2020

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

MARIANA HENAO MORALES
LAURA QUINTERO MONTOYA

14 DE SEPTIEMBRE DE 2020

UTP | Pereira

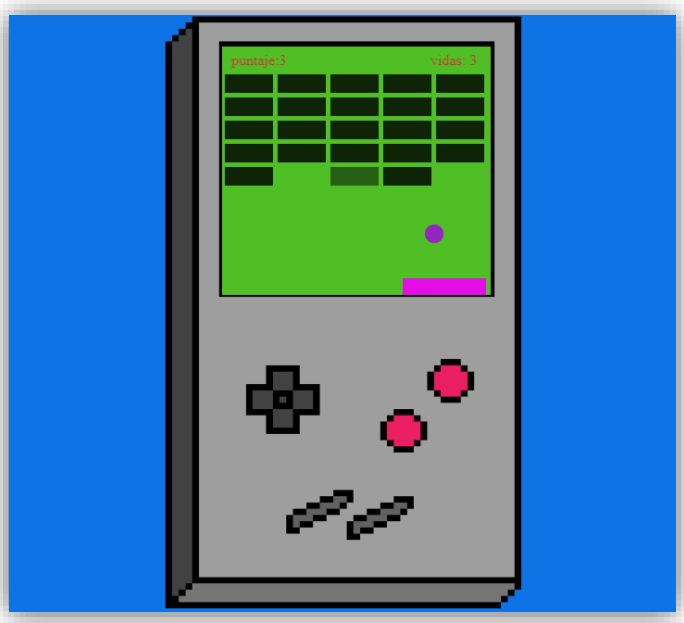
1 CONTENIDO

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 una orientación metódica 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

AUTORES: MARIANA HENAO MORALES Y LAURA QUINTERO MONTOYA

3 FASE 1: DIBUJAR Y MOVER UNA BOLA

El primer paso consiste en elaborar una página HTML básica. Agregaremos a dicha página un elemento CANVAS, el cual nos servirá como base para el desarrollo del juego 2D.

Dándole un estilo atractivo a la vista a este proyecto, se diseñará el fondo de la página. Si queremos centrar una imagen que deseamos que haga arte del fondo, colocamos la etiqueta de center, después la etiqueta de imagen junto con sus dimensiones. Además, para cambiar el color de la pantalla, dentro de la etiqueta de "body" colocamos (style="background-color) enseguida, ingresamos el código del color que queremos que sea el fondo de nuestra página y cerramos con punto y coma, (el body no se cierra hasta terminar todo el código).

Después, tenemos la necesidad de crear el CANVAS, para que allí se desarrolle el dibujo y la animación de la bola; donde en su style, incluimos varias variables para que se centre el lienzo, entre ellos, se agrega adicionalmente la posición, top, bottom, left y right, para obtener el CANVAS en el centro de la página.

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

La correcta visualización y procedimiento del CANVAS requiere de la adición de algunas características de estilo. Una vez hecho esto, se procede a establecer la codificación pertinente del JavaScript. Debe notarse la inclusión de algunas variables que definen la funcionalidad del juego en sus aspectos básicos: las coordenadas en las que se encuentra la bola (con las etiquetas de var x = canvas.width /num; y var y = canvas.height num;) y los valores de incremento para modificar su posición (con las variables) var dx = num; y var dy = num;)

Se definen tres funciones importantes. La primera de ellas, dibujarBola(), se encarga de dibujar sobre la pantalla una bola con el color indicado en los estilos. La segunda función se denomina dibujar(), y es la encargada de limpiar el CANVAS, dibujar la bola y cambiar los valores de las coordenadas. Finalmente, la función setInterval(dibujar, 50), llama a la función dibujar cada 50 milisegundos.

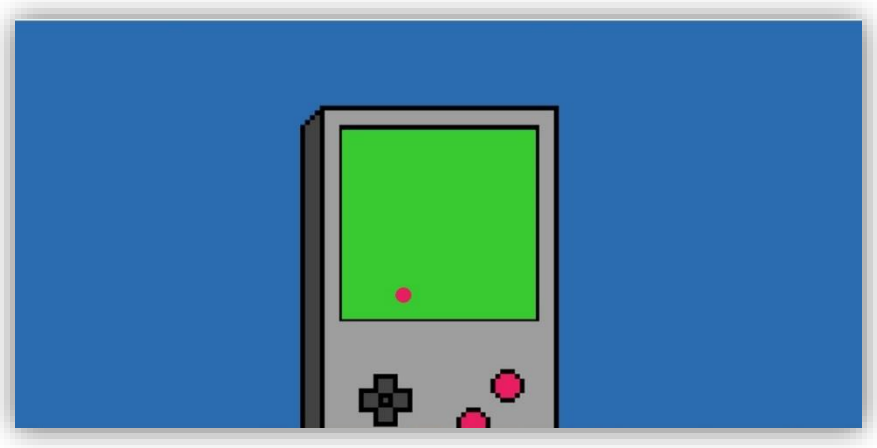
El código fuente del programa es el siguiente:

```
1. <!DOCTYPE html>
2. <html>
3. <head>
4.     <meta charset="utf-8" />
5.     <title>Juego 2D: JavaScript - 01</title>
```

```
6.
7.      <!-- Define los estilos de la interfaz visual
8.           padding es la distancia de un objeto en relación con el
   marco que lo contiene
9.           margin es la distancia que separa a un objeto de otro
10.               background es el color de fondo
11.               display: block; Estos elementos fluyen hacia abajo
12.               margin: 0 auto; Centra el canvas en la pantalla -->
13.           <style>
14.               * {
15.                   padding: 0;
16.                   margin: 0;
17.               }
18.               canvas {
19.                   background: #4FBF25;
20.                  display: block;
21.                   margin: 0 auto;
22.                   position: 0 ;
23.                    top:0;
24.                   Bottom:0;
25.                   Left: 0;
26.                   Right:0;
27.       }
28.           </style>
29.
30.       </head>
31.       <body style="background-color:#0E74E6" >
32.       <center> <img src="gor.png" width="1000" height="445">
33.       </center>
34.       <canvas id="miCanvas" width="290" height="268"></canvas>
35.
36.       <script>
37.           var canvas = document.getElementById("miCanvas");
38.           var ctx = canvas.getContext("2d");
39.
40.           // Coloca x en la mitad del ancho deL CANVAS
41.           var x = canvas.width/2;
42.
43.           // Coloca y en la mitad de la altura del CANVAS (restando
   30 a dicho valor)
44.           var y = canvas.height-55;
45.
46.           /* DEFINE LOS INCREMENTOS EN X y en Y. El valor dy es
   negativo
47.              para que inicialmente el movimiento de la bola sea
   hacia arriba */
48.           var dx = -10;
49.           var dy = 0;
50.
51.
52.           function dibujarBola() {
53.               // Inicia el dibujo
54.               ctx.beginPath();
55.
```

```
56.              /* Define un círculo en las coordenadas (x, y) con
   radio 10
57.                 El ángulo va desde 0 hasta 2*PI (360 grados) */
58.              ctx.arc(x, y, 10, 0, Math.PI*2);
59.
60.              // Color de llenado
61.              ctx.fillStyle = "#9525BF";
62.
63.              // Se llena el círculo con el color indicado
64.              ctx.fill();
65.
66.              // Finaliza el dibujo
67.              ctx.closePath();
68.          }
69.
70.          /* LA FUNCIÓN dibujar REALIZA TRES TAREAS:
71.             1) Limpia el CANVAS. Inicio= (0,0) Ancho=canvas.width
   Altura=canvas.height
72.             2) Dibuja una bola en las coordenadas (x, y)
73.             3) Cambiar las coordenadas (x, y) agregando los
   valores dx, dy
74.                Con este cambio cada vez que se dibuja la bola,
   está en una nueva posición */
75.          function dibujar() {
76.
77.              // Limpia el CANVAS
78.              ctx.clearRect(0, 0, canvas.width, canvas.height);
79.
80.              // Dibuja la bola
81.              dibujarBola();
82.
83.              // Se incrementa x en el valor dx
84.              x = x + dx;
85.
86.              // Se incrementa y en el valor dy
87.              y = y + dy;
88.          }
89.
90.          /* EJECUTA LA FUNCIÓN dibujar CADA 50 MILISEGUNDOS
91.             Este es el mecanismo utilizado para construir un
   sistema que
92.             ejecuta acciones de manera permanente y periódica */
93.          setInterval(dibujar, 50);
94.      </script>
95.
96.      </body>
97.      </html>
```

Al ejecutar este código se obtiene la siguiente interfaz visual:



Gráfica 2. La interfaz inicial del juego

En la gráfica 2 se aprecia el dibujo de la bola, y la secuencia de movimiento a partir de los incrementos en X y Y que fueron definidos. En este caso se muestra la pelotica moviéndose de lado izquierdo de manera horizontal, debido a que el var dx es negativo y el var dy es 0, de por lo tanto su movimiento comienza de lado izquierdo y uniforme horizontal.

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.

4 FASE 2: REBOTANDO EN LAS PAREDES

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

En este paso se crean dos condiciones las cuales generan los límites permitidos a los que la bola puede llegar y va a rebotar:

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.

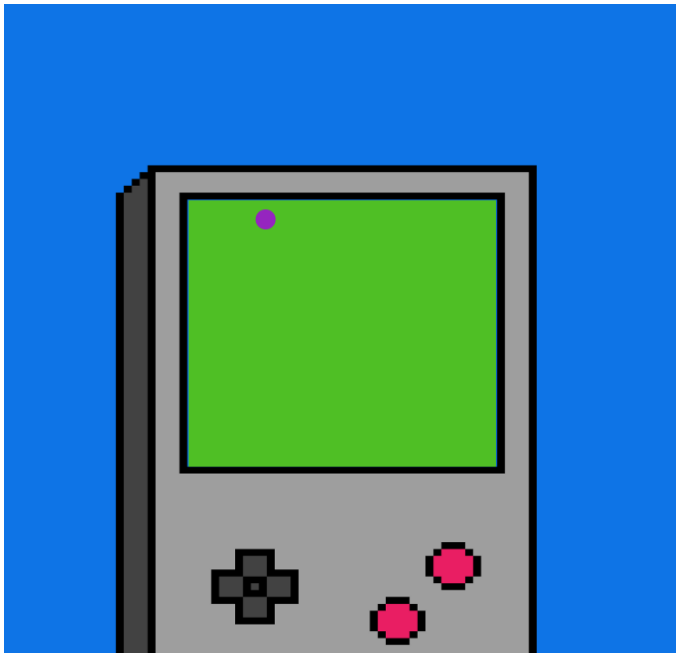
La segunda condición es if(y + dy > canvas.height-ballRadius || y + dy < ballRadius) {dy = -dy;}, esta condición crea el rango vertical al que la pelota se puede desplazar.

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8" />
    <title>Juego 2D - lección 02</title>
    <style>* { padding: 0; margin: 0; } canvas {background:#4FBF25;
          display: block;
           margin: 0 auto;
           position: 0 ;
                  top:0;
                Bottom:0;
                Left: 0;
                Right:0;
     }</style>
    </head>
     <body style="background-color:# 4FBF25" >
    <center> <img src="gor.png" width="445" height="1000">
    </center>

    <canvas id="miCanvas" width="290" height="268"></canvas>

    <script>
        var canvas = document.getElementById("miCanvas");
        var ctx = canvas.getContext("2d");
        var ballRadius = 10;
        var x = canvas.width/2;
        var y = canvas.height-30;
        var dx = 2;
        var dy = -2;

        function dibujarBola() {
            ctx.beginPath();
            ctx.arc(x, y, ballRadius, 0, Math.PI*2);
            ctx.fillStyle = "#9525BF";
            ctx.fill();
```

```
36.             ctx.closePath();
37.         }
38.
39.         function dibujar() {
40.             ctx.clearRect(0, 0, canvas.width, canvas.height);
41.             dibujarBola();
42.
43.             /* IMPORTANTE:
44.
45.                EL OPERADOR || es el operador lógico OR
46.                Este operador se utiliza para indicar la condición
   de conjunción
47.                SI SE CUMPLE UNA CONDICIÓN, O SE CUMPLE OTRA
   CONDICIÓN, ENTONCES
48.                SE CUMPLE LA CONDICIÓN
49.
50.                EL OPERADOR && es el oerador lógico AND
51.               Este operador se utiliza para indicar la condicion
   de disyunción
52.               SI SE CUMPLE UNA CONDICIÓN, Y SE CUMPLE OTRA
   CONDICIÓN (simultánea), ENTONCES
53.               SE CUMPLE LA CONDICIÓN
54.
55.             */
56.
57.             /* DESPUÉS DE DIBUJAR LA BOLA, SE DEBEN CAMBIAR LAS
   COORDENADAS
58.                EN LA lección 01 NO SE TENÍA CONTROL SOBRE LOS
   LÍMITES DE LA CAJA
59.                ------------------------------------------------
   --------------
60.                SI x + dx ES MAYOR AL ANCHO DEL CANVAS O MENOR AL
   TAMAÑO DEL
61.                RADIO DE LA BOLA (caso en el cual se encuentra
   hacia la izquierda)
62.                SE CAMBIA LA DIRECCIÓN DE AVANCE HORIZONTAL.
63.                ESTO SE LOGRA CAMBIANDO EL SIGNO DE LA VARIABLE dx
64.                ESTO HACE QUE SE CAMBIE EL SENTIDO DEL MOVIMIENTO
   HORIZONTAL */
65.             if(x + dx > canvas.width-ballRadius || x + dx <
   ballRadius) {
66.                 dx = -dx;
67.             }
68.
69.             /* SI y + dy ES MAYOR A LA ALTURA DEL CANVAS O MENOR
   AL TAMAÑO DEL
70.                RADIO DE LA BOLA, SE CAMBIA LA DIRECCIÓN DEL
   AVANCE VERTICAL.
71.                ESTO SE LOGRA CAMBIANDO EL SIGNO DE LA VARIABLE dy
72.                ESTE CAMBIO EN dy HACE QUE SE MUEVA VERTICALMENTE
   EN SENTIDO
73.                OPUESTO */
74.             if(y + dy > canvas.height-ballRadius || y + dy <
   ballRadius) {
```

```
75.                 dy = -dy;
76.             }
77.
78.             /* AQUÍ SE CAMBIA LA POSICIÓN DE LA BOLA. SE TOMA EN
    CUENTA LAS
79.                MODIFICACIONES A dx y dy, EN CASO DE QUE SE
    HUBIERAN PRODUCIDO */
80.             x += dx;
81.             y += dy;
82.         }
83.
84.         setInterval(dibujar, 10);
85.     </script>
86.
87.     </body>
88.     </html>
```

Al ejecutar este código se obtiene la siguiente interfaz visual; si se puede apreciar nota la pelota, rebota en dentro del cuadro , no se sale de ese límite, debido a las dos condiciones (horizontal y vertical) que se coloco anteriormente.



En la figura 3 podemos observar a la bola rebotando y cumpliendo con los límites anteriormente definidos en las condiciones.

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 PALA Y EL TECLADO

El paso numero 3 consiste en crear la paleta en la cual la bola rebotara, la cual estará situada en la parte inferior del juego y será controlada por ambas flechas del teclado.

Para empezar con el tercer paso primero se deben crear dos variables a las cuales se les asignara el movimiento de la paleta con las flechas del teclado. Estas variables llevaran el siguiente nombre flechaDerechaPulsada y flechaIzquierdaPulsada. Primero se empieza a definir sus variables en su altura, anchura y posición en el eje x en donde empieza a dibujar, así:

```
var paddleHeight = x;

var paddleWidth = x;

var paddleX = (canvas.width-paddleWidth)/x;
```

Después se añade la función de dibujo, tal como la bola solo que con la diferencia en hacer la forma rectangular, que es el siguiente:

```
ctx.rect(paddleX, canvas.height-paddleHeight, paddleWidth, paddleHeight).
```

Luego de esto se crea una función la cual maneja el movimiento de la tecla presionada y otro de la tecla liberada: con estas variables:

```
var rightPressed = false;

var leftPressed = false;
```

Se coloca “false” debido a en el principio las teclas no están siendo ejecutadas, por tal forma se establece la función “escuchadores de eventos” las siguientes variables:

```
document.addEventListener("keydown", keyDownHandler, false);

document.addEventListener("keyup", keyUpHandler, false);

function keyDownHandler(e) {

   if(e.keyCode == 39) {rightPressed = true; }

   else if(e.keyCode == 37) {leftPressed = true;}}


function keyUpHandler(e) {if(e.keyCode == 39)
```

{ rightPressed = false;}else if(e.keyCode == 37) {leftPressed = false; }}

Y por última esta la lógica del movimiento, donde allí se estable su velocidad y los límites de la barra dentro del Canvas, ejecutando esos códigos:

if(rightPressed) {paddleX += x;}

else if(leftPressed) {paddleX -= x;}

Esta determina su velocidad por píxeles, en que la barra se moverá, dependerá de los números de píxeles en que la barra se moverá más lento o despacio.

Mientras que con esta condiciónes:

if(rightPressed && paddleX < canvas.width-paddleWidth) {paddleX += x;}

else if(leftPressed && paddleX > 0) {paddleX -= x;}

Determina, que la barra no se saldrá de los límites del Canvas.

Estos son los códigos que se dio en este paso:

```
1. <!DOCTYPE html>
2. <html>
3. <head>
4.     <meta charset="utf-8" />
5.     <title>Juego 2D - #03 - Paleta y Control por Teclado</title>
6. <style>* { padding: 0; margin: 0; } canvas
   {background:#4FBF25;display: block;margin: 0 auto; position: 0
   top:0;Bottom:0;left: 0;Right:0;
7. } canvas { background: #4FBF25; display: block; margin: 0 auto;
   }</style>
8. </head>
9. <body style="background-color:# 4FBF25" >
10.      <center> <img src="gor.png" width="445" height="1000">
11.      </center>
12.
13.      <canvas id="miCanvas" width="290" height="268"></canvas>
14.
15.      <script>
16.          var canvas = document.getElementById("miCanvas");
17.          var ctx = canvas.getContext("2d");
18.
19.          /* Variables básicas:
20.
21.             radioBola: radio de la esfera
22.             x: columna en la que se encuentra situada la bola
23.             y: fila en la que se encuentra situada la bola
24.             dx: desplazamiento horizontal de la bola
25.             dy: desplazamiento vertical de la bola
26.
```

```
27.             NOTAS: originalmente, la bola está en centro del
   CANVAS
28.                    en el sentido horizontal. Y se encuentra en la
29.                    base inferior, pues el eje Y crece de arriba
   hacia
30.                    abajo. A este valor se le resta 30, para tomar
   en
31.                    cuenta el tamaño de la bola (que es de 20 si
   tomamos
32.                    en cuenta el diámetro)
33.
34.             NOTAS: El desplazamiento en el eje X y en el eje Y,
   son
35.                    controlados por la variable dx y la variable dy
36.                    Estos valores son de 2 pixeles, y gracias a
   este
37.                    avance que se realiza en un ciclo ejecutado
   cada
38.                    10 milisegundos, se genera el efecto de avance
   de
39.                    la bola. Dentro del ciclo se cambia la
   coordenada
40.                    (x, y) agregando los valores (dx, dy), motivo
   por
41.                    el cual la bola cambia su posición cada 10
   milisegundos
42.          */
43.          var radioBola = 11;
44.          var x = canvas.width/3;
45.          var y = canvas.height-28;
46.          var dx = 4;
47.          var dy = -4;
48.
49.          /* Las variables a continuación, tienen el siguiente
   significado:
50.
51.                Se define una paleta en la que rebotará la bola
52.                La paleta está situada en la base de la pantalla
   de juego
53.                Dicha paleta será controlada por la flecha
   izquierda y
54.                la flecha derecha del teclado (luego será
   controlador por el ratón)
55.
56.                alturaPaleta: define la altura de la paleta en
   pixeles
57.                anchuraPaleta: define la anchura de la paleta
58.
59.                NOTA: Estos dos valores determinan el tamaño de la
   paleta
60.                      La paleta se encuentra situada en la base de
   la pantalla
61.                      Para calcular la posición en X de la paleta,
   se debe tomar
```

```
62.                         el ancho del CANVAS, restarle la anchura de
   la paleta, y
63.                         el espacio que sobre debe dividirse entre
   dos
64.                         Esto garantiza que originalmente la paleta
   estará centrada
65.                         en la base de la pantalla
66.
67.                 Al inicio del juego, aún no se ha presionado
   ninguna de las
68.                 flechas. Esta es la razón por la cual se definen
   dos variables que
69.                 "recuerdan" cual de las flechas se ha presionado,
   pero que
70.                 inicialmente están puestas a: false, indicando el
   estado inicial
71.                 Cuando se pulse cualquiera de las dos flechas, su
   valor será:
72.                 true (verdadero), y este valor permitira
   establecer en qué
73.                 dirección se debe mover la paleta (dentro del
   ciclo del juego)
74.                 Las variables son:
75.
76.                 flechaDerechaPulsada
77.                 flechaIzquierdaPulsada
78.
79.                 NOTA: Desde ahora debe tomarse en cuenta que
   cuando se pulse
80.                         cualquiera de las dos flechas, solamente se
   hará un
81.                         desplazamiento de la paleta a la izquierda o
   hacia la
82.                         derecha. Si se mantiene pulsada la tecla, la
   paleta se
83.                         continuará desplazando, hasta alcanzar el
   extremo derecho
84.                         o izquierdo de la pantalla del juego
85.          */
86.          var alturaPaleta = 18;
87.          var anchuraPaleta = 90;
88.          var paletaPosX = (canvas.width-anchuraPaleta)/3;
89.          var flechaDerechaPulsada = false;
90.          var flechaIzquierdaPulsada = false;
91.
92.          /* La instruccion: addEventListener, se utiliza para
   crear un
93.             mecanismo de respuesta ante eventos que se produzcan
   en el juego
94.
95.             addEventListener "agregar un mecanismo que deteca y
   recibe eventos"
96.
97.             addEventListener recibe tres parámetros:
```

```
98.
99.             1) El evento que se va a detectar
100.            2) El nombre que le asignamos a la función que
   responde ante el evento
101.            3) Valor true o false que determina la reacción ante
   el evento
102.
103.            Los dos primeros parámetros son fáciles de entender.
   Pero el tercero
104.            requiere de una explicación adicional:
105.
106.            Para entender el tercer parámetro, primero hemos de
   saber lo que es
107.            el flujo de eventos.
108.
109.                Supongamos que tenemos este tres objetos en la
   página:
110.
111.            <body>
112.                <div>
113.                    <button>HAZME CLIC</button>
114.                </div>
115.            </body>
116.
117.            El <body> contiene un <div>, y dentro de él esta
   un <button>
118.
119.            Cuando hacemos clic en el botón no sólo lo estamos
   haciendo sobre él,
120.            sino sobre los elementos que lo contienen en el árbol
   de la página,
121.            es decir, hemos hecho clic, además, sobre el
   elemento <body> y sobre
122.            el elemento <div>. Sí sólo hay una función asignada a
   una escucha
123.            para el botón, no hay mayor problema, pero si además
   hay una
124.            eschucha para el body y otra para el div,
125.            ¿cuál es el orden en que se deben lanzar las tres
   funciones?
126.
127.                Para contestar a esa pregunta existe un modelo de
   comportamiento,
128.                el flujo de eventos. Según éste, cuando se hace
   clic sobre un
129.                elemento, el evento se propaga en dos fases, una
   que es la
130.                captura —el evento comienza en el nivel superior
   del documento
131.                y recorre los elementos de padres a hijos— y la
   otra la burbuja
132.                —el orden inverso, ascendiendo de hijos a padres—.
133.
```

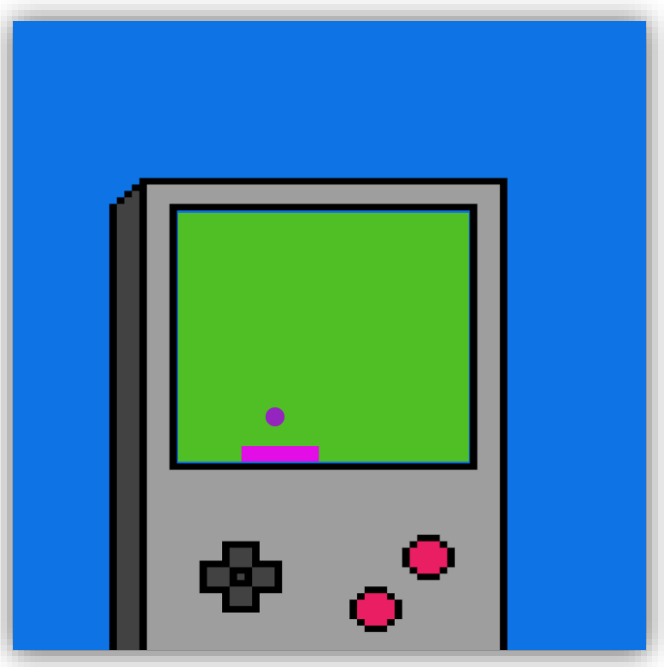
```
134.                Así, el orden por defecto de lanzamiento de las
   funciones
135.                de escucha, sería: primero la función de escuch de
   body,
136.                luego la función de escucha de div, y por último
   la función
137.                de escucha de button.
138.
139.                Una vez visto esto, podemos comprender el tercer
   parámetro de addEventListener, que lo que hace es permitirnos
   escoger el orden de propagación:
140.
141.            true: El orden de propagación para el ejemplo sería,
   por tanto,
142.            body-div-button
143.
144.            false: La propagación seguiría el modelo burbuja.
145.            Así, el orden sería button-div-body.
146.
147.            NOTA: omo en nuestro ejemplo utilizamos "false",
   estamos
148.                  eaccionando primero ante el evento sobre las
   teclas,
149.                   posteriormente sobre los eventos asociados al
   CANVAS.
150.                  ste es el mecanismo más usual, pero se utilizará
   "true"
151.                  n las situaciones que lo requieran
152.         */
153.         document.addEventListener("keydown",
   manejadorTeclaPresionada, false);
154.         document.addEventListener("keyup",
   manejadorTeclaLiberada, false);
155.
156.         // Función que maneja tecla presionada
157.         function manejadorTeclaPresionada(e) {
158.             if(e.keyCode == 39) {
159.                     /* e: Es el evento que se produce, en este
   caso
160.                                         tecla presionada. La
   propiedad: keyCode permite
161.                                         descubrir de qué tecla se
   trata. Si el código es 39,
162.                                         se ha presionado la flecha
   derecha. En este caso
163.                                         se coloca la variable:
   flechaDerechaPulsada a true
164.                                 */
165.                 flechaDerechaPulsada = true;
166.             }
167.             else if(e.keyCode == 37) {
168.                     /* e: Es el evento que se produce, en este
   caso
```

```
169.                                        tecla presionada. La
   propiedad: keyCode permite
170.                                        descubrir de qué tecla se
   trata. Si el código es 37,
171.                                        se ha presionado la flecha
   izquierda. En este caso
172.                                        se coloca la variable:
   flechaIquierdaPulsada a true
173.                               */
174.                 flechaIzquierdaPulsada = true;
175.             }
176.         }
177.
178.         // Función que maneja tecla liberada
179.         function manejadorTeclaLiberada(e) {
180.             if(e.keyCode == 39) {
181.                     /* Si la tecla liberada es la 39, se ha
   dejado de
182.                        presionar la flecha derecha. En este caso,
   la variable
183.                        se pone en: false
184.                     */
185.                 flechaDerechaPulsada = false;
186.             }
187.             else if(e.keyCode == 37) {
188.                     /* Si la tecla liberada es la 37, se ha
   dejado de
189.                        presionar la flecha izquierda. En este
   caso, la variable
190.                        se pone en: false
191.                     */
192.                 flechaIzquierdaPulsada = false;
193.             }
194.         }
195.
196.         // Dibuja la bola. Código explicado en anteriores
   programas
197.         function dibujarBola() {
198.             ctx.beginPath();
199.             ctx.arc(x, y, radioBola, 0, Math.PI*2);
200.             ctx.fillStyle = "#9525BF";
201.             ctx.fill();
202.             ctx.closePath();
203.         }
204.
205.         function dibujarPaleta() {
206.             // Se inicia el dibujo de la paleta
207.             ctx.beginPath();
208.             /* Se crea un rectángulo utilizando la posición en X
209.                El valor de Y está en la base de la pantalla menos
   la
210.                altura de la paleta
211.                Y a continuación se indica la anchura y la altura
   de la paleta
```

```
212.                     */
213.             ctx.rect(paletaPosX, canvas.height-alturaPaleta,
   anchuraPaleta, alturaPaleta);
214.             ctx.fillStyle = "#E30EE6";
215.             ctx.fill();
216.             // Se "cierra" la paleta, terminando su dibujo
217.             ctx.closePath();
218.         }
219.
220.         // Función principal. A partir de aquí se origina el
   proceso
221.         // general del juego
222.         function dibujar() {
223.             ctx.clearRect(0, 0, canvas.width, canvas.height);
224.
225.             // En primer lugar, dibuja la bola
226.             dibujarBola();
227.
228.             // Seguidamente, dibuja la paleta
229.             dibujarPaleta();
230.
231.             /* IMPORTANTE:
232.
233.                EL OPERADOR || es el operador lógico OR
234.                Este operador se utiliza para indicar la condición
   de conjunción
235.                SI SE CUMPLE UNA CONDICIÓN, O SE CUMPLE OTRA
   CONDICIÓN, ENTONCES
236.                SE CUMPLE LA CONDICIÓN
237.
238.                EL OPERADOR && es el oerador lógico AND
239.               Este operador se utiliza para indicar la condicion
   de disyunción
240.               SI SE CUMPLE UNA CONDICIÓN, Y SE CUMPLE OTRA
   CONDICIÓN (simultánea), ENTONCES
241.               SE CUMPLE LA CONDICIÓN
242.
243.            */
244.
245.             // Aquí se controla los límites a los que puede llegar
   la bola
246.             // En caso de intentar sobrepasar dichos límites, se
   cambia
247.             // el sentido del movimiento
248.             // Este código se explicó en el anterior programa
249.             if(x + dx > canvas.width-radioBola || x + dx <
   radioBola) {
250.                 dx = -dx;
251.              }
252.              if(y + dy > canvas.height-radioBola || y + dy <
   radioBola) {
253.                 dy = -dy;
254.             }
255.
```

```
256.            /* Si se ha pulsado la flecha derecha, y la paleta aún
   puede
257.               desplazarse hacia la derecha sin que se sobrepase
   el límite de la
258.               pantalla, entonces se procede a cambiar su posición
259.               En este caso, la función: dibujarPaleta (la cual se
   ejecuta de
260.               manera cíclica) redibujará la paleta en la nueva
   posición
261.            */
262.            if(flechaDerechaPulsada && paletaPosX < canvas.width-
   anchuraPaleta) {
263.                        // Se desplaza la paleta hacia la derecha
264.                        // Aquí, paletaPosX += 7 equivale a:
   paletaPosX = paletaPosX + 7
265.                paletaPosX += 7;
266.            }
267.            else if(flechaIzquierdaPulsada && paletaPosX > 0) {
268.                        // Se desplaza la paleta hacia la izquierda
269.                        // Aquí, paletaPosX -= 9 equivale a:
   paletaPosX = paletaPosX - 9
270.                 paletaPosX -= 9;
271.             }
272.
273.             x += dx;
274.             y += dy;
275.         }
276.
277.         /* Con esta instrucción se crea un ciclo. Cada 10
   milisegundos se
278.            ejecuta la funcion: dibujar(). Esto genera el ciclo
   que permitirá
279.            actualizar el juego, detectar eventos y cambiar el
   estado
280.            de los objetos según las nuevas posiciones que ocupen
   los
281.            elementos del juego
282.
283.            NOTA: La función que se ejecuta es: dibujar
284.                  Por tanto, dicha función es la encargada de
   "lanzar" el juego
285.                  y dentro de ella se realizarán las acciones que
   desencadenan
286.                  el juego como tal
287.         */
288.         setInterval(dibujar, 27);
289.     </script>
290.
291.     </body>
292.     </html>
```

Al ejecutar este código se obtiene la siguiente interfaz visual:



En la figura 4 podemos observar la bola y la paleta en la parte inferior del juego creadas anteriormente en la parte número 3 del código. Ahí se denota la utilidad de la barra y como esta cumple perfectamente su trabajo.

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.

6 FASE 4: FIN DEL JUEGO

En esta parte del programa programaremos que se pueda detectar cuando la bola toca la base de la pantalla, en una coordenada diferente a la de donde se encuentra la paleta, lo que hará que el juego se pierda.

Para este caso analizaremos un código en la función dibujar, el código seria:

(y + dy > canvas.height - radioBola)

El cual se utilizaría para cuando la bola toque la parte inferior del juego lo cual haría que el juego se pierda. Pero se necesitaría que la pelota golpea e rebote la paleta, de tal forma colocaremos la forma **"elfe if"** y el siguiente código:

```
else if(y + dy > canvas.height-ballRadius) {
  if(x > paddleX && x < paddleX + paddleWidth) {
    dy = -dy;
  }
```

Para finalizar hay que agregar el letrero donde le dice al jugador que se acabó el juego, con la etiqueta **alert**, así quedaría el código de esa fase:

```
  else {
    alert("GAME OVER");
    document.location.reload();
  }
}
```

El código completa de esta fase es el siguiente:

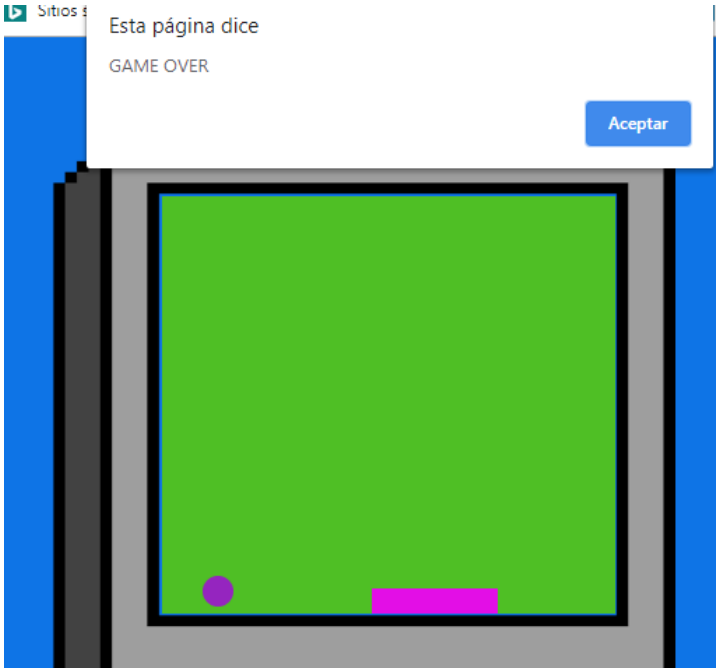
```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8" />
    <title>Juego 2D - #04 - Game Over</title>
<style>* { padding: 0; margin: 0; }
canvas {background:#4FBF25;display: block;margin: 0 auto; position:
0 top:0;Bottom:0;left: 0;Right:0;}
canvas { background: #4FBF25; display: block; margin: 0 auto;
}</style>
```

```
9. </head>
10.       <body style="background-color:#4FBF25" >
11.       <center> <img src="gor.png" width="445" height="1000">
12.       </center>
13.
14.       <canvas id="miCanvas" width="290" height="268"></canvas>
15.
16.       <script>
17.               /* Este programa detecta cuando la bola toca la base
   de la pantalla
18.                  Lo anterior significa que la paleta está en otra
   posición distinta
19.                  al punto de toque de la bola con la base de la
   pantalla
20.                  En este caso, se considera que el jugador ha
   perdido una vida
21.                  El sistema lo informa generando una alerta
22.                  El código se encuentra dentro de la función
   dibujar
23.               */
24.
25.          var canvas = document.getElementById("miCanvas");
26.          var ctx = canvas.getContext("2d");
27.
28.          var radioBola = 11;
29.          var x = canvas.width/3;
30.          var y = canvas.height-28;
31.          var dx = 4;
32.          var dy = -4;
33.
34.          var alturaPaleta = 18;
35.          var anchuraPaleta = 90;
36.          var paletaPosX = (canvas.width-anchuraPaleta)/3;
37.
38.          var flechaDerechaPresionada = false;
39.          var flechaIzquierdaPresionada = false;
40.
41.          document.addEventListener("keydown",
   manejadorTeclaPresionada, false);
42.          document.addEventListener("keyup",
   manejadorTeclaLiberada, false);
43.
44.          function manejadorTeclaPresionada(e) {
45.              if(e.keyCode == 39) {
46.                  flechaDerechaPresionada = true;
47.              }
48.              else if(e.keyCode == 37) {
49.                  flechaIzquierdaPresionada = true;
50.              }
51.          }
52.          function manejadorTeclaLiberada(e) {
53.              if(e.keyCode == 39) {
54.                  flechaDerechaPresionada = false;
55.              }
```

```
56.                 else if(e.keyCode == 37) {
57.                     flechaIzquierdaPresionada = false;
58.                 }
59.             }
60.
61.             function dibujarBola() {
62.                 ctx.beginPath();
63.                 ctx.arc(x, y, radioBola, 0, Math.PI*2);
64.                 ctx.fillStyle = "#9525BF";
65.                 ctx.fill();
66.                 ctx.closePath();
67.             }
68.             function dibujarPaleta() {
69.                 ctx.beginPath();
70.                 ctx.rect(paletaPosX, canvas.height-alturaPaleta,
    anchuraPaleta, alturaPaleta);
71.                 ctx.fillStyle = "#E30EE6";
72.                 ctx.fill();
73.                 ctx.closePath();
74.             }
75.
76.             function dibujar() {
77.                 ctx.clearRect(0, 0, canvas.width, canvas.height);
78.
79.                 dibujarBola();
80.                 dibujarPaleta();
81.
82.                 if(x + dx > canvas.width-radioBola || x + dx <
    radioBola) {
83.                     dx = -dx;
84.                 }
85.                 if(y + dy < radioBola) {
86.                     dy = -dy;
87.                 }
88.
89.                 /* Si y + dy alcanza la frontera inferior de la
    pantalla
90.                    (y + dy > canvas.height - radioBola)
91.                    existe la posibilidad de que el jugador pierda el
    juego
92.                    Para ello debe evaluarse una segunda opción:
93.                    La variable x determina la posición de la bola
94.                    Lo que debe hacerse es mirar si x está DENTRO de
    la palata:
95.                    (x > paletaPosX && x < paletaPosX + anchuraPaleta)
96.                            ----------------------------------------
    -----------------
97.                            Si x está dentro de la paleta, todo va
    bien y se incrementa y
98.                            ----------------------------------------
    -----------------
99.                            Si x NO ESTÁ dentro de la paleta (else),
    la bola ha llegado
```

```
100.                        a la frontera inferior, y no encuentra la
    paleta en su camino
101.                        En este caso, SE DETIENE EL CICLO DE
    ANIMACIÓN, y se genera
102.                        un ALERT indicando que el jugador ha
    perdido (GAME OVER)
103.            */
104.            else if(y + dy > canvas.height-radioBola) {
105.                if(x > paletaPosX && x < paletaPosX +
    anchuraPaleta) {
106.                    dy = -dy;
107.                }
108.                else {
109.                    clearInterval(juego);
110.                    alert("GAME OVER");
111.                    document.location.reload();
112.                }
113.            }
114.
115.            if(flechaDerechaPresionada && paletaPosX <
    canvas.width-anchuraPaleta) {
116.                paletaPosX += 9;
117.            }
118.            else if(flechaIzquierdaPresionada && paletaPosX > 0)
    {
119.                paletaPosX -= 9;
120.            }
121.
122.            x += dx;
123.            y += dy;
124.        }
125.
126.        /* En este programa se asigna a una variable el proceso
    cíclico
127.           Esto tiene mucha importancia, porque si en algún
    momento se requiere
128.           eliminar el ciclo, se utilizará la variable asignada
129.        */
130.        var juego = setInterval(dibujar, 27);
131.    </script>
132.
133.    </body>
134.    </html>
```

Al ejecutar este código se obtiene la siguiente interfaz visual:



En la figura 5 podemos observar como la bola al tocar la parte inferior del juego y al estar en una coordenada diferente a la paleta aparece un game over que significa que el juego se ha perdido y se ha acabado, si de da aceptar el programa se reinicia.

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.

7 FASE 5: MURO DE LADRILLOS

En esta parte del juego crearemos unas variables las cuales crearan un muro de ladrillos dentro del juego en los cuales rebotara la bola con estos códigos, para que no se dibujen tocando los bordes:

```
var brickRowCount = 3;

var brickColumnCount = 5;

var brickWidth = 75;

var brickHeight = 20;

var brickPadding = 10;

var brickOffsetTop = 30;

var brickOffsetLeft = 30;
```

Analizaremos la sgte funcion: function dibujarLadrillos() , esta función se apoya de varias variables para la creación del muro de los ladrillos la cual la hace analizando la columna y la fila en la que quedara asignado cada ladrillo.

```
function drawBricks() {
   for(c=0; c<brickColumnCount; c++) {
      for(r=0; r<brickRowCount; r++) {
         var brickX = (c*(brickWidth+brickPadding))+brickOffsetLeft;
         var brickY = (r*(brickHeight+brickPadding))+brickOffsetTop;
         bricks[c][r].x = brickX;
         bricks[c][r].y = brickY;
         ctx.beginPath();
         ctx.rect(brickX, brickY, brickWidth, brickHeight);
         ctx.fillStyle = "#0095DD";
         ctx.fill();
         ctx.closePath();}}}
```

Se presenta el siguiente código complete de la fase:

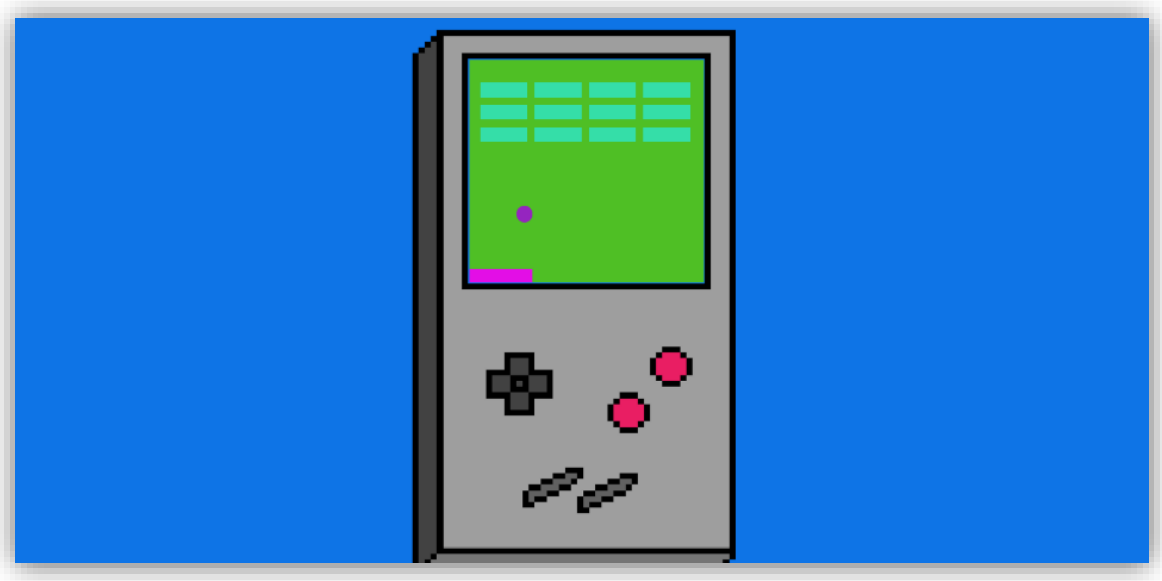
```
1. <!DOCTYPE html>
2. <html>
3. <head>
4.     <meta charset="utf-8" />
5.     <title>Juego 2D: #05 - Construcción de los ladrillos</title>
6. <style>* { padding: 0; margin: 0; }
7.  canvas {background:#4FBF25; padding: 0;display: block; margin: 0
   auto; position: absolute top:0;bottom:0;left: 0;right:0;
8. }</style>
9. </head>
10.      <body style="background-color:#0E74E6" >
11.      <center> <img src="gor.png" width=445 height=1000">
12.      </center>
13.      <canvas id="miCanvas" width="290" height="268"></canvas>
14.
15.      <script>
16.          var canvas = document.getElementById("miCanvas");
17.          var ctx = canvas.getContext("2d");
18.          var radioBola = 11;
19.          var x = canvas.width/3;
20.          var y = canvas.height-28;
21.          var dx = 4;
22.          var dy = -4;
23.          var alturaPaleta = 18;
24.          var anchuraPaleta = 90;
25.          var paletaPosX = (canvas.width-anchuraPaleta)/3;
26.          var flechaDerechaPresionada = false;
27.          var flechaIzquierdaPresionada = false;
28.
29.          /* NUEVAS VARIABLES asociadas a los ladrillos
30.          */
31.          var nroFilasLadrillos = 4;
32.          var nroColumnasLadrillos = 3;
33.          var anchoLadrillo = 65;
34.          var alturaLadrillo = 20;
35.          var rellenoLadrillo = 10;
36.          var vacioSuperiorLadrillo = 30;
37.          var vacioIzquierdoLadrillo = 15;
38.
39.          // Crea el conjunto de ladrillos. Inicialmente, vacío
40.          var ladrillos = [];
41.
42.          // Recorre cinco columnas
43.          for(var columna=0; columna<nroColumnasLadrillos;
   columna++) {
44.              // Define la primera columna. Es una lista vertical
45.              ladrillos[columna] = [];
46.
47.              // Para la columna, recorre las tres filas, una
   después de otra
48.              for(var fila=0; fila<nroFilasLadrillos; fila++) {
49.                  // Para cada (columna, fila) se define un ladrillo
50.
51.
```

```
52.                 /* IMPORTANTE:
53.                    Como se puede observar, cada ladrillo está
   definido como: ==> ladrillos[c][f]
54.                    Los valores c y f, se corresponden con la fila
   y la columna, DENTRO
55.                    DE LA MATRIZ DE LADRILLOS
56.                    ------------------------------------------------
   ------------------------------
57.                    A cada ladrillo en la posicion (c, f), se le
   asignan tres valores:
58.
59.                       x: Su coordenada horizontal EN LA PANTALLA
60.                       y: Su coordenada vertical EN LA PANTALLA
61.
62.                    ------------------------------------------------
   ------------------------------
63.                    Los valores x y y valen originalmente cero (0)
64.                    Esto cambia cuando se dibujan (más adelante,
   en la función: dibujarLadrillos())
65.                 */
66.                 ladrillos[columna][fila] = { x: 0, y: 0 };
67.             }
68.         }
69.
70.         document.addEventListener("keydown",
   manejadorTeclaPresionada, false);
71.         document.addEventListener("keyup",
   manejadorTeclaLiberada, false);
72.
73.         function manejadorTeclaPresionada(e) {
74.             if(e.keyCode == 39) {
75.                 flechaDerechaPresionada = true;
76.             }
77.             else if(e.keyCode == 37) {
78.                 flechaIzquierdaPresionada = true;
79.             }
80.         }
81.         function manejadorTeclaLiberada(e) {
82.             if(e.keyCode == 39) {
83.                 flechaDerechaPresionada = false;
84.             }
85.             else if(e.keyCode == 37) {
86.                 flechaIzquierdaPresionada = false;
87.             }
88.         }
89.         function dibujarBola() {
90.             ctx.beginPath();
91.             ctx.arc(x, y, radioBola, 0, Math.PI*2);
92.             ctx.fillStyle = "#9525BF";
93.             ctx.fill();
94.             ctx.closePath();
95.         }
96.
97.         function dibujarPaleta() {
```

```
98.             ctx.beginPath();
99.             ctx.rect(paletaPosX, canvas.height-alturaPaleta,
   anchuraPaleta, alturaPaleta);
100.            ctx.fillStyle = "#E30EE6";
101.            ctx.fill();
102.            ctx.closePath();
103.        }
104.
105.        /* FUNCIÓN QUE DIBUJA LOS LADRILLOS
106.           -------------------------------
107.        */
108.        function dibujarLadrillos() {
109.            // Recorre todas las columnas
110.            for(var columna=0; columna<nroColumnasLadrillos;
   columna++) {
111.                // Para cada columna, recorre sus filas
112.                for(var fila=0; fila<nroFilasLadrillos; fila++) {
113.                    // Calcula la coordenada x del ladrillo, según
   en que fila se encuentre
114.                    // según el ancho del ladrillo, el valor de
   relleno interno
115.                    // y el espacio que debe dejar a la izquierda
116.                    // NOTA: Se sugiere asignar valores y dibujar
   el esquema a mano
117.                    var
   brickX = (fila*(anchoLadrillo+rellenoLadrillo))+vacioIzquierdoLadri
   llo;
118.
119.                    // Repite el proceso para calcular la
   coordenada y del ladrillo
120.                    var
   brickY = (columna*(alturaLadrillo+rellenoLadrillo))+vacioSuperiorLa
   drillo;
121.
122.                    // ASIGNA AL LADRILLO EN LA columna, fila QUE
   LE CORRESPONDE EN LA MATRIZ
123.                    // EL VALOR CALCULADO (brickX) A SU COORDENADA
   x
124.                    ladrillos[columna][fila].x = brickX;
125.
126.                    // IGUAL PARA EL VALOR y EN PANTALLA
127.                    ladrillos[columna][fila].y = brickY;
128.
129.                    // DIBUJA EL LADRILLO CON LOS VALORES
   ASOCIADOS:
130.                    // Coordenada: (brickX, brickY)
131.                    // Anchura: anchoLadrillo
132.                    // Altrua: alturaLadrillo
133.                    ctx.beginPath();
134.                    ctx.rect(brickX, brickY, anchoLadrillo,
   alturaLadrillo);
135.                    ctx.fillStyle = "#35DEA8";
136.                    ctx.fill();
137.                    ctx.closePath();
```

```
138.
139.                     // COMO SE RECORRE TODO EL CICLO, SE DIBUJAN
   TODOS LOS LADRILLOS
140.                 }
141.             }
142.         }
143.
144.         function dibujar() {
145.             ctx.clearRect(0, 0, canvas.width, canvas.height);
146.
147.             // DIBUJA EL CONJUNTO DE LADRILLOS
148.             dibujarLadrillos();
149.
150.             dibujarBola();
151.             dibujarPaleta();
152.
153.             if(x + dx > canvas.width-radioBola || x + dx <
   radioBola) {
154.                 dx = -dx;
155.             }
156.             if(y + dy < radioBola) {
157.                 dy = -dy;
158.             }
159.             else if(y + dy > canvas.height-radioBola) {
160.                 if(x > paletaPosX && x < paletaPosX +
   anchuraPaleta) {
161.                     dy = -dy;
162.                 }
163.                 else {
164.                     clearInterval(juego);
165.                     alert("GAME OVER");
166.
167.                     // RECARGA LA PÁGINA - El juego vuelve a
   empezar
168.                     document.location.reload();
169.                 }
170.             }
171.
172.             if(flechaDerechaPresionada && paletaPosX <
   canvas.width-anchuraPaleta) {
173.                 paletaPosX += 9;
174.             }
175.             else if(flechaIzquierdaPresionada && paletaPosX > 0)
   {
176.                 paletaPosX -= 9;
177.             }
178.             x += dx;
179.             y += dy;
180.         }
181.         var juego = setInterval(dibujar, 27);
182.     </script>
183.
184.     </body>
185.     </html>
```

Al ejecutar este código se obtiene la siguiente interfaz visual:



En la figura 6 podemos observar la creación de la pared de ladrillos dentro del campo del juego., sin embargo no se rompe los ladrillos

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.

8 FASE 6: DETECCIÓN DE COLISIONES

En esta parte del programa realizaremos la función que hará que se detecte la colisión de la bola con alguno de los ladrillos y al ocurrir esto hará que el ladrillo con el que colisiono desaparezca.

Procederemos a analizar la función que hace esto posible, la función será la siguiente:

```
function detectarColision(),

for(var c=0; c<nroColumnasLadrillos; c++) {

        for(var f=0; f<nroFilasLadrillos; f++) {var b = ladrillos[c][f];

        if(b.estado == 10) {

 if(x > b.x && x < b.x+anchuraLadrillos && y > b.y && y < b.y+alturaLadrillos) {

                dy = -dy;

                b.estado = b.estado - 1;

                b.ciclo = b.ciclo - 1

                }}} }}}
```

Esta es la función que permite que cuando la bola colisione con alguno de los ladrillos desaparezca, esto se realiza creando una variable temporal en la cual se asigna el ladrillo y analizando su columna y su fila y así saber si fue impactado y adicionalmente se le agrega los ciclos, donde con esos establecemos que hará el ladrillo al colisionar con la pelota y estos ladrillos se desvanecen .

```
function dibujarLadrillos() {

    for(var c=0; c<nroColumnasLadrillos; c++) {

      for(var r=0; r<nroFilasLadrillos; r++) {

        if(ladrillos[c][r].estado > 0) {

          var b = ladrillos[c][r];

          var posXLadrillo = (r*(anchuraLadrillos+rellenoLadrillos))+vacioIzquierdoLadrillo;

          var posYLadrillo = (c*(alturaLadrillos+rellenoLadrillos))+vacioSuperiorLadrillo;

          if (b.estado < 10 && b.estado > 1) {

            b.ciclo = b.ciclo - 1;

            if (b.ciclo < 1) {
```

```
        if (b.estado > 1) {
            b.estado = b.estado - 1;
            b.ciclo = numeroCiclos;
        }
    }
}

ladrillos[c][r].x = posXLadrillo;
ladrillos[c][r].y = posYLadrillo;
ctx.beginPath();
ctx.rect(posXLadrillo, posYLadrillo, anchuraLadrillos, alturaLadrillos);
if (b.estado == 10) {
    ctx.fillStyle = "#0F2407"; }
else if (b.estado == 9) {
  ctx.fillStyle = "#14300A";  }
else if (b.estado == 8) {ctx.fillStyle = "#193C0C"; }
else if (b.estado == 7) { ctx.fillStyle = "#1E480E"; }
else if (b.estado == 6) { ctx.fillStyle = "#235411"; }
else if (b.estado == 5) {ctx.fillStyle = "#286013";  }
else if (b.estado == 4) {ctx.fillStyle = "#317717"; }
else if (b.estado == 3) {ctx.fillStyle = "#3B8F1C"; }
else if (b.estado == 2) {ctx.fillStyle = "#45A720"; }
else if (b.estado == 1) {
  ctx.fillStyle = "#4FBF25";
}
else {
  ctx.fillStyle = "#4FBF25";
```

```
            }


            ctx.fill();
            ctx.closePath();
          }
        }
      }
    }
```

También se crean las siguientes variables: la primera es clearInterval(juego); la cual hace que se detenga el ciclo del juego, otra es alert("GAME OVER"); la cual hace que al perder el juego salga un letrero con la palabra GAME OVER que significa que se ha acabado el juego y por último la siguiente variable document.location.reload(); que hace que el juego se recargue nuevamente y se pueda volver a empezar, agregando este código en **la función dectectarcolision()**:

```
else if(y + dy > canvas.height-radioBola) {
        if(x > paletaPosX && x < paletaPosX + anchuraPaleta) {
           dy = -dy;
        }
        else {
          clearInterval(juego);
           alert("GAME OVER");
           document.location.reload();
        }
    }
```

```
1.  <!DOCTYPE html>
2. <html>
3. <head>
4.     <meta charset="utf-8" />
5.     <title>Juego 2D - #06 - Detección de colisión</title>
6.     <style>* { padding: 0; margin: 0; } canvas { background:
   #4FBF25;
7.     padding: 0;
8.     margin: auto;
9.     display: block;
```

```
10.          position: absolute;
11.          top: 0;
12.          bottom: 0;
13.          left: 0;
14.          right: 0;} </style>
15.      </head>
16.      <center>
17.           <img src="gor.png" height=”1000” width=445”>
18.              </center>
19.      <body style="background-color:#0E74E6 ;">
20.
21.      <canvas id="miCanvas" width="290" height="268"></canvas>
22.
23.      <script>
24.          var canvas = document.getElementById("miCanvas");
25.          var ctx = canvas.getContext("2d");
26.
27.          var radioBola = 11;
28.          var x = canvas.width/3;
29.          var y = canvas.height-30;
30.          var dx = 4;
31.          var dy = -4;
32.
33.          var alturaPaleta = 18;
34.          var anchuraPaleta = 480;
35.          var paletaPosX = (canvas.width-anchuraPaleta)/3;
36.
37.          var flechaDerechaPresionada = false;
38.          var flechaIzquierdaPresionada = false;
39.
40.          var nroFilasLadrillos = 5;
41.          var nroColumnasLadrillos = 5;
42.          var anchuraLadrillo = 52;
43.          var alturaLadrillo = 20;
44.          var rellenoLadrillo = 5;
45.          var vacioSuperiorLadrillo = 30;
46.          var vacioIzquierdoLadrillo = 3;
47.       var numeroCiclos = 10;
48.      var colorTexto = "#000000"
49.
50.          var ladrillos = [];
51.          for(var c=0; c<nroColumnasLadrillos; c++) {
52.              ladrillos[c] = [];
53.              for(var f=0; f<nroFilasLadrillos; f++) {
54.
55.                  ladrillos[c][f] = { x: 0, y: 0, estado: 10,
   ciclo: numeroCiclos };
56.
57.              }
58.          }
59.          document.addEventListener("keydown",
   manejadorTeclaPresionada, false);
60.          document.addEventListener("keyup",
   manejadorTeclaLiberada, false);
```

```
61.       document.addEventListener("mousemove", manejadorRaton,
   false);
62.          function manejadorTeclaPresionada(e) {
63.              if(e.keyCode == 39) {
64.                  flechaDerechaPresionada = true;
65.              }
66.              else if(e.keyCode == 37) {
67.                  flechaIzquierdaPresionada = true;
68.              }
69.          }
70.
71.          function manejadorTeclaLiberada(e) {
72.              if(e.keyCode == 39) {
73.                  flechaDerechaPresionada = false;
74.              }
75.              else if(e.keyCode == 37) {
76.                  flechaIzquierdaPresionada = false;
77.              }
78.          }
79.      function manejadorRaton(e) {
80.              var posXRatonDentroDeCanvas = e.clientX -
   canvas.offsetLeft;
81.              if(posXRatonDentroDeCanvas > 0 &&
   posXRatonDentroDeCanvas < canvas.width) {
82.                  paletaPosX = posXRatonDentroDeCanvas -
   anchuraPaleta/2;
83.              }
84.          }
85.
86.
87.
88.          function detectarColision() {
89.
90.                    for(var c=0; c<nroColumnasLadrillos; c++) {
91.                  for(var f=0; f<nroFilasLadrillos; f++) {
92.
93.
94.
95.                     var b = ladrillos[c][f];
96.
97.
98.
99.                     if(b.estado == 10) {
100.
101.
102.                        if(x > b.x && x < b.x+anchuraLadrillos &&
   y > b.y && y < b.y+alturaLadrillos) {
103.
104.
105.
106.                           dy = -dy;
107.
108.
109.
```

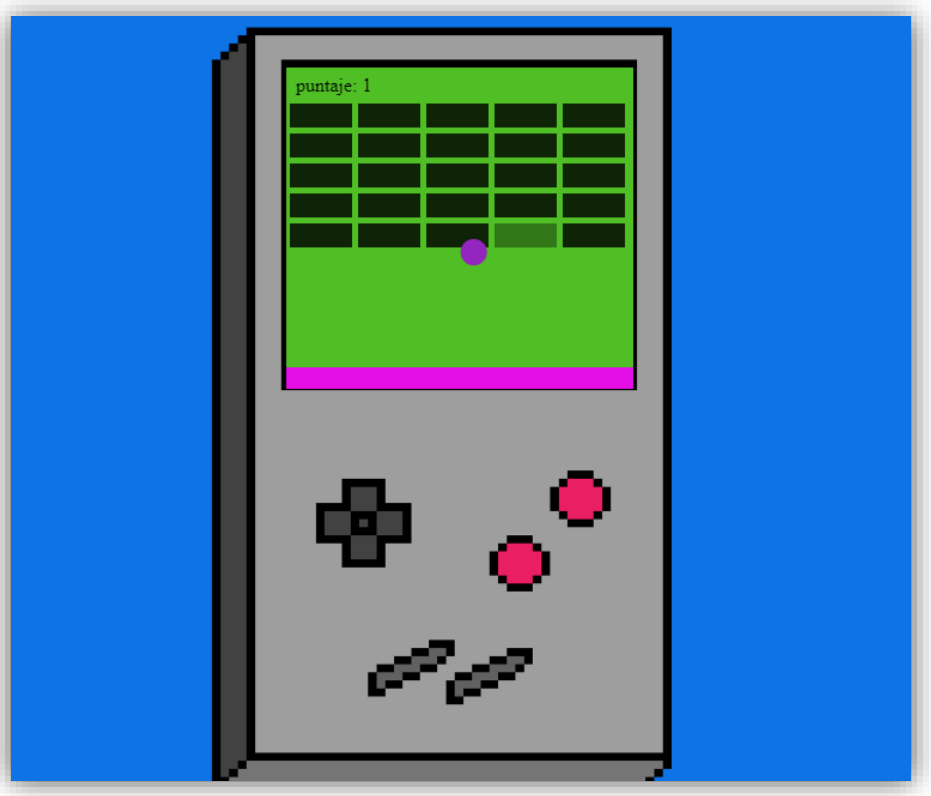
```
110.
111.                               b.estado = b.estado - 1;
112.
113.
114.
115.
116.                               b.ciclo = b.ciclo - 1;
117.
118.
119.
120.                           puntaje++;
121.                               if(puntaje ==
   nroFilasLadrillos*nroColumnasLadrillos) {
122.                                   alert("USTED GANA!
   FELICITACIONES!!!");
123.                                   document.location.reload();
124.                               }
125.
126.                           }
127.                       }
128.                   }
129.               }
130.       }
131.           function dibujarBola() {
132.               ctx.beginPath();
133.               ctx.arc(x, y, radioBola, 0, Math.PI*2);
134.               ctx.fillStyle = "#9525BF";
135.               ctx.fill();
136.               ctx.closePath();
137.           }
138.
139.           function dibujarPaleta() {
140.               ctx.beginPath();
141.               ctx.rect(paletaPosX, canvas.height-alturaPaleta,
   anchuraPaleta, alturaPaleta);
142.               ctx.fillStyle = "#E30EE6";
143.               ctx.fill();
144.               ctx.closePath();
145.           }
146.
147.          function dibujarLadrillos() {
148.
149.
150.               for(var c=0; c<nroColumnasLadrillos; c++) {
151.                   for(var r=0; r<nroFilasLadrillos; r++) {
152.
153.
154.                       if(ladrillos[c][r].estado > 0) {
155.
156.                           var b = ladrillos[c][r];
157.
158.                           var posXLadrillo =
   (r*(anchuraLadrillos+rellenoLadrillos))+vacioIzquierdoLadrillo;
```

```
159.                            var posYLadrillo =
   (c*(alturaLadrillos+rellenoLadrillos))+vacioSuperiorLadrillo;
160.
161.                            if (b.estado < 10 && b.estado > 1) {
162.
163.                                b.ciclo = b.ciclo - 1;
164.
165.                                if (b.ciclo < 1) {
166.                                    if (b.estado > 1) {
167.                                        b.estado = b.estado - 1;
168.                                        b.ciclo = numeroCiclos;
169.                                    }
170.                                }
171.                            }
172.
173.                            ladrillos[c][r].x = posXLadrillo;
174.                            ladrillos[c][r].y = posYLadrillo;
175.                            ctx.beginPath();
176.                            ctx.rect(posXLadrillo, posYLadrillo,
   anchuraLadrillos, alturaLadrillos);
177.
178.                           if (b.estado == 10) {
179.                                ctx.fillStyle = "#0F2407";
180.                            }
181.                            else if (b.estado == 9) {
182.                              ctx.fillStyle = "#14300A";
183.                            }
184.                            else if (b.estado == 8) {
185.                              ctx.fillStyle = "#193C0C";
186.                            }
187.                            else if (b.estado == 7) {
188.                              ctx.fillStyle = "#1E480E";
189.                            }
190.                            else if (b.estado == 6) {
191.                              ctx.fillStyle = "#235411";
192.                            }
193.                            else if (b.estado == 5) {
194.                              ctx.fillStyle = "#286013";
195.                            }
196.                            else if (b.estado == 4) {
197.                              ctx.fillStyle = "#317717";
198.                            }
199.                            else if (b.estado == 3) {
200.                              ctx.fillStyle = "#3B8F1C";
201.                            }
202.                            else if (b.estado == 2) {
203.                              ctx.fillStyle = "#45A720";
204.                            }
205.                            else if (b.estado == 1) {
206.                              ctx.fillStyle = "#4FBF25";
207.                            }
208.                            else {
209.                              ctx.fillStyle = "#4FBF25";
210.                            }
```

```
211.
212.                         ctx.fill();
213.                         ctx.closePath();
214.                     }
215.                 }
216.             }
217.         }
218.         function dibujarPuntaje() {
219.             ctx.font = "16px ArcadeClassic ";
220.             ctx.fillStyle = colorTexto;
221.             ctx.fillText("puntaje: "+puntaje, 8, 20);
222.         }
223.        function dibujar() {
224.            ctx.clearRect(0, 0, canvas.width, canvas.height);
225.            dibujarLadrillos();
226.            dibujarBola();
227.            dibujarPaleta();
228.            deteccionColision();
229.            dibujarPuntaje();
230.            if(x + dx > canvas.width-radioBola || x + dx <
   radioBola) {
231.                dx = -dx;
232.            }
233.            if(y + dy < radioBola) {
234.                dy = -dy;
235.            }
236.            else if(y + dy > canvas.height-radioBola) {
237.                if(x > paletaPosX && x < paletaPosX +
   anchuraPaleta) {
238.                    dy = -dy;
239.                }
240.                else {
241.                    // Detiene el ciclo del juego
242.                    clearInterval(juego);
243.                    // Genera mensaje, pues el jugador ha perdido
244.                    alert("GAME OVER");
245.                    // Recarga la página, para iniciar de nuevo
   el juego
246.                    document.location.reload();
247.                }
248.            }
249.
250.            if(flechaDerechaPresionada && paletaPosX <
   canvas.width-anchuraPaleta) {
251.                paletaPosX += 9;
252.            }
253.            else if(flechaIzquierdaPresionada && paletaPosX > 0)
   {
254.                paletaPosX -= 9;
255.            }
256.
257.            x += dx;
258.            y += dy;
259.         }
```

```
260.
261.          var juego = setInterval(dibujar, 25);
262.      </script>
263.
264.      </body>
265.      </html>
```

Al ejecutar este código se obtiene la siguiente interfaz visual:



En la figura 7 podemos observar como algunos ladrillos se desaparecieron luego de ser golpeados por la bola.

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.

9 FASE 7: CONTAR PUNTOS Y GANAR

En esta parte del programa se realiza la variable para darle algún valor cuando la bola golpee algún ladrillo y se convierta en un punto y se sumen estos puntos hasta ganar el juego con el máximo de puntos que se puedan obtener

Se crea una variable llamada puntaje la cual controla la cantidad de ladrillos que han sido golpeados por la bola así:

var puntaje = 0;

Y después se hace su función:

function dibujarPuntaje() {

ctx.font = "16px ArcadeClassic";

ctx.fillStyle = colorTexto;

ctx.fillText("puntaje: "+puntaje, 40, 20); }

Cada que la bola impacta un ladrillo se le agrega un valor a esta variable hasta que el puntaje es igual al número de ladrillos haciendo que el juego se gane. Agregando este código en la **función detectarColision**:

puntaje++; // puntaje = puntaje + 1;

if(puntaje == nroFilasLadrillos*nroColumnasLadrillos) {

alert("USTED GANA! FELICITACIONES!!!");

document.location.reload();

}

```
1. <!DOCTYPE html>
2. <html>
3. <head>
4.     <meta charset="utf-8" />
5.     <title>Juego 2D - #06 - Detección de colisión</title>
6.     <style>* { padding: 0; margin: 0; } canvas { background:
   #4FBF25;
7.     padding: 0;
8.     margin: auto;
9.     display: block;
10.          position: absolute;
11.          top: 0;
12.          bottom: 0;
13.          left: 0;
14.          right: 0;} </style>
```

```
15.      </head>
16.      <center>
17.          <img src="gor.png" height=”1000” width=445”>
18.             </center>
19.      <body style="background-color:#0E74E6 ;">
20.
21.      <canvas id="miCanvas" width="290" height="268"></canvas>
22.
23.      <script>
24.         var canvas = document.getElementById("miCanvas");
25.         var ctx = canvas.getContext("2d");
26.
27.         var radioBola = 11;
28.         var x = canvas.width/3;
29.         var y = canvas.height-30;
30.         var dx = 4;
31.         var dy = -4;
32.
33.         var alturaPaleta = 18;
34.         var anchuraPaleta = 480;
35.         var paletaPosX = (canvas.width-anchuraPaleta)/3;
36.
37.         var flechaDerechaPresionada = false;
38.         var flechaIzquierdaPresionada = false;
39.
40.         var nroFilasLadrillos = 5;
41.         var nroColumnasLadrillos = 5;
42.         var anchuraLadrillo = 52;
43.         var alturaLadrillo = 20;
44.         var rellenoLadrillo = 5;
45.         var vacioSuperiorLadrillo = 30;
46.         var vacioIzquierdoLadrillo = 3;
47.       var numeroCiclos = 10;
48.      var colorTexto = "#000000"
49.         // LA VARIABLE puntaje CONTROLA EL NÚMERO DE LADRILLOS
   QUE HAN SIDO
50.         // IMPACTADOS POR LA BOLA. Cada vez que la bola golpee un
   ladrillo,
51.         // la variable "puntaje" se incrementa en uno
52.         var puntaje = 0;
53.         var ladrillos = [];
54.         for(var c=0; c<nroColumnasLadrillos; c++) {
55.            ladrillos[c] = [];
56.            for(var f=0; f<nroFilasLadrillos; f++) {
57.
58.                     /* IMPORTANTE:
59.                                Como se puede observar, cada
   ladrillo está definido como: ==> ladrillos[c][f]
60.                                Los valores c y f, se corresponden
   con la fila y la columna, DENTRO
61.                                DE LA MATRIZ DE LADRILLOS
62.                                ---------------------------------
   ----------------------------------------
```

```
63.                                 A cada ladrillo en la posicion (c,
   f), se le asignan tres valores:
64.
65.                                    x: Su coordenada horizontal EN
   LA PANTALLA
66.                                    y: Su coordenada vertical EN LA
   PANTALLA
67.                                    status: Indica si está visible
   o invisible. 1 = Visible, 0 = INVISIBLE
68.
69.                                 Inicialmente el ladrillo debe esta
   visible. Si la bola "toca" al ladrillo,
70.                                 el ladrillo se debe volver
   INVISIBLE (status = 0)
71.                                 ----------------------------------
   ----------------------------------------
72.                                 Los valores x y y valen
   originalmente cero (0)
73.                                 Esto cambia cuando se dibujan (más
   adelante, en la función: dibujarLadrillos())
74.                              */
75.                   ladrillos[c][f] = { x: 0, y: 0, estado: 10,
   ciclo: numeroCiclos };
76.               }
77.           }
78.
79.           document.addEventListener("keydown",
   manejadorTeclaPresionada, false);
80.           document.addEventListener("keyup",
   manejadorTeclaLiberada, false);
81.
82.           function manejadorTeclaPresionada(e) {
83.               if(e.keyCode == 39) {
84.                   flechaDerechaPresionada = true;
85.               }
86.               else if(e.keyCode == 37) {
87.                   flechaIzquierdaPresionada = true;
88.               }
89.           }
90.
91.           function manejadorTeclaLiberada(e) {
92.               if(e.keyCode == 39) {
93.                   flechaDerechaPresionada = false;
94.               }
95.               else if(e.keyCode == 37) {
96.                   flechaIzquierdaPresionada = false;
97.               }
98.           }
99.
100.          // EN ESTA FUNCIÓN SE DETECTA LA COLISIÓN DE LA BOLA CON
   EL LADRILLO
101.
102.          function detectarColision() {
103.
```

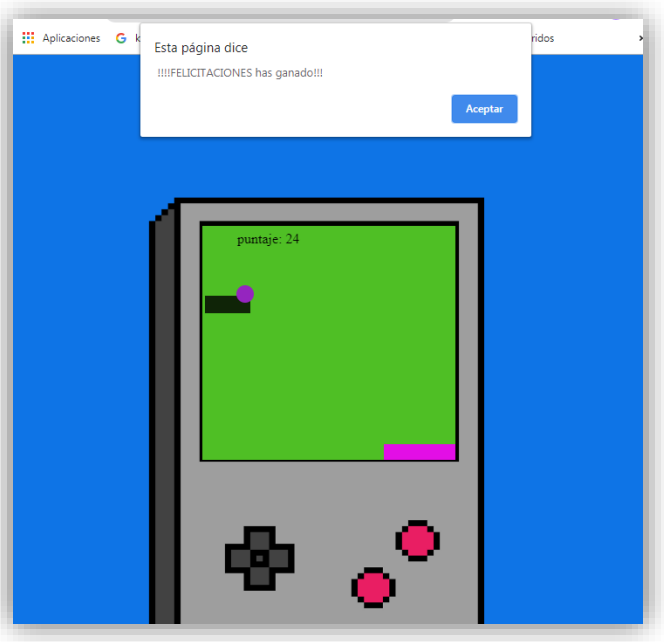
```
104.             // LOS DOS CICLOS SIGUIENTES RECORREN TODOS LOS
   LADRILLOS
105.            for(var c=0; c<nroColumnasLadrillos; c++) {
106.                 for(var f=0; f<nroFilasLadrillos; f++) {
107.
108.
109.
110.                     var b = ladrillos[c][f];
111.
112.
113.
114.                     if(b.estado == 10) {
115.
116.                         // LADRILLO NO HA RECIBIDO NINGÚN IMPACTO
   !!!!!!
117.
118.                         if(x > b.x && x < b.x+anchuraLadrillos &&
   y > b.y && y < b.y+alturaLadrillos) {
119.
120.
121.
122.                             dy = -dy;
123.
124.
125.
126.
127.                             b.estado = b.estado - 1;
128.
129.
130.
131.
132.                             b.ciclo = b.ciclo - 1;
133.
134.
135.
136.
137.                             // LA INSTRUCCIÓN puntaje++ EQUIVALE
   A: puntaje = puntaje + 1
138.                             // ---------------------------------
   ------------------------
139.                             // EN ESTE PUNTO DEL CÓDIGO LA BOLA
   HA IMPACTADO UN LADRILLO
140.                             // POR ESTE MOTIVO, SE INCREMENTA EL
   VALOR DE puntaje
141.                             // Si el puntaje es igual al número
   total de ladrillos (valor que
142.                             // se obtiene multiplicando el número
   de filas de ladrillos por el
143.                             // número de columnas de ladrillos),
   entonces el jugador ha ganado
144.                             puntaje++; // puntaje = puntaje + 1;
145.                             if(puntaje ==
   nroFilasLadrillos*nroColumnasLadrillos) {
```

```
146.                                 alert("USTED GANA!
   FELICITACIONES!!!");
147.                                 document.location.reload();
148.                             }
149.
150.                         }
151.                     }
152.                 }
153.             }
154.     }
155.         function dibujarBola() {
156.             ctx.beginPath();
157.             ctx.arc(x, y, radioBola, 0, Math.PI*2);
158.             ctx.fillStyle = "#9525BF";
159.             ctx.fill();
160.             ctx.closePath();
161.         }
162.
163.         function dibujarPaleta() {
164.             ctx.beginPath();
165.             ctx.rect(paletaPosX, canvas.height-alturaPaleta,
   anchuraPaleta, alturaPaleta);
166.             ctx.fillStyle = "#E30EE6";
167.             ctx.fill();
168.             ctx.closePath();
169.         }
170.
171.        function dibujarLadrillos() {
172.
173.
174.             for(var c=0; c<nroColumnasLadrillos; c++) {
175.                 for(var r=0; r<nroFilasLadrillos; r++) {
176.
177.
178.                     if(ladrillos[c][r].estado > 0) {
179.
180.                         var b = ladrillos[c][r];
181.
182.                         var posXLadrillo =
   (r*(anchuraLadrillos+rellenoLadrillos))+vacioIzquierdoLadrillo;
183.                         var posYLadrillo =
   (c*(alturaLadrillos+rellenoLadrillos))+vacioSuperiorLadrillo;
184.
185.                         if (b.estado < 10 && b.estado > 1) {
186.
187.                             b.ciclo = b.ciclo - 1;
188.
189.                             if (b.ciclo < 1) {
190.                                 if (b.estado > 1) {
191.                                     b.estado = b.estado - 1;
192.                                     b.ciclo = numeroCiclos;
193.                                 }
194.                             }
195.                         }
```

```
196.
197.                         ladrillos[c][r].x = posXLadrillo;
198.                         ladrillos[c][r].y = posYLadrillo;
199.                         ctx.beginPath();
200.                         ctx.rect(posXLadrillo, posYLadrillo,
   anchuraLadrillos, alturaLadrillos);
201.
202.                        if (b.estado == 10) {
203.                             ctx.fillStyle = "#0F2407";
204.                         }
205.                         else if (b.estado == 9) {
206.                           ctx.fillStyle = "#14300A";
207.                         }
208.                         else if (b.estado == 8) {
209.                           ctx.fillStyle = "#193C0C";
210.                         }
211.                         else if (b.estado == 7) {
212.                           ctx.fillStyle = "#1E480E";
213.                         }
214.                         else if (b.estado == 6) {
215.                           ctx.fillStyle = "#235411";
216.                         }
217.                         else if (b.estado == 5) {
218.                           ctx.fillStyle = "#286013";
219.                         }
220.                         else if (b.estado == 4) {
221.                           ctx.fillStyle = "#317717";
222.                         }
223.                         else if (b.estado == 3) {
224.                           ctx.fillStyle = "#3B8F1C";
225.                         }
226.                         else if (b.estado == 2) {
227.                           ctx.fillStyle = "#45A720";
228.                         }
229.                         else if (b.estado == 1) {
230.                           ctx.fillStyle = "#4FBF25";
231.                         }
232.                         else {
233.                           ctx.fillStyle = "#4FBF25";
234.                         }
235.
236.                         ctx.fill();
237.                         ctx.closePath();
238.                     }
239.                 }
240.             }
241.         }
242.         function dibujarPuntaje() {
243.             ctx.font = "16px ArcadeClassic ";
244.             ctx.fillStyle = colorTexto;
245.             ctx.fillText("puntaje: "+puntaje, 8, 20);
246.         }
247.        function dibujar() {
248.            ctx.clearRect(0, 0, canvas.width, canvas.height);
```

```
249.             dibujarLadrillos();
250.             dibujarBola();
251.             dibujarPaleta();
252.             deteccionColision();
253.             dibujarPuntaje();
254.             if(x + dx > canvas.width-radioBola || x + dx <
   radioBola) {
255.                 dx = -dx;
256.             }
257.             if(y + dy < radioBola) {
258.                 dy = -dy;
259.             }
260.             else if(y + dy > canvas.height-radioBola) {
261.                 if(x > paletaPosX && x < paletaPosX +
   anchuraPaleta) {
262.                     dy = -dy;
263.                 }
264.                 else {
265.                     // Detiene el ciclo del juego
266.                     clearInterval(juego);
267.                     // Genera mensaje, pues el jugador ha perdido
268.                     alert("GAME OVER");
269.                     // Recarga la página, para iniciar de nuevo
   el juego
270.                     document.location.reload();
271.                 }
272.             }
273.
274.             if(flechaDerechaPresionada && paletaPosX <
   canvas.width-anchuraPaleta) {
275.                 paletaPosX += 9;
276.             }
277.             else if(flechaIzquierdaPresionada && paletaPosX > 0)
   {
278.                 paletaPosX -= 9;
279.             }
280.
281.             x += dx;
282.             y += dy;
283.         }
284.
285.         var juego = setInterval(dibujar, 25);
286.     </script>
287.
288.     </body>
289.     </html>
```

Al ejecutar este código se obtiene la siguiente interfaz visual:



En la figura 8 se puede observar como la bola al impactar en los ladrillos estos desaparecen y el puntaje incrementa hasta desaparecer todos los ladrillos y ganar el juego.

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.

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)** que da el siguiente código:

```
function manejadorRaton(e) {
    var posXRatonDentroDeCanvas = e.clientX - canvas.offsetLeft;
    if(posXRatonDentroDeCanvas > 0 && posXRatonDentroDeCanvas < canvas.width) {
      paletaPosX = posXRatonDentroDeCanvas - anchuraPaleta/2;
    }
  }
```

A la cual se le da una variable y una condición que al cumplirla hace que la paleta pueda desplazada mediante el mouse:

```
document.addEventListener("mousemove", manejadorRaton, false);
```

```
1. <!DOCTYPE html>
2. <html>
3. <head>
4.     <meta charset="utf-8" />
5.     <title>Juego 2D - #06 - Detección de colisión</title>
6.     <style>* { padding: 0; margin: 0; } canvas { background:
   #4FBF25;
7.     padding: 0;
8.     margin: auto;
9.     display: block;
10.          position: absolute;
11.          top: 0;
12.          bottom: 0;
13.          left: 0;
14.          right: 0;} </style>
15.     </head>
16.     <center>
17.          <img src="gor.png" height=”1000” width=445”>
18.             </center>
19.     <body style="background-color:#0E74E6 ;">
20.
21.     <canvas id="miCanvas" width="290" height="268"></canvas>
22.
23.     <script>
24.          var canvas = document.getElementById("miCanvas");
```

```
25.          var ctx = canvas.getContext("2d");
26.
27.          var radioBola = 11;
28.          var x = canvas.width/3;
29.          var y = canvas.height-30;
30.          var dx = 4;
31.          var dy = -4;
32.
33.          var alturaPaleta = 18;
34.          var anchuraPaleta = 480;
35.          var paletaPosX = (canvas.width-anchuraPaleta)/3;
36.
37.          var flechaDerechaPresionada = false;
38.          var flechaIzquierdaPresionada = false;
39.
40.          var nroFilasLadrillos = 5;
41.          var nroColumnasLadrillos = 5;
42.          var anchuraLadrillo = 52;
43.          var alturaLadrillo = 20;
44.          var rellenoLadrillo = 5;
45.          var vacioSuperiorLadrillo = 30;
46.          var vacioIzquierdoLadrillo = 3;
47.       var numeroCiclos = 10;
48.      var colorTexto = "#000000"
49.
50.          var ladrillos = [];
51.          for(var c=0; c<nroColumnasLadrillos; c++) {
52.              ladrillos[c] = [];
53.              for(var f=0; f<nroFilasLadrillos; f++) {
54.
55.                  ladrillos[c][f] = { x: 0, y: 0, estado: 10,
   ciclo: numeroCiclos };
56.
57.              }
58.          }
59.          document.addEventListener("keydown",
   manejadorTeclaPresionada, false);
60.          document.addEventListener("keyup",
   manejadorTeclaLiberada, false);
61.       document.addEventListener("mousemove", manejadorRaton,
   false);
62.          function manejadorTeclaPresionada(e) {
63.              if(e.keyCode == 39) {
64.                  flechaDerechaPresionada = true;
65.              }
66.              else if(e.keyCode == 37) {
67.                  flechaIzquierdaPresionada = true;
68.              }
69.          }
70.
71.          function manejadorTeclaLiberada(e) {
72.              if(e.keyCode == 39) {
73.                  flechaDerechaPresionada = false;
74.              }
```

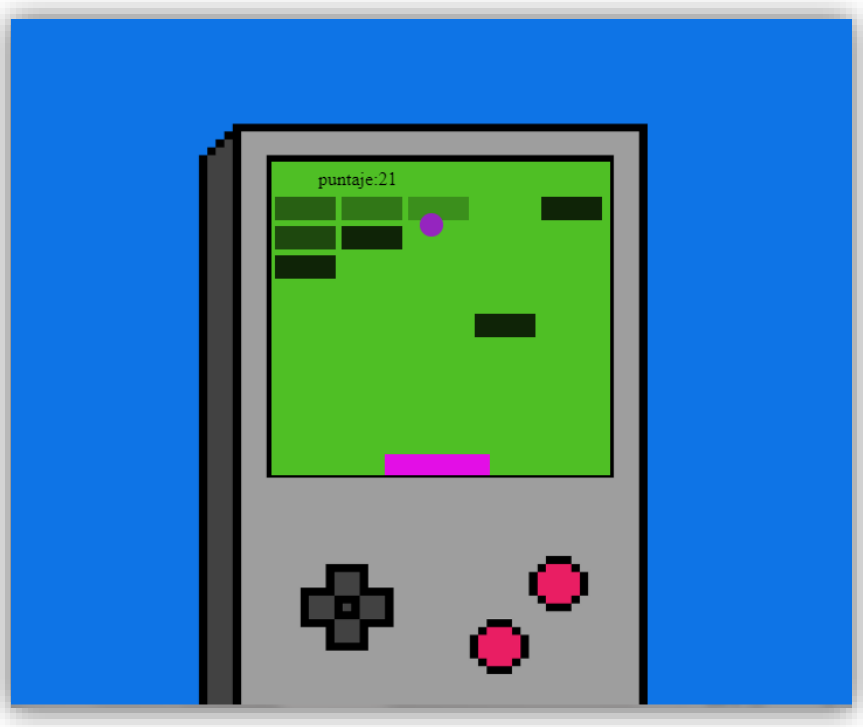
```
75.             else if(e.keyCode == 37) {
76.                 flechaIzquierdaPresionada = false;
77.             }
78.         }
79.     function manejadorRaton(e) {
80.             var posXRatonDentroDeCanvas = e.clientX -
   canvas.offsetLeft;
81.             if(posXRatonDentroDeCanvas > 0 &&
   posXRatonDentroDeCanvas < canvas.width) {
82.                 paletaPosX = posXRatonDentroDeCanvas -
   anchuraPaleta/2;
83.             }
84.         }
85.
86.
87.
88.         function detectarColision() {
89.
90.                   for(var c=0; c<nroColumnasLadrillos; c++) {
91.                 for(var f=0; f<nroFilasLadrillos; f++) {
92.
93.
94.
95.                     var b = ladrillos[c][f];
96.
97.
98.
99.                     if(b.estado == 10) {
100.
101.
102.                        if(x > b.x && x < b.x+anchuraLadrillos &&
   y > b.y && y < b.y+alturaLadrillos) {
103.
104.
105.
106.                            dy = -dy;
107.
108.
109.
110.
111.                            b.estado = b.estado - 1;
112.
113.
114.
115.
116.                            b.ciclo = b.ciclo - 1;
117.
118.
119.
120.                        puntaje++;
121.                            if(puntaje ==
   nroFilasLadrillos*nroColumnasLadrillos) {
122.                                alert("USTED GANA!
   FELICITACIONES!!!");
```

```
123.                                 document.location.reload();
124.                             }
125.
126.                         }
127.                     }
128.                 }
129.             }
130.     }
131.         function dibujarBola() {
132.             ctx.beginPath();
133.             ctx.arc(x, y, radioBola, 0, Math.PI*2);
134.             ctx.fillStyle = "#9525BF";
135.             ctx.fill();
136.             ctx.closePath();
137.         }
138.
139.         function dibujarPaleta() {
140.             ctx.beginPath();
141.             ctx.rect(paletaPosX, canvas.height-alturaPaleta,
    anchuraPaleta, alturaPaleta);
142.             ctx.fillStyle = "#E30EE6";
143.             ctx.fill();
144.             ctx.closePath();
145.         }
146.
147.        function dibujarLadrillos() {
148.
149.
150.             for(var c=0; c<nroColumnasLadrillos; c++) {
151.                 for(var r=0; r<nroFilasLadrillos; r++) {
152.
153.
154.                     if(ladrillos[c][r].estado > 0) {
155.
156.                         var b = ladrillos[c][r];
157.
158.                         var posXLadrillo =
    (r*(anchuraLadrillos+rellenoLadrillos))+vacioIzquierdoLadrillo;
159.                         var posYLadrillo =
    (c*(alturaLadrillos+rellenoLadrillos))+vacioSuperiorLadrillo;
160.
161.                         if (b.estado < 10 && b.estado > 1) {
162.
163.                             b.ciclo = b.ciclo - 1;
164.
165.                             if (b.ciclo < 1) {
166.                                 if (b.estado > 1) {
167.                                     b.estado = b.estado - 1;
168.                                     b.ciclo = numeroCiclos;
169.                                 }
170.                             }
171.                         }
172.
173.                         ladrillos[c][r].x = posXLadrillo;
```

```
174.                         ladrillos[c][r].y = posYLadrillo;
175.                         ctx.beginPath();
176.                         ctx.rect(posXLadrillo, posYLadrillo,
   anchuraLadrillos, alturaLadrillos);
177.
178.                        if (b.estado == 10) {
179.                             ctx.fillStyle = "#0F2407";
180.                         }
181.                         else if (b.estado == 9) {
182.                           ctx.fillStyle = "#14300A";
183.                         }
184.                         else if (b.estado == 8) {
185.                           ctx.fillStyle = "#193C0C";
186.                         }
187.                         else if (b.estado == 7) {
188.                           ctx.fillStyle = "#1E480E";
189.                         }
190.                         else if (b.estado == 6) {
191.                           ctx.fillStyle = "#235411";
192.                         }
193.                         else if (b.estado == 5) {
194.                           ctx.fillStyle = "#286013";
195.                         }
196.                         else if (b.estado == 4) {
197.                           ctx.fillStyle = "#317717";
198.                         }
199.                         else if (b.estado == 3) {
200.                           ctx.fillStyle = "#3B8F1C";
201.                         }
202.                         else if (b.estado == 2) {
203.                           ctx.fillStyle = "#45A720";
204.                         }
205.                         else if (b.estado == 1) {
206.                           ctx.fillStyle = "#4FBF25";
207.                         }
208.                         else {
209.                           ctx.fillStyle = "#4FBF25";
210.                         }
211.
212.                         ctx.fill();
213.                         ctx.closePath();
214.                     }
215.                 }
216.             }
217.         }
218.         function dibujarPuntaje() {
219.             ctx.font = "16px ArcadeClassic ";
220.             ctx.fillStyle = colorTexto;
221.             ctx.fillText("puntaje: "+puntaje, 8, 20);
222.         }
223.        function dibujar() {
224.            ctx.clearRect(0, 0, canvas.width, canvas.height);
225.            dibujarLadrillos();
226.            dibujarBola();
```

```
227.             dibujarPaleta();
228.             deteccionColision();
229.             dibujarPuntaje();
230.             if(x + dx > canvas.width-radioBola || x + dx <
   radioBola) {
231.                 dx = -dx;
232.             }
233.             if(y + dy < radioBola) {
234.                 dy = -dy;
235.             }
236.             else if(y + dy > canvas.height-radioBola) {
237.                  if(x > paletaPosX && x < paletaPosX +
   anchuraPaleta) {
238.                     dy = -dy;
239.                  }
240.                  else {
241.                      // Detiene el ciclo del juego
242.                      clearInterval(juego);
243.                      // Genera mensaje, pues el jugador ha perdido
244.                      alert("GAME OVER");
245.                      // Recarga la página, para iniciar de nuevo
   el juego
246.                      document.location.reload();
247.                  }
248.              }
249.
250.              if(flechaDerechaPresionada && paletaPosX <
   canvas.width-anchuraPaleta) {
251.                 paletaPosX += 9;
252.             }
253.             else if(flechaIzquierdaPresionada && paletaPosX > 0)
   {
254.                 paletaPosX -= 9;
255.             }
256.
257.             x += dx;
258.             y += dy;
259.         }
260.
261.         var juego = setInterval(dibujar, 25);
262.     </script>
263.
264.     </body>
265.     </html>
```

Al ejecutar este código se obtiene la siguiente interfaz visual:



En la figura 9 se puede observar como la paleta es controlada de derecha a izquierda mediante el uso del mouse

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, crea el código así:

```
function dibujarVidas() {
    ctx.font = "16px ArcadeClassic";
    ctx.fillStyle = colorTexto;
        ctx.fillText("vidas: "+vidas, canvas.width-65, 20);
  }
```

Además se le agregan un junto con los condicionales lo siguiente:

```
vidas--;
 if(!vidas) {
alert("GAME OVER");
document.location.reload();}
            else {
             x = canvas.width/2;
               y = canvas.height-30;
               dx = 4;
               dy = -4;
               paletaPosX = (canvas.width-anchuraPaleta)/2}}}
```

Si se produce un contacto de la bola con la base del canvas, se pierde una vida. Para ello, la instrucción **vidas--;lo** cual equivale a: **vidas = vidas - 1**

Y se crea otra variable **canvas.style.cursor = 'none'** para ocultar el mouse dentro del campo del juego, también se crea la instrucción vidas; la cual lleva la cuenta de las vidas que tiene y que ha perdido.

```
1. <!DOCTYPE html>
2. <html>
3. <head>
4.     <meta charset="utf-8" />
5.     <title>Juego 2D - #09 - Juego completo</title>
6.     <!-- 1. Se oculta el ratón
7.          2. Se agregan vidas al jugador
8.          3. Ya no se utiliza "setInterval" -->
9.     <style>* { padding: 0; margin: 0; } canvas { background:
   #4FBF25;
10.          padding: 0;
11.          margin: auto;
12.          display: block;
13.          position: absolute;
14.          top: 0;
15.          bottom: 0;
16.          left: 0;
17.          right: 0;} } </style>
18.      </head>
19.      <center>
20.           <img src="gor.png" height=”1000” width=445”>
21.              </center>
22.      <body style="background-color:#0E74E6 ;">
23.      <canvas id="miCanvas" width="290" height="268"></canvas>
24.
25.      <script>
26.          var canvas = document.getElementById("miCanvas");
27.          var ctx = canvas.getContext("2d");
28.
29.          var bolaRadio = 10;
30.          var x = canvas.width/2;
31.          var y = canvas.height-30;
32.          var dx = 4;
33.          var dy = -4;
34.
35.          var alturaPaleta = 18;
36.          var anchuraPaleta = 90;
37.          var paletaPosX = (canvas.width-anchuraPaleta)/2;
38.
39.          var flechaDerechaPresionada = false;
40.          var flechaIzquierdaPresionada = false;
41.
42.          var nroFilasLadrillos = 5;
43.          var nroColumnasLadrillos = 5;
44.          var anchuraLadrillo = 52;
45.          var alturaLadrillo = 20;
46.          var rellenoLadrillo = 5;
47.          var vacioSuperiorLadrillo = 30;
48.          var vacioIzquierdoLadrillo = 3;
49.
50.          var puntaje = 0;
51.          var numeroCiclos = 10;
52.          // ESTA INSTRUCCIÓN CONTROLA EL NÚMERO DE VIDAS DEL
   JUGADOR
```

```
53.          // CUANDO LA INSTRUCCIÓN vidas DISMINUYE A CERO, EL
   JUGADOR PIERDE,
54.          // PUESTO QUE HA PERDIDO TRES VECES
55.          var vidas = 3;
56.
57.          // ESTA VARIABLE DEFINE UN COLOR
58.          // Se pueden utilizar otros colores para los diferentes
   elementos del juego
59.          var colorFigura = "#ff0000";
60.          var colorBola = "#9525BF";
61.          var colorPaleta = "#E30EE6";
62.          var colorTexto = "#d63535";
63.
64.          // ESTA INSTRUCCIÓN OCULTA EL CURSOR DEL RATON (DENTRO
   DEL CANVAS)
65.          canvas.style.cursor = 'none';
66.
67.          var ladrillos = [];
68.          for(var c=0; c<nroColumnasLadrillos; c++) {
69.             ladrillos[c] = [];
70.             for(var f=0; f<nroFilasLadrillos; f++) {
71.                 ladrillos[c][f] = { x: 0, y: 0, estado: 10, ciclo:
   numeroCiclos };
72.             }
73.         }
74.
75.         document.addEventListener("keydown",
   manejadorTeclaPresionada, false);
76.         document.addEventListener("keyup", manejadorTeclaLiberada,
   false);
77.         document.addEventListener("mousemove", manejadorRaton,
   false);
78.
79.         function manejadorTeclaPresionada(e) {
80.             if(e.keyCode == 39) {
81.                 flechaDerechaPresionada = true;
82.             }
83.             else if(e.keyCode == 37) {
84.                 flechaIzquierdaPresionada = true;
85.             }
86.         }
87.
88.         function manejadorTeclaLiberada(e) {
89.             if(e.keyCode == 39) {
90.                 flechaDerechaPresionada = false;
91.             }
92.             else if(e.keyCode == 37) {
93.                 flechaIzquierdaPresionada = false;
94.             }
95.         }
96.
97.         function manejadorRaton(e) {
98.             var posXRatonDentroDeCanvas = e.clientX -
   canvas.offsetLeft;
```

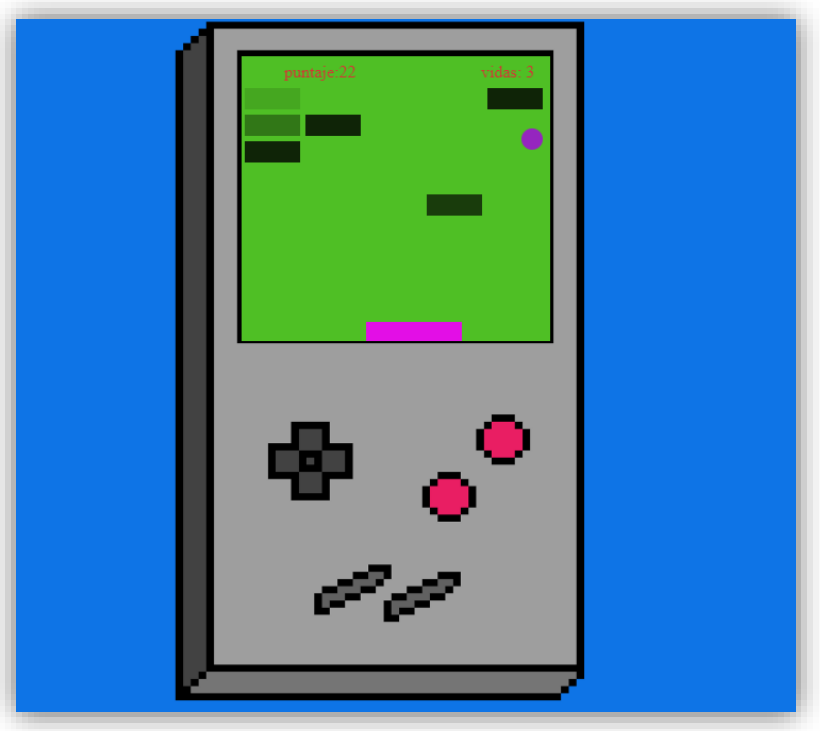
```
99.             if(posXRatonDentroDeCanvas > 0 &&
   posXRatonDentroDeCanvas < canvas.width) {
100.                paletaPosX = posXRatonDentroDeCanvas -
   anchuraPaleta/2;
101.            }
102.        }
103.
104.        function detectarColision() {
105.            for(var c=0; c<nroColumnasLadrillos; c++) {
106.                for(var f=0; f<nroFilasLadrillos; f++) {
107.                    var b = ladrillos[c][f];
108.                    if(b.estado == 10) {
109.
110.                        if(x > b.x && x < b.x+anchuraLadrillos &&
   y > b.y && y < b.y+alturaLadrillos) {
111.                            dy = -dy;
112.                            b.estado = b.estado - 1;
113.
114.                            b.ciclo = b.ciclo - 1;
115.
116.                            puntaje++;
117.                            if(puntaje ==
   nroFilasLadrillos*nroColumnasLadrillos) {
118.                                alert(" !!!FELICITACIONES HAS
   GANADO EL JUEGO!!!");
119.                                document.location.reload();
120.                            }
121.
122.                        }
123.                    }
124.                }
125.            }
126.        }
127.        function dibujarBola() {
128.            ctx.beginPath();
129.            ctx.arc(x, y, bolaRadio, 0, Math.PI*2);
130.            // SE UTILIZA EL COLOR PREVIAMENTE DEFINIDO
131.            ctx.fillStyle = colorBola;
132.            ctx.fill();
133.            ctx.closePath();
134.        }
135.        function dibujarPaleta() {
136.            ctx.beginPath();
137.            ctx.rect(paletaPosX, canvas.height-alturaPaleta,
   anchuraPaleta, alturaPaleta);
138.            ctx.fillStyle = colorPaleta;
139.            ctx.fill();
140.            ctx.closePath();
141.        }
142.        function dibujarLadrillos() {
143.            for(var c=0; c<nroColumnasLadrillos; c++) {
144.                for(var f=0; f<nroFilasLadrillos; f++) {
145.                    if(ladrillos[c][r].estado > 0) {
146.
```

```
147.                            var b = ladrillos[c][r];
148.
149.                            var posXLadrillo =
   (r*(anchuraLadrillos+rellenoLadrillos))+vacioIzquierdoLadrillo;
150.                            var posYLadrillo =
   (c*(alturaLadrillos+rellenoLadrillos))+vacioSuperiorLadrillo;
151.
152.                            if (b.estado < 10 && b.estado > 1) {
153.
154.                                b.ciclo = b.ciclo - 1;
155.
156.                                if (b.ciclo < 1) {
157.                                    if (b.estado > 1) {
158.                                        b.estado = b.estado - 1;
159.                                        b.ciclo = numeroCiclos;
160.                                    }
161.                                }
162.                            }
163.
164.                            ladrillos[c][r].x = posXLadrillo;
165.                            ladrillos[c][r].y = posYLadrillo;
166.                            ctx.beginPath();
167.                            ctx.rect(posXLadrillo, posYLadrillo,
   anchuraLadrillos, alturaLadrillos);
168.
169.                           if (b.estado == 10) {
170.                                ctx.fillStyle = "#0F2407";
171.                            }
172.                            else if (b.estado == 9) {
173.                              ctx.fillStyle = "#14300A";
174.                            }
175.                            else if (b.estado == 8) {
176.                              ctx.fillStyle = "#193C0C";
177.                            }
178.                            else if (b.estado == 7) {
179.                              ctx.fillStyle = "#1E480E";
180.                            }
181.                            else if (b.estado == 6) {
182.                              ctx.fillStyle = "#235411";
183.                            }
184.                            else if (b.estado == 5) {
185.                              ctx.fillStyle = "#286013";
186.                            }
187.                            else if (b.estado == 4) {
188.                              ctx.fillStyle = "#317717";
189.                            }
190.                            else if (b.estado == 3) {
191.                              ctx.fillStyle = "#3B8F1C";
192.                            }
193.                            else if (b.estado == 2) {
194.                              ctx.fillStyle = "#45A720";
195.                            }
196.                            else if (b.estado == 1) {
197.                              ctx.fillStyle = "#4FBF25";
```

```
198.                         }
199.                         else {
200.                           ctx.fillStyle = "#4FBF25";
201.                         }
202.
203.                         ctx.fill();
204.                         ctx.closePath();
205.                     }
206.                 }
207.             }
208.         }
209.         function dibujarPuntaje() {
210.             ctx.font = "16px ArcadeClassic";
211.             ctx.fillStyle = colorTexto;
212.             ctx.fillText("puntaje: "+puntaje, 10, 20);
213.         }
214.
215.         function dibujarVidas() {
216.             ctx.font = "16px ArcadeClassic";
217.             ctx.fillStyle = colorTexto;
218.             // SE MUESTRA EL NÚMERO DE VIDAS DISPONIBLES
219.             ctx.fillText("vidas: "+vidas, canvas.width-65, 20);
220.         }
221.
222.         function dibujar() {
223.             ctx.clearRect(0, 0, canvas.width, canvas.height);
224.             dibujarLadrillos();
225.             dibujarBola();
226.             dibujarPaleta();
227.             dibujarPuntaje();
228.             dibujarVidas();
229.             detectarColision();
230.
231.             if(x + dx > canvas.width-bolaRadio || x + dx <
    bolaRadio) {
232.                 dx = -dx;
233.             }
234.             if(y + dy < bolaRadio) {
235.                 dy = -dy;
236.             }
237.             else if(y + dy > canvas.height-bolaRadio) {
238.                 if(x > paletaPosX && x < paletaPosX +
    anchuraPaleta) {
239.                     dy = -dy;
240.                 }
241.                 else {
242.                     // SI SE PRODUCE UN CONTACTO DE LA BOLA CON
    LA BASE DEL CANVAS
243.                     // SE PIERDE UNA VIDA. PARA ELLO, LA
    INSTRUCCIÓN vidas--;
244.                     // LO CUAL EQUIVALE A: vidas = vidas - 1
245.                     vidas--;
246.                     if(!vidas) {
```

```
247.                                // SI vidas == 0 (lo cual también
   puede escribir: !vidas)
248.                                // EL JUGADOR HA PERDIDO
249.                            alert("GAME OVER");
250.                            document.location.reload();
251.                        }
252.                        else {
253.                                // SI vidas > 0 (diferente de CERO)
   EL JUEGO CONTINUA
254.                            x = canvas.width/2;
255.                            y = canvas.height-30;
256.                            dx = 4;
257.                            dy = -4;
258.                            paletaPosX = (canvas.width-
   anchuraPaleta)/2;
259.                        }
260.                    }
261.                }
262.
263.                if(flechaDerechaPresionada && paletaPosX <
   canvas.width-anchuraPaleta) {
264.                    paletaPosX += 9;
265.                }
266.                else if(flechaIzquierdaPresionada && paletaPosX > 0)
   {
267.                    paletaPosX -= 9;
268.                }
269.
270.                x += dx;
271.                y += dy;
272.
273.                // ESTE ES UN SEGUNDO MÉTODO PARA REALIZAR LA
   ANIMACIÓN DEL JUEGO
274.                // LA INSTRUCCIÓN: requestAnimationFrame SE EJECUTA
   60 VECES POR SEGUNDO
275.                // Y AL EJECUTARSE LLAMA A LA FUNCIÓN ENTRE
   PARÉNTESIS
276.                // POR TANTO, dibujar SE EJECUTA 60 VECES POR SEGUNDO
277.                // GENERANDO EL CICLO DEL JUEGO
278.                requestAnimationFrame(dibujar);
279.            }
280.
281.            dibujar();
282.        </script>
283.
284.        </body>
285.        </html>
```

Al ejecutar este código se obtiene la siguiente interfaz visual:



En la imagen 10 podemos observar el juego ya completado totalmente, y en el podemos observar las vidas y el puntaje que lleva el jugador durante el juego y la desaparición del mando dentro del canvas.

Este el prototipo final del proyecto, aquí se aplicó todo lo mostrado en la guía y modificado en varios de sus aspectos.

12 CONCLUSIONES

En conclusión podemos observar como después de seguir una cierta cantidad de pasos pudimos llegar a nuestro objetivo el cual era construir un juego en 2D.

Este juego realizado a través de un código html asignado a JavaSvript, en el cual usando las herramientas prestadas por html y creando y probando las funciones correctas con sus variables y problemas que surgen dentro de este código podemos llegar a tener un juego en la red virtual, combinado sus variables, funciones y estilo propio de la persona.

Este es un juego que nos ayuda a mejorar en el aprendizaje dentro del campo de la programación tanto con el lenguaje html como con tantos lenguajes que existe el día de hoy en el campo de la programación; este es un ejemplo en que podemos utilizar un programa simple (html) en hacer variable programas .

13 BIBLIOGRAFÍA

https://developer.mozilla.org/es/docs/Games/Workflows/Famoso_juego_2D_usando_JavaScript_puro/Construye_grupo_bloques