como color de canvas, color de bola etc.
554. 551.
555. 552.    */
556.
557. function comprobarColor(elemento, variable) {
558.     if ((valor(elemento).length == 4 || valor(elemento).length == 7) && valor(elemento).charAt(0) == "#")
559.     {
560.         if (elemento === "cT") {
561.             colorTexto = valor(elemento);
562.         } else if (elemento === "cC1") {
563.             colorCanvas1 = valor(elemento);
564.         } else if (elemento === "cC2") {
565.             colorCanvas2 = valor(elemento);
566.         } else if (elemento === "cB1") {
567.             colorBola = valor(elemento);
568.         } else if (elemento === "cB2") {
569.             colorBola2 = valor(elemento);
570.         } else if (elemento === "cL1") {
571.             colorLadrillo = valor(elemento);
572.         } else if (elemento === "cL2") {
573.             colorLadrillo2 = valor(elemento);
574.         } else if (elemento === "cP") {
575.             colorPaleta = valor(elemento);
576.         }
577.     } else if (valor(elemento).length !== 0) {
578.         alert("Introduce bien los valores.");
579.     }
580. }
581. /* Creamos esta variable ya que si el se gana el canvas se seguira dibujando lo cual no queremos que pase
           eso, ya que como el canvas se seguira dibujando seguira perdiendo vidas por consiguiente se mostrara la v
           entana de ganar y perder consecutivamente */
582.
583. var clickJugar = false;
584.
585. /* En la siguiente funcion nos permitira comenzar el juego
586. 581.
587. 582.    primero al darle click a un boton este por lo general recarga nuevamente la pagina lo cual la pr
           imera instruccion es para evitar esto event.preventDefault();
588. 583.
589. 584.    Lo siguiente es mostrar el canvas lo cual al elemnto cuya clase es inicio le agrega una segunda c
           lase llamada mover, tambien pasa con el canvas, esto para permitir mover las secciones con CSS
590. 585.
591. 586.    Debemos comprobar los colores como los de los ladrillos etc, lo cual llamamos la funcion definida
           anteriormente
592. 587.
593. 588.    Ahora debemos comprobar el valor para las características de la bola, la cantidad de ladrillos et
           c. Como debemos poner datos numericos debemos convertir el texto en numeros, esto lo logramos con parseFl
           oat y parseInt, y así dependiendo del elemento donde se introdujo el texto se agregara a las característi
           cas o variables que definimo al principio del código numfilasLadrillos, ballradius etc
594. 589.    */
595.
596. function jugar() {
597.     event.preventDefault();
598.     document.querySelector(".inicio").classList.toggle("mover");
599.     document.getElementById("miCanvas").classList.toggle("moverCanvas");
600.
601.     document.querySelector("body").style.overflow = 'hidden';
602.     window.location.href = '#arriba';
603.
604.     comprobarColor("cT");
605.     comprobarColor("cC1");
606.     comprobarColor("cC2");
607.     comprobarColor("cB1");
608.     comprobarColor("cB2");
609.     comprobarColor("cL1");
610.     comprobarColor("cL2");
611.     comprobarColor("cP");
612.
613.     if (parseFloat(valor("tB1")) > 0 && parseFloat(valor("tB1")) <= 40) {
614.         ballRadius = parseFloat(valor("tB1"));
615.     }
616.

```

```

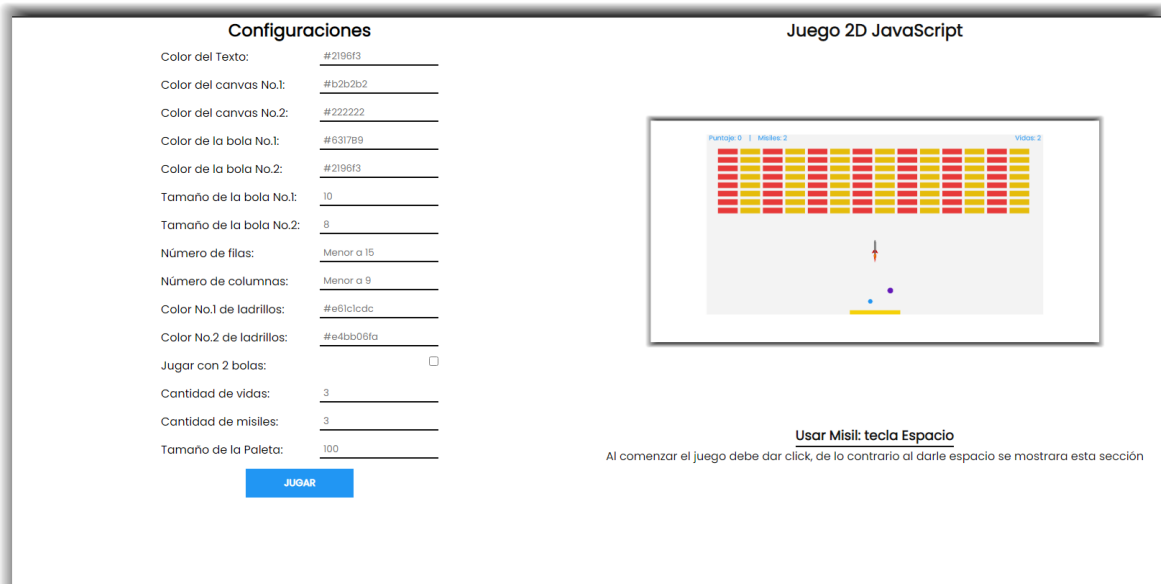
617.     if (parseFloat(valor("tB2")) > 0 && parseFloat(valor("tB2")) <= 40) {
618.         ballRadius2 = parseFloat(valor("tB2"));
619.     }
620.
621.     if (parseInt(valor("numFilas")) > 0 && parseInt(valor("numFilas")) <= 14) {
622.         numFilasLadrillos = parseInt(valor("numFilas"));
623.     }
624.
625.     if (parseInt(valor("numColumnas")) > 0 && parseInt(valor("numColumnas")) <= 8) {
626.         numColumnasLadrillos = parseInt(valor("numColumnas"));
627.     }
628.
629.     /* Como tenemos un checkBox debemos saber si este se encuentra seleccionado para ver si dibujamos una
segunda bola */
630.
631.     if (document.getElementById("check").checked) {
632.         segundaBola = true;
633.     }
634.
635.     if (parseInt(valor("vidas")) > 0) {
636.         vidas = parseInt(valor("vidas"));
637.     }
638.
639.     if (parseInt(valor("misiles")) > 0) {
640.         cantidadMisiles = parseInt(valor("misiles"));
641.     }
642.
643.     if (parseFloat(valor("paleta")) > 0) {
644.         anchuraPaleta = parseFloat(valor("paleta"));
645.     }
646.
647.     /* Llamamos poner Ladrillos esto para evitar que el programa se dañe ya que si lo ponemos en dibujar(
) Los ladrillos siempre se dibujaran por lo tanto nunca desapareceran */
648.
649.     ponerLadrillos();
650.
651.     /* Cambiamos el valor de la variable y llamamos a la función dibujar() para que comience el juego */
652.
653.     clickJugar = true;
654.
655.     dibujar();
656. }
657.
658. function dibujar() {
659.     ctx.clearRect(0, 0, canvas.width, canvas.height);
660.
661.     dibujarPuntos_Misiles();
662.     dibujarVidas();
663.     dibujarLadrillos();
664.     dibujarBola(x, y, ballRadius, colorBola);
665.     if (segundaBola) {
666.         dibujarBola(x2, y2, ballRadius2, colorBola2);
667.     }
668.     dibujarPaleta();
669.     detectarColision();
670.
671.     // Rebote de la bola
672.     if (x + dx > canvas.width - ballRadius || x + dx < ballRadius) {
673.         dx = -dx;
674.         colorLadrillos = false; // Cambia el color de los ladrillos si la bola principal rebota en x
675.     }
676.     if (y + dy < ballRadius) {
677.         dy = -dy;
678.         canvas.style.backgroundColor = colorCanvas2;
679.         colorLadrillos = true;
680.     }
681.
682.     if (x2 + dx2 > canvas.width - ballRadius2 || x2 + dx2 < ballRadius2) {
683.         dx2 = -dx2;
684.     }
685.     if (y2 + dy2 < ballRadius2) {
686.         dy2 = -dy2;
687.     }
688.
689.     // Rebotar en Paleta o Game Over

```



```
690.     if (y + dy > canvas.height - ballRadius) {
691.         if (x > paletaPosX && x < paletaPosX + anchuraPaleta + ballRadius / 2) {
692.             dy = -dy;
693.             canvas.style.backgroundColor = colorCanvas1;
694.         } else {
695.             quitarVidas();
696.         }
697.     }
698.     if (segundaBola) {
699.         if (y2 + dy2 > canvas.height - ballRadius2) {
700.             if (x2 > paletaPosX && x2 < paletaPosX + anchuraPaleta) {
701.                 dy2 = -dy2;
702.             } else {
703.                 quitarVidas();
704.             }
705.         }
706.     }
707.
708.     // Movimiento Paleta
709.     if (flechaDerechaPulsada && paletaPosX < canvas.width - anchuraPaleta) {
710.         paletaPosX += 15;
711.     } else if (flechaIzquierdaPulsada && paletaPosX > 0) {
712.         paletaPosX -= 15;
713.     }
714.
715.     // Movimiento del misil
716.     if (dispararMisil) {
717.         misil(posX_Misil);
718.         posY_Misil -= 5;
719.     }
720.     if (posY_Misil == 0) {
721.         ctx.drawImage(explosion, posX_Misil - 50, posY_Misil);
722.         dispararMisil = false;
723.         posY_Misil = canvas.height;
724.     }
725.
726.     // Se incrementan los valores de la poción de las bolas
727.     x += dx;
728.     y += dy;
729.     x2 += dx2;
730.     y2 += dy2;
731.
732.     /* Si clickJugar es true entonces el requestAnimationFrame comenzara a ejecutar la funcion dibujar */
733.     if(clickJugar){
734.         requestAnimationFrame(dibujar);
735.     }
736. }
```

El resultado es el siguiente:

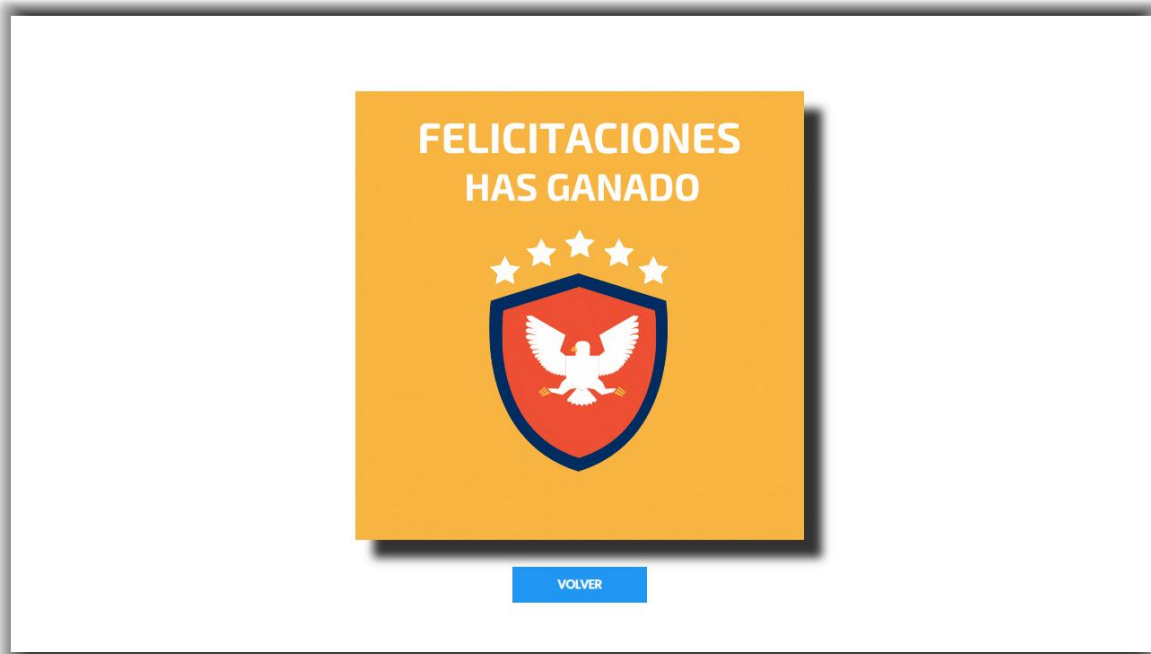


Gráfica 18. Interfaz inicial de la página.

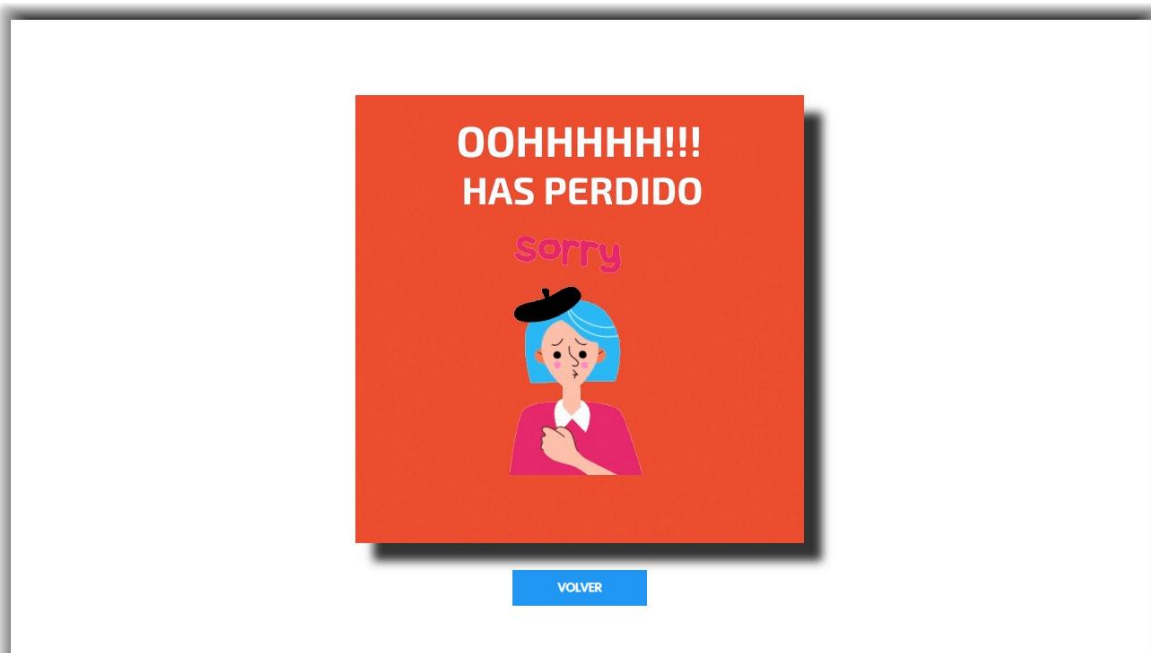
En la gráfica anterior podemos observar el resultado tras cargar la página el menu que creamos con los elementos HTML y los estilos en CSS.



Gráfica 19. Interfaz cuando le damos al botón JUGAR.



Gráfica 20. Interfaz cuando eliminamos todos los ladrillos.



Gráfica 21. Interfaz cuando agotamos todas las vidas.



En las dos ultimas graficas observamos el botón VOLVER este permitirá racargar la página lo cual obtenemos lo mostrado en la gráfica 18.

Puedes ejecutar esta última fase en: <https://repl.it/@JuanDiegoDieg90/Juego-2D-JavaScript>

Puedes encontrar las fases en: <https://github.com/JuanDiegoVelez/Juego-2D-Monografia-y-fases-del-juego.git>





## 14 CONCLUSIONES

---

En conclusiones podemos observar que gracias a los códigos combinados en HTML, CSS y JavaScript podemos obtener un juego 2D totalmente en la web.

Después de seguir ciertas cantidades de pasos como añadir imágenes, crear las variables y funciones, como también los estilos, obtenemos el resultado final, en todo esto aprendemos lo especial de cada lenguaje de la web para construir cualquier programa, al aprender el lenguaje de programación JavaScript nos brindará una forma de pensar, una lógica, un estilo y los conocimientos necesarios para implementarlos en diferentes softwares y en otros lenguajes de programación.



## 15 BIBLIOGRAFÍA

---

[https://developer.mozilla.org/es/docs/Games/Workflows/Famoso\\_juego\\_2D\\_usando\\_JavaScript\\_puro/Construye\\_grupo\\_bloques](https://developer.mozilla.org/es/docs/Games/Workflows/Famoso_juego_2D_usando_JavaScript_puro/Construye_grupo_bloques)