

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

JAIDER ALBERTO RENDÓN MORENO
OCTUBRE DE 2020



1 CONTENIDO

1	CONTENIDO.....	1
2	PRESENTACIÓN.....	2
3	FASE 1: Dibujar y mover un cuadrado.....	3
4	FASE 2: Rebotando en las paredes.....	6
5	FASE 3: Control de la pala y el tedado.....	9
6	FASE 4: Fin del juego.....	13
7	FASE 5: Muro de ladrillos.....	17
8	FASE 6: Detección de colisiones.....	22
9	FASE 7: Contar puntos y ganar.....	28
10	FASE 8: Controlando el ratón.....	36
11	FASE 9: Finalizando el juego.....	45
12	CONCLUSIONES.....	54
13	BIBLIOGRAFÍA.....	55

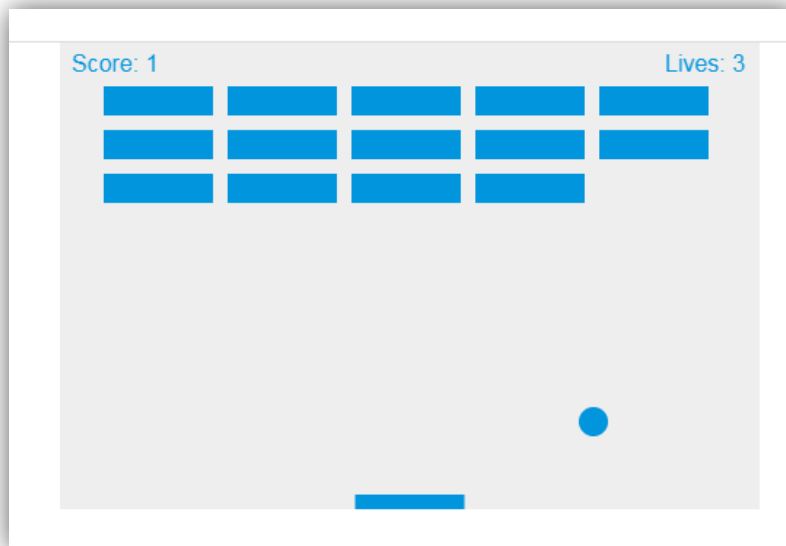
2 PRESENTACIÓN

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

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

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

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



Gráfica 1. Juego 2D en JavaScript

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

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

AUTOR: JAIDER ALBERTO RENDÓN MORENO

3 FASE 1: DIBUJAR Y MOVER UN CUADRADO

Lo primero que debemos hacer es elaborar una página en formato HTML, en esta agregamos un elemento CANVAS que en este caso contará con unas dimensiones de (800x600), este será la base de donde se desarrollara el juego 2D, en este caso se usó una imagen de fondo dentro del CANVAS por medio de la etiqueta `style>...</style>`

Lo que utilizaremos para llevar a cabo esto será operar por medio de JavaScript sobre el elemento CANVAS para esto utilizaremos las etiquetas `<script>...</script>`

Luego de estar definido el estilo del CANVAS se comienza a desarrollar la codificación del JavaScript donde empezaremos con la definición de las variables de las coordenadas respecto a la posición del cuadrado y de los valores que cambiarán su posición.

Es importante determinar tres funciones que harán posible el movimiento del cuadrado, estas son, `ProyectilVoid()`, que se encarga de insertar una imagen en las coordenadas x, y esta imagen será el cuadrado que utilizaremos para romper los ladrillos. La siguiente función es `Dibujarvoid()`, que se encarga de limpiar el CANVAS y de cambiar las coordenadas de aparición del cuadrado. Y para finalizar esta la función `setInterval(Dibujarvoid, 10)`; que se encarga de llamar la función `Dibujarvoid()` cada 10 milisegundos.

El código fuente es el siguiente:

```
1. <!DOCTYPE html>
2. <html>
3. <head>
4.     <title>Juego 2D: JavaScript VOID</title>
5.     <style>
6.         * {
7.             padding: 0;
8.             margin: 0;
9.         }
10.        canvas {
11.            background-image: url(img/FONDO.png); /* Imagen
de fondo para el CANVAS*/
12.            display: block;
13.            margin: 0 auto;
14.        }
15.    </style>
16. </head>
17. <body>
18.     <center>
19.         <p style="font-family:Comic Sans MS, cursive;font-size:
40px;">JUEGO <span style="color:#FF0000">ROMPER TODOS</span> LOS
LADRILLOS</p>
20.     </center>
21.
```

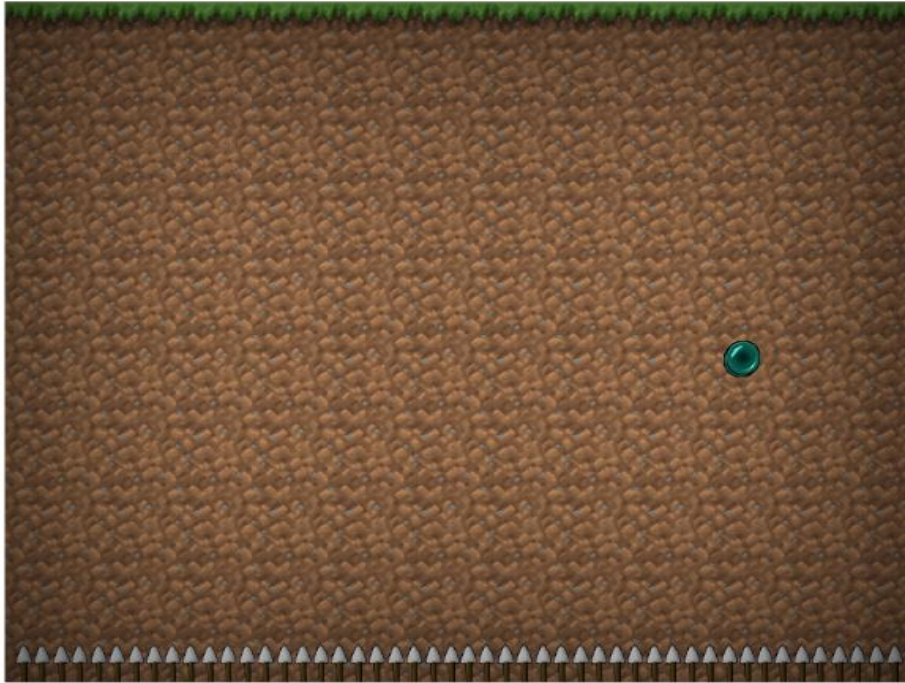
```

22.     <canvas id="miCanvas" width="800" height="600"></canvas> <!--
      Aquí se agrega el Elemento CANVAS-->
23.
24.     <script> //Inicio del JavaScript
25.     var canvas = document.getElementById("miCanvas");
26.     var ctx = canvas.getContext("2d");
27.
28.     var x=canvas.width/2;    //Coordenadas inicio horizontal
29.     var y=canvas.height -65; //Coordenadas inicio vertical
30.
31.     var dx=2; //Dirección del cuadrado respecto a x
32.     var dy=-2; //Dirección del cuadrado respecto a y
33.
34.     function ProyectilVoid() { //Función de creación del cuadrado
35.     var img = new Image();
36.     img.src = "img/PROYECTIL.png"; //Imagen utilizada
37.     ctx.drawImage(img, x, y); //Posición de aparición de la
      imagen
38.     img.onload = function(){
39.     ctx.drawImage(img, x, y);
40.     }
41.     }
42.     function Dibujarvoid() { //Función de llamado a la anterior
      función
43.     ctx.clearRect(0, 0, canvas.width, canvas.height); //Función
      para limpiar el CANVAS
44.     ProyectilVoid(); //Llamado de función
45.     x = x + dx; //Nuevo valor para x
46.     y = y + dy; //Nuevo valor para y
47.     }
48.
49.     setInterval(Dibujarvoid, 10); //Llamado de la función
      anterior cada diez milisegundos.
50. </script>

```

Al ejecutar este código se obtiene la siguiente interfaz visual:

JUEGO ROMPER TODOS LOS LADRILLOS



Gráfica 2. La interfaz inicial del juego

En la gráfica 2 se aprecia la imagen del cuadrado, y la secuencia de movimiento a partir de los incrementos en X y Y que fueron definidos. Cabe aclarar que las imágenes utilizadas están en una carpeta determinada “img” y en esta se encuentran elementos como “FONDO.png” que como su nombre lo indica es el fondo del CANVAS y “PROYECTIL.png” que es la imagen del cuadrado.

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

Esta fase consiste en definir los límites de la pelota en la pantalla, tomando en cuenta el tamaño del radio de la pelota con el ancho y alto del CANVAS.

La primera condición es `if(x + dx > 775 || x + dx < 0) { dx = -dx;}`, esta condición crea el rango horizontal al que la pelota se puede desplazar.

La segunda condición es `if(y + dy > 540 || y + dy < 0) { dy = -dy;}`, esta condición crea el rango vertical al que la pelota se puede desplazar.

```

1. <!DOCTYPE html>
2. <html>
3. <head>
4.     <title>Juego 2D: JavaScript VOID</title>
5.     <style>
6.         * {
7.             padding: 0;
8.             margin: 0;
9.         }
10.        canvas {
11.            background-image: url(img/FONDO.png); /*
Imagen de fondo para el CANVAS*/
12.            display: block;
13.            margin: 0 auto;
14.        }
15.    </style>
16. </head>
17. <body>
18.     <center>
19.         <p style="font-family:Comic Sans MS, cursive;font-
size: 40px;">JUEGO <span style="color:#FF0000">ROMPER
TODOS</span> LOS LADRILLOS</p>
20.     </center>
21.
22.     <canvas id="miCanvas" width="800"
height="600"></canvas> <!--Aqui se agrega el Elemeto CANVAS--
>
23.
24.     <script> //Inicio del JavaScript
25.     var canvas = document.getElementById("miCanvas");
26.     var ctx = canvas.getContext("2d");
27.
28.     var x=canvas.width/2; //Coordenadas inicio
horizontal
29.     var y=canvas.height -65; //Coordenadas inicio vertical
30.
31.     var dx=2; //Dirección del cuadrado respecto a x
32.     var dy=-2; //Dirección del cuadrado respecto a y
33.

```

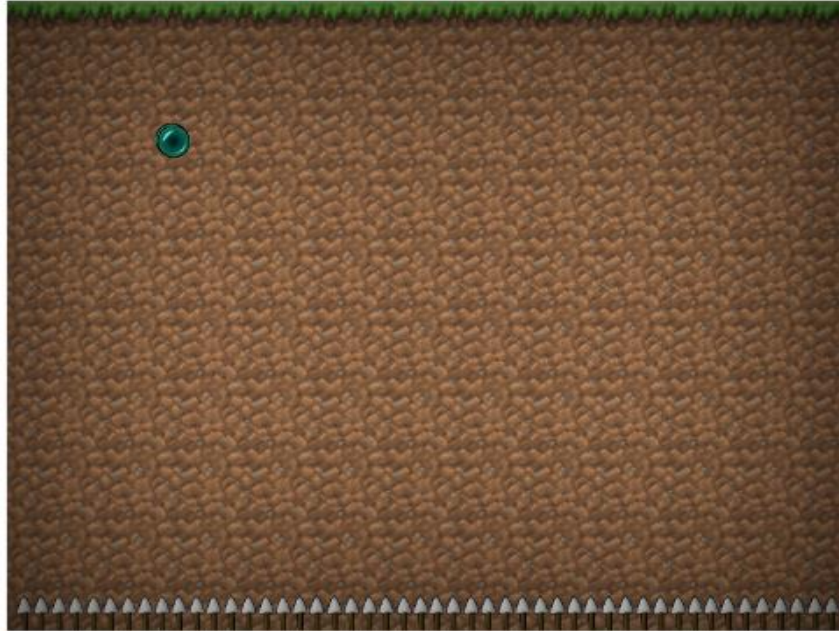
```

34.     function ProyectilVoid() { //Función de creación del
cuadrado
35.     var img = new Image();
36.     img.src = "img/PROYECTIL.png"; //Imagen utilizada
37.     ctx.drawImage(img, x, y); //Posición de aparición de la
imagen
38.     img.onload = function() {
39.     ctx.drawImage(img, x, y);
40.     }
41.     if(y + dy > 540 || y + dy < 0) { //Si la suma de y Y
dy es mayor que 540 o menor que 0 se le dara un nuevo valor a
dy
42.         dy = -dy; //Esto hará que dy cambia a un valor
negativo haciendo que cambia su rumbo
43.     }
44.     if(x + dx > 775 || x + dx < 0) { //lo mismo se aplica
con las coordenadas de x
45.         dx = -dx;
46.     }
47.     }
48.     function Dibujarvoid() { //Función de llamado a la
anterior función
49.     ctx.clearRect(0, 0, canvas.width, canvas.height);
//Función para limpiar el CANVAS
50.     ProyectilVoid(); //Llamado de función
51.     x = x + dx; //Nuevo valor para x
52.     y = y + dy; //Nuevo valor para y
53.     }
54.
55.     setInterval(Dibujarvoid, 10); //Llamado de la función
anterior cada diez milisegundos.
56.     </script>
57.

```

Al ejecutar este código se obtiene la siguiente interfaz visual:

JUEGO ROMPER TODOS LOS LADRILLOS



Gráfica 3. Rebote de la pelota en las paredes

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 número 3 consiste en crear la paleta en la cual la bola rebotara, la cual estará situada en la parte inferior del juego y sera 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 FlechaDerPVoid y FlechaIzqPVoid Luego de esto se crea una función la cual maneja el movimiento de la tecla presionada y otro de la tecla liberada

```

1. <!DOCTYPE html>
2. <html>
3. <head>
4.     <title>Juego 2D: JavaScript VOID</title>
5.     <style>
6.         * {
7.             padding: 0;
8.             margin: 0;
9.         }
10.        canvas {
11.            background-image: url(img/FONDO.png); /* Imagen
12.            de fondo para el CANVAS*/
13.            display: block;
14.            margin: 0 auto;
15.        }
16.    </style>
17. </head>
18. <body>
19.     <center>
20.         <p style="font-family:Comic Sans MS, cursive;font-size:
21.         40px;">JUEGO <span style="color:#FF0000">ROMPER TODOS</span> LOS
22.         LADRILLOS</p>
23.     </center>
24.
25.     <canvas id="miCanvas" width="800" height="600"></canvas> <!--
26.     Aqui se agrega el Elemento CANVAS-->
27.     <script> //Inicio del JavaScript
28.
29.         ///////////VARIABLES//////////
30.         var canvas = document.getElementById("miCanvas");
31.         var ctx = canvas.getContext("2d");
32.         var img = new Image();
33.         //
34.         var x=canvas.width/2;    //Coordenadas incio horizontal
35.         var y=canvas.height -65; //Coordenadas inicio vertical
36.         //
37.         var dx=2; //Dirección del cuadrado respecto a x
38.         var dy=-2; //Dirección del cuadrado respecto a y

```

```

35.  //
36.  var AlturaPaletaVoid = 30; //Se define la altura de la paleta
37.  var AnchuraPaletaVoid = 150; //se define el ancho de la
    paleta
38.  var PaletaPosXVoid = (canvas.width - AnchuraPaletaVoid) / 2;
    //Se define la posición de la paleta respecto a x
39.  //
40.  var FlechaDerPVoid = false; //Se define la variable en estado
    false
41.  var FlechaIzqPVoid = false; //Se define la variable en estado
    false
42.  //
43.  document.addEventListener("keydown", ManejarTeclaPresionada,
    false); //Se definen los observadores de eventos para la tecla
    presionada
44.  document.addEventListener("keyup", ManejarTeclaLiberada,
    false); // y para tecla liberada
45.  //
46.  //Funciones
47.  //
48.  function ManejarTeclaPresionada(e){ //Funcion que cambia el
    valor de la variable a true cuando se presiona la tecla
49.      if(e.keyCode == 39){ //el codigo de la tecla derecha es
        39
50.          FlechaDerPVoid = true;
51.      }
52.      else if(e.keyCode == 37){ //codigo tecla izquierda es 39
53.          FlechaIzqPVoid = true;
54.      }
55.  }
56.  //
57.  function ManejarTeclaLiberada(e){ //funcion que detecta
    cuando la tecla ya no es presionada
58.      if(e.keyCode == 39){
59.          FlechaDerPVoid = false;
60.      }
61.      else if(e.keyCode == 37){
62.          FlechaIzqPVoid = false;
63.      }
64.  }
65.  //
66.  function DibujarPaletaVoid(){ //Funcion que define la forma
    de la paleta
67.      img.src = "img/nether.png"; //ruta de la imagen utilizada
    como textura de la paleta
68.      ctx.drawImage(img, PaletaPosXVoid, canvas.height-
        AlturaPaletaVoid, AnchuraPaletaVoid, AlturaPaletaVoid);
        //Posicionamiento de la imagen encima del area de la paleta
69.      img.onload = function(){
70.          ctx.drawImage(img, PaletaPosXVoid, canvas.height-
            AlturaPaletaVoid, AnchuraPaletaVoid, AlturaPaletaVoid);
71.      }
72.  }
73.  //

```

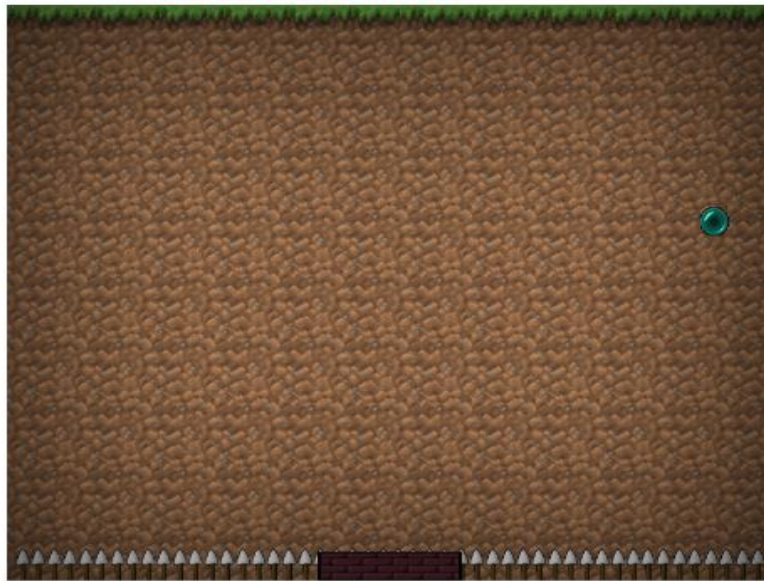
```

74.     function ProyectilVoid() { //Función de creación del cuadrado
75.         var img = new Image();
76.         img.src = "img/PROYECTIL.png"; //Imagen utilizada
77.         ctx.drawImage(img, x, y); //Posición de aparición de la
imagen
78.         img.onload = function() {
79.             ctx.drawImage(img, x, y);
80.         }
81.         if(y + dy > 555 || y + dy < 0) {
82.             dy = -dy;
83.         }
84.         if(x + dx > 750 || x + dx < 0) {
85.             dx = -dx;
86.         }
87.     }
88.     //
89.     function DibujarVoid() { //Función de llamado a la anterior
función
90.         ctx.clearRect(0, 0, canvas.width, canvas.height);
//Función para limpiar el CANVAS
91.         //
92.         ProyectilVoid(); //Llamado de función
93.         x = x + dx; //Nuevo valor para x
94.         y = y + dy; //Nuevo valor para y
95.         //
96.         DibujarPaletaVoid(); // se agrega el llamado de la
funcion
97.
98.
99.         if(FlechaDerPVoid && PaletaPosXVoid < canvas.width-
AnchuraPaletaVoid) { //condicional para que la paleta no se
salga del lienzo y que permita el movimiento de la paleta
100.             PaletaPosXVoid += 7;
101.         }
102.         else if(FlechaIzqPVoid && PaletaPosXVoid > 0) { // <-
-----
103.             PaletaPosXVoid -= 7;
104.         }
105.         x += dx;
106.         y += dy;
107.     }
108.     setInterval(DibujarVoid, 10); //Llamado de la función
anterior cada diez milisegundos.
109.     //Final del JavaScript
110. </script>

```

Al ejecutar este código se obtiene la siguiente interfaz visual:

JUEGO ROMPER TODOS LOS LADRILLOS



Gráfica 4. Paleta

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.

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 FASE4: 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 sería: $(y + dy > 521)$ el cual se utilizaría para cuando la bola toque la parte inferior del juego lo cual haría que el juego se pierda. Pero para estar seguros de que el juego se ha perdido analizaremos el siguiente código: $(x > \text{PaletaPosXVoid} \ \&\& \ x < \text{PaletaPosXVoid} + \text{AnchuraPaletaVoid})$ el cual hace que se analice la posición de la bola y en caso de que la bola toque la parte inferior hace que se detenga el ciclo de animación del juego y se pierda.

```

1. <!DOCTYPE html>
2. <html>
3. <head>
4.   <title>Juego 2D: JavaScript VOID</title>
5.   <style>
6.     * {
7.       padding: 0;
8.       margin: 0;
9.     }
10.    canvas {
11.      background-image: url(img/FONDO.png); /* Imagen
de fondo para el CANVAS*/
12.      display: block;
13.      margin: 0 auto;
14.    }
15.  </style>
16. </head>
17. <body>
18.   <center>
19.     <p style="font-family:Comic Sans MS, cursive;font-size:
40px;">JUEGO <span style="color:#FF0000">ROMPER TODOS</span> LOS
LADRILLOS</p>
20.   </center>
21.
22.   <canvas id="miCanvas" width="800" height="600"></canvas> <!--
Aquí se agrega el Elemento CANVAS-->
23.   <script> //Inicio del JavaScript
24.
25.   ///////////////VARIABLES/////////////////
26.   var canvas = document.getElementById("miCanvas");
27.   var ctx = canvas.getContext("2d");
28.   var img = new Image();
29.   //
30.   var x=canvas.width/2;    //Coordenadas inicio horizontal
31.   var y=canvas.height -78; //Coordenadas inicio vertical
32.   //

```

```

33.  var dx=2; //Dirección del cuadrado respecto a x
34.  var dy=-2; //Dirección del cuadrado respecto a y
35.  //
36.  var AlturaPaletaVoid = 35;
37.  var AnchuraPaletaVoid = 150;
38.  var PaletaPosXVoid = (canvas.width - AnchuraPaletaVoid) / 2;
39.  //
40.  var FlechaDerPVoid = false;
41.  var FlechaIzqPVoid = false;
42.  //
43.  document.addEventListener("keydown", ManejarTeclaPresionada,
false);
44.  document.addEventListener("keyup", ManejarTeclaLiberada,
false);
45.  //
46.  //Funciones
47.  //
48.  function ManejarTeclaPresionada(e) {
49.      if(e.keyCode == 39) {
50.          FlechaDerPVoid = true;
51.      }
52.      else if(e.keyCode == 37) {
53.          FlechaIzqPVoid = true;
54.      }
55.  }
56.  //
57.  function ManejarTeclaLiberada(e) {
58.      if(e.keyCode == 39) {
59.          FlechaDerPVoid = false;
60.      }
61.      else if(e.keyCode == 37) {
62.          FlechaIzqPVoid = false;
63.      }
64.  }
65.  //
66.  function DibujarPaletaVoid() {
67.      img.src = "img/Nether.png";
68.      ctx.drawImage(img, PaletaPosXVoid, canvas.height-
AlturaPaletaVoid, AnchuraPaletaVoid, AlturaPaletaVoid);
69.      img.onload = function() {
70.          ctx.drawImage(img, PaletaPosXVoid, canvas.height-
AlturaPaletaVoid, AnchuraPaletaVoid, AlturaPaletaVoid);
71.      }
72.  }
73.  //
74.  function ProyectilVoid() { //Función de creación del cuadrado
75.      var img = new Image();
76.      img.src = "img/PROYECTIL.png"; //Imagen utilizada
77.      ctx.drawImage(img, x, y); //Posición de aparición de la
imagen
78.      img.onload = function() {
79.          ctx.drawImage(img, x, y);
80.      }
81.  }

```

```

82.  //
83.  function DibujarVoid() { //Función de llamado a la anterior
    función
84.      ctx.clearRect(0, 0, canvas.width, canvas.height);
    //Función para limpiar el CANVAS
85.      //
86.      ProyectilVoid(); //Llamado de función
87.      x = x + dx; //Nuevo valor para x
88.      y = y + dy; //Nuevo valor para y
89.      //
90.      DibujarPaletaVoid();
91.      if(y + dy > 522 || y + dy < 0) {
92.          dy = -dy;
93.      }
94.      if(x + dx > 750 || x + dx < 0) {
95.          dx = -dx;
96.      }
97.      else if(y + dy > 521) { //se crea una condicion en la
    cual al pasar de cierta posición en la pantalla ejecuta otra
    condición
98.          if(x > PaletaPosXVoid-45 && x < PaletaPosXVoid +
    AnchuraPaletaVoid+45) { //Si al momento de estar en esa posición
    esta en el rango de la Paleta esta rebota
99.              dy = -dy;
100.          }
101.          else { //si no ocurre lo anterior y esta bajo la
    coordenada
102.              clearInterval(juego); //Se limpia el intervalo y
    no se dibujan más fotogramas
103.              alert("Perdiste :c"); //Y aparece el mensaje
    emergente
104.              document.location.reload();//Al aceptarlo se
    reinicia la pagina
105.          }
106.      }
107.
108.
109.      if(FlechaDerPVoid && PaletaPosXVoid < canvas.width-
    AnchuraPaletaVoid) {
110.          PaletaPosXVoid += 7;
111.      }
112.      else if(FlechaIzqPVoid && PaletaPosXVoid > 0) { // <-
    -----
113.          PaletaPosXVoid -= 7;
114.      }
115.      x += dx;
116.      y += dy;
117.
118.  }
119.  var juego = setInterval(DibujarVoid, 10);
120.  //Final del JavaScript
121.  </script>

```


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 “Perdiste :c” que significa que el juego se ha perdido y se ha acabado

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.

Analizaremos la siguiente función: `function DibujarLadrillosVoid()`, 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.

```

1. <!DOCTYPE html>
2. <html>
3.   <head>
4.     <title>Juego 2D: JavaScript VOID</title>
5.     <style>
6.       * {
7.         padding: 0;
8.         margin: 0;
9.       }
10.      canvas {
11.        background-image: url(img/FONDO.png); /* Imagen
de fondo para el CANVAS*/
12.        display: block;
13.        margin: 0 auto;
14.      }
15.    </style>
16.  </head>
17.  <body>
18.    <center>
19.      <p style="font-family:Comic Sans MS,
cursive;font-size: 40px;">JUEGO <span
style="color:#FF0000">ROMPER TODOS</span> LOS LADRILLOS</p>
20.    </center>
21.  </body>

```

```

22.         <canvas id="miCanvas" width="800"
height="600"></canvas> <!--Aquí se agrega el Elemento CANVAS-->
23.         <script> //Inicio del JavaScript
24.
25.             ///////////////VARIABLES/////////////////
26.         var canvas = document.getElementById("miCanvas");//Elemento
Canvas
27.         var ctx = canvas.getContext("2d");
28.         var img = new Image();
29.         //
30.         var x=canvas.width/2;    //Coordenadas inicio horizontal
31.         var y=canvas.height -78; //Coordenadas inicio vertical
32.         //
33.         var dx=2; //Dirección del cuadrado respecto a x
34.         var dy=-2; //Dirección del cuadrado respecto a y
35.         //
36.         var AlturaPaletaVoid = 35;    //Altura paleta
37.         var AnchuraPaletaVoid = 150;   //Anchura paleta
38.         var PaletaPosXVoid = (canvas.width - AnchuraPaletaVoid) / 2;
//Poscion paleta
39.         //
40.         var FlechaDerPVoid = false; //Estado de tecla presionada
41.         var FlechaIzqPVoid = false; //Estado de tecla preionada
42.         //
43.         var nroFilasLadrillosVoid = 5;    // Número de Filas
44.         var nroColumnasLadrillosVoid = 2;    // Número de columnas
45.         var anchoLadrilloVoid = 140;    // Ancho de los
ladrillos
46.         var alturaLadrilloVoid = 40;    // Altura de los
ladrillos
47.         var rellenoLadrilloVoid = 10;    // Relleno de los
ladrillos
48.         var vacioSuperiorLadrilloVoid = 30;    // Vavio Superior de
los ladrillos
49.         var vacioIzquierdoLadrilloVoid = 30;    // vacio de la
izquierda a los ladrillos
50.         //
51.         var ladrillosVoid = []; //Conjunto de ladrillos vacios
52.         //
53.         for(var columnaVoid=0; columnaVoid<nroColumnasLadrillosVoid;
columnaVoid++){
54.             //Define la primera columna. Lista vertical
55.             ladrillosVoid[columnaVoid] = [];
56.             for(var filaVoid=0; filaVoid<nroFilasLadrillosVoid;
filaVoid++){
57.                 ladrillosVoid[columnaVoid][filaVoid] = {x: 0, y: 0};
58.             }
59.         }
60.         document.addEventListener("keydown", ManejarTeclaPresionada,
false); //Observador
61.         document.addEventListener("keyup", ManejarTeclaLiberada,
false); //Observador
62.         //
63.         //Funciones

```

```

64.  //
65.  function ManejarTeclaPresionada(e){ //Funcion encargada de
    cambiar el estado de la tecla cuando se preiosna(true)
66.      if(e.keyCode == 39){
67.          FlechaDerPVoid = true;
68.      }
69.      else if(e.keyCode == 37){
70.          FlechaIzqPVoid = true;
71.      }
72.  }
73.  //
74.  function ManejarTeclaLiberada(e){ //Funcion encargada de
    cambiar el estado de la tecla cuando se preiosna(false)
75.      if(e.keyCode == 39){
76.          FlechaDerPVoid = false;
77.      }
78.      else if(e.keyCode == 37){
79.          FlechaIzqPVoid = false;
80.      }
81.  }
82.  //
83.  function DibujarPaletaVoid(){ //Funcion que dibuja la paleta
84.      img.src = "img/Nether.png";
85.      ctx.drawImage(img,PaletaPosXVoid, canvas.height-
        AlturaPaletaVoid, AnchuraPaletaVoid, AlturaPaletaVoid);
86.      img.onload = function(){
87.          ctx.drawImage(img,PaletaPosXVoid, canvas.height-
        AlturaPaletaVoid, AnchuraPaletaVoid, AlturaPaletaVoid);
88.      }
89.  }
90.  //
91.  function ProyectilVoid() { //Función de creación del cuadrado
92.      var img = new Image();
93.      img.src = "img/PROYECTIL.png"; //Imagen utilizada
94.      ctx.drawImage(img, x, y); //Posición de aparición de la
    imagen
95.      img.onload = function(){
96.          ctx.drawImage(img, x, y);
97.      }
98.  }
99.  //
100. function DibujarLadrillosVoid(){
101.     for(var columnaVoid=0;
        columnaVoid<nroColumnasLadrillosVoid; columnaVoid++){
102.         for(var filaVoid =0; filaVoid<nroFilasLadrillosVoid;
            filaVoid++){
103.             var brickX =
                (filaVoid*(anchoLadrilloVoid+rellenoLadrilloVoid))+vacioIzquierd
                oLadrilloVoid;
104.             var brickY =
                (columnaVoid*(alturaLadrilloVoid+rellenoLadrilloVoid))+vacioSupe
                riorLadrilloVoid;
105.             ladrillosVoid[columnaVoid][filaVoid].x = brickX
106.             ladrillosVoid[columnaVoid][filaVoid].y = brickY

```

```

107.     var img = new Image();
108.     img.src = "img/Piedra.png";
109.     ctx.drawImage(img,brickX, brickY, anchoLadrilloVoid,
    alturaLadrilloVoid);
110.     ctx.onload= function(){
111.         ctx.drawImage(img,brickX, brickY, anchoLadrilloVoid,
    alturaLadrilloVoid);
112.     }
113. }
114. }
115. }
116. //
117. function Dibujarvoid() { //Función de llamado a la anterior
    función
118.     ctx.clearRect(0, 0, canvas.width, canvas.height);
    //Función para limpiar el CANVAS
119.     //
120.     DibujarLadrillosVoid();
121.     //
122.     ProyectilVoid(); //Llamado de función
123.     x = x + dx; //Nuevo valor para x
124.     y = y + dy; //Nuevo valor para y
125.     //
126.     DibujarPaletaVoid();
127.     if(y + dy > 522 || y + dy < 0) { //Condicion para que el
    cuadrado rebote en las coordenada y
128.         dy = -dy;
129.     }
130.     if(x + dx > 750 || x + dx < 0) { //Condicion para que el
    cuadrado rebote en las coordenada y
131.         dx = -dx;
132.     }
133.     else if(y + dy > 521) { //Condicion para que el
    cuadrado
134.         if(x > PaletaPosXVoid-45 && x < PaletaPosXVoid +
    AnchuraPaletaVoid+45) { //Se define una condición donde si al
    momento del cuadro pasar de cierta coordenada y y al estar en el
    mismo espacio de la paleta esat rebotara
135.             dy = -dy;
136.         }
137.         else { //De lo contrario se limpiara el intervalo de
    la variable juego que es la que llama la funcion de Dibujar todo
    cada 10 milisegundos
138.             clearInterval(juego);
139.             alert("Perdiste :c");
140.             document.location.reload();
141.         }
142.     }
143.
144.
145.     if(FlechaDerPVoid && PaletaPosXVoid < canvas.width-
    AnchuraPaletaVoid) {
146.         PaletaPosXVoid += 7;
147.     }

```

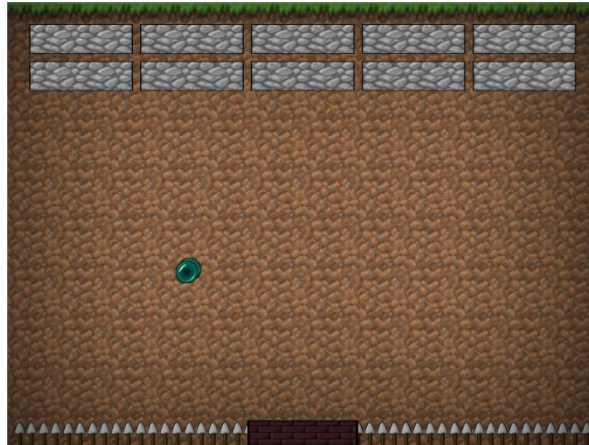
```

148.         else if(FlechaIzqPVoid && PaletaPosXVoid > 0) { // <-
-----
149.             PaletaPosXVoid -= 7;
150.         }
151.         x += dx;
152.         y += dy;
153.
154.     }
155.     var juego = setInterval(Dibujarvoid, 10);
156.     //Final del JavaScript
157. </script>

```

Al ejecutar este código se obtiene la siguiente interfaz visual:

JUEGO ROMPER TODOS LOS LADRILLOS



En la figura 6 podemos observar la creación de la pared de ladrillos dentro del campo del 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.

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 DeteccionColisionVoid()`, 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.

También se crean las siguientes variables: la primera es `clearInterval(juego)`; la cual hace que se detenga el ciclo del juego, otra es `alert("Perdiste :c")`; 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

```

1. <!DOCTYPE html>
2. <html>
3.   <head>
4.     <title>Juego 2D: JavaScript VOID</title>
5.     <style>
6.       * {
7.         padding: 0;
8.         margin: 0;
9.       }
10.      canvas {
11.        background-image: url(img/FONDO.png); /* Imagen
de fondo para el CANVAS*/
12.        display: block;
13.        margin: 0 auto;
14.      }
15.    </style>
16.  </head>
17.  <body>
18.    <center>
19.      <p style="font-family:Comic Sans MS,
cursive;font-size: 40px;">JUEGO <span
style="color:#FF0000">ROMPER TODOS</span> LOS LADRILLOS</p>
20.    </center>
21.
22.    <canvas id="miCanvas" width="800"
height="600"></canvas> <!--Aqui se agrega el Elemeto CANVAS-->
23.    <script> //Inicio del JavaScript
24.
25.      ///////////////////////////////////

```

```

26.   var canvas = document.getElementById("miCanvas");//Elemento
      Canvas
27.   var ctx = canvas.getContext("2d");
28.   var img = new Image();
29.   //
30.   var x=canvas.width/2;    //Coordenadas inicio horizontal
31.   var y=canvas.height -78; //Coordenadas inicio vertical
32.   //
33.   var dx=2; //Dirección del cuadrado respecto a x
34.   var dy=-2; //Dirección del cuadrado respecto a y
35.   //
36.   var AlturaPaletaVoid = 35;    //Altura paleta
37.   var AnchuraPaletaVoid = 150;   //Anchura paleta
38.   var PaletaPosXVoid = (canvas.width - AnchuraPaletaVoid) / 2;
      //Poscion paleta
39.   //
40.   var FlechaDerPVoid = false; //Estado de tecla presionada
41.   var FlechaIzqPVoid = false; //Estado de tecla preionada
42.   //
43.   var nroFilasLadrillosVoid = 5;    // Número de Filas
44.   var nroColumnasLadrillosVoid = 2;  // Número de columnas
45.   var anchoLadrilloVoid = 140;      // Ancho de los
      ladrillos
46.   var alturaLadrilloVoid = 40;      // Altura de los
      ladrillos
47.   var rellenoLadrilloVoid = 10;     // Relleno de los
      ladrillos
48.   var vacioSuperiorLadrilloVoid = 30; // Vavio Superior de
      los ladrillos
49.   var vacioIzquierdoLadrilloVoid = 30; // vacio de la
      izquierda a los ladrillos
50.   //
51.   var nroFilasLadrillosVoid2 = 5;    // Número de Filas
52.   var nroColumnasLadrillosVoid2 = 2;  // Número de columnas
53.   var anchoLadrilloVoid2 = 140;      // Ancho de los
      ladrillos
54.   var alturaLadrilloVoid2 = 40;      // Altura de los
      ladrillos
55.   var rellenoLadrilloVoid2 = 10;     // Relleno de los
      ladrillos
56.   var vacioSuperiorLadrilloVoid2 = 140; // Vavio Superior de
      los ladrillos
57.   var vacioIzquierdoLadrilloVoid2 = 30; // vacio de la
      izquierda a los ladrillos
58.   //
59.   var ladrillosVoid = []; //Conjunto de ladrillos vacios
60.   //
61.   for(var c=0; c<nroColumnasLadrillosVoid; c++){
62.       //Define la primera columna. Lista vertical
63.       ladrillosVoid[c] = [];
64.       for(var f=0; f<nroFilasLadrillosVoid2; f++){
65.           ladrillosVoid[c][f] = {x: 0, y: 0, estado: 1};
66.       }
67.   }

```



```

68.  //
69.  var ladrillosVoid2 = []; //Conjunto de ladrillos vacios
70.  //
71.  for(var c2=0; c2<nroColumnasLadrillosVoid2; c2++){
72.    //Define la primera columna. Lista vertical
73.    ladrillosVoid2[c2] = [];
74.    for(var f2=0; f2<nroFilasLadrillosVoid2; f2++){
75.      ladrillosVoid2[c2][f2] = {x: 0, y: 0, estado2: 1};
76.    }
77.  }
78.  //
79.  document.addEventListener("keydown", ManejarTeclaPresionada,
false); //Observador
80.  document.addEventListener("keyup", ManejarTeclaLiberada,
false); //Observador
81.  //
82.  //Funciones
83.  //
84.  function ManejarTeclaPresionada(e){ //Funcion encargada de
cambiar el estado de la tecla cuando se preiosna(true)
85.    if(e.keyCode == 39){
86.      FlechaDerPVoid = true;
87.    }
88.    else if(e.keyCode == 37){
89.      FlechaIzqPVoid = true;
90.    }
91.  }
92.  //
93.  function ManejarTeclaLiberada(e){ //Funcion encargada de
cambiar el estado de la tecla cuando se preiosna(false)
94.    if(e.keyCode == 39){
95.      FlechaDerPVoid = false;
96.    }
97.    else if(e.keyCode == 37){
98.      FlechaIzqPVoid = false;
99.    }
100.  }
101.  //
102.  function DeteccionColisionVoid(){
103.    for(var c=0; c<nroColumnasLadrillosVoid; c++){
104.      for(var f=0; f<nroFilasLadrillosVoid; f++){
105.        var b = ladrillosVoid[c][f];
106.        if(b.estado == 1){
107.          if(x > b.x && x < b.x+anchoLadrilloVoid && y > b.y &&
y < b.y+alturaLadrilloVoid){
108.            dy = -dy;
109.            b.estado = 0;
110.          }
111.        }
112.      }
113.    }
114.  }
115.  //
116.  function DeteccionColisionVoid2(){

```

```

117.     for(var c2=0; c2<nroColumnasLadrillosVoid2; c2++){
118.         for(var f2=0; f2<nroFilasLadrillosVoid2; f2++){
119.             var b2 = ladrillosVoid2[c2][f2];
120.             if(b2.estado2 == 1){
121.                 if(x > b2.x && x < b2.x+anchoLadrilloVoid2 && y >
b2.y && y < b2.y+alturaLadrilloVoid2){
122.                     dy = -dy;
123.                     b2.estado2 = 0;
124.                 }
125.             }
126.         }
127.     }
128. }
129. //
130. function DibujarPaletaVoid(){ //Funcion que dibuja la paleta
131.     img.src = "img/Nether.png";
132.     ctx.drawImage(img,PaletaPosXVoid, canvas.height-
AlturaPaletaVoid, AnchuraPaletaVoid, AlturaPaletaVoid);
133.     img.onload = function(){
134.         ctx.drawImage(img,PaletaPosXVoid, canvas.height-
AlturaPaletaVoid, AnchuraPaletaVoid, AlturaPaletaVoid);
135.     }
136. }
137. //
138. function ProyectilVoid() { //Función de creación del cuadrado
139.     var img = new Image();
140.     img.src = "img/PROYECTIL.png"; //Imagen utilizada
141.     ctx.drawImage(img, x, y); //Posición de aparición de la
imagen
142.     img.onload = function(){
143.         ctx.drawImage(img, x, y);
144.     }
145. }
146. //
147.
148. function DibujarLadrillosVoid(){
149.     for(var c=0; c<nroColumnasLadrillosVoid; c++){
150.         for(var f =0; f<nroFilasLadrillosVoid; f++){
151.             if(ladrillosVoid[c][f].estado == 1){
152.                 var brickX =
(f*(anchoLadrilloVoid+rellenoLadrilloVoid))+vacioIzquierdoLadril
loVoid;
153.                 var brickY =
(c*(alturaLadrilloVoid+rellenoLadrilloVoid))+vacioSuperiorLadril
loVoid;
154.                 ladrillosVoid[c][f].x = brickX;
155.                 ladrillosVoid[c][f].y = brickY;
156.                 var img = new Image();
157.                 img.src = "img/Piedra.png";
158.                 ctx.drawImage(img,brickX, brickY, anchoLadrilloVoid,
alturaLadrilloVoid);
159.                 ctx.onload= function(){
160.                     ctx.drawImage(img,brickX, brickY, anchoLadrilloVoid,
alturaLadrilloVoid);

```

```

161.     }
162.   }
163. }
164. }
165. }
166. //
167. function DibujarLadrillosVoid2 () {
168.   for(var c2=0; c2<nroColumnasLadrillosVoid2; c2++){
169.     for(var f2 =0; f2<nroFilasLadrillosVoid2; f2++){
170.       if(ladrillosVoid2[c2][f2].estado2 == 1){
171.         var brickX2 =
172.           (f2*(anchoLadrilloVoid2+rellenoLadrilloVoid2))+vacioIzquierdoLad
173.           rilloVoid2;
174.         var brickY2 =
175.           (c2*(alturaLadrilloVoid2+rellenoLadrilloVoid2))+vacioSuperiorLad
176.           rilloVoid2;
177.         ladrillosVoid2[c2][f2].x = brickX2;
178.         ladrillosVoid2[c2][f2].y = brickY2;
179.         var img = new Image();
180.         img.src = "img/Madera.png";
181.         ctx.drawImage(img,brickX2, brickY2,
182.           anchoLadrilloVoid2, alturaLadrilloVoid2);
183.         ctx.onload= function() {
184.           ctx.drawImage(img,brickX2, brickY2,
185.             anchoLadrilloVoid2, alturaLadrilloVoid2);
186.         }
187.       }
188.     }
189.   }
190. }
191. //
192. function Dibujarvoid() { //Función de llamado a la anterior
193.   función
194.   ctx.clearRect(0, 0, canvas.width, canvas.height);
195.   //Función para limpiar el CANVAS
196.   //
197.   DibujarLadrillosVoid();
198.   //
199.   DibujarLadrillosVoid2();
200.   //
201.   ProyectilVoid(); //Llamado de función
202.   x = x + dx; //Nuevo valor para x
203.   y = y + dy; //Nuevo valor para y
204.   //
205.   DibujarPaletaVoid();
206.   DeteccionColisionVoid();
207.   DeteccionColisionVoid2();
208.   if(y + dy > 522 || y + dy < 0) { //Condicion para que el
209.     cuadrado rebote en las coordenada y
210.     dy = -dy;
211.   }
212.   if(x + dx > 750 || x + dx < 0) { //Condicion para que el
213.     cuadrado rebote en las coordenada y
214.     dx = -dx;

```

```

205.     }
206.     else if(y + dy > 521) { //Condicion para que el
    cuadrado
207.         if(x > PaletaPosXVoid-45 && x < PaletaPosXVoid +
    AnchuraPaletaVoid+45) { //Se define una condición donde si al
    momento del cuadro pasar de cierta coordenada y y al estar en el
    mismo espacio de la paleta esat rebotara
208.             dy = -dy;
209.         }
210.         else { //De lo contrario se limpiara el intervalo de
    la variable juego que es la que llama la funcion de Dibujar todo
    cada 10 milisegundos
211.             clearInterval(juego);
212.             alert("Perdiste :c");
213.             document.location.reload();
214.         }
215.     }
216.
217.
218.     if(FlechaDerPVoid && PaletaPosXVoid < canvas.width-
    AnchuraPaletaVoid) {
219.         PaletaPosXVoid += 7;
220.     }
221.     else if(FlechaIzqPVoid && PaletaPosXVoid > 0) { // <-
    -----
222.         PaletaPosXVoid -= 7;
223.     }
224.     x += dx;
225.     y += dy;
226.
227. }
228. var juego = setInterval(Dibujarvoid, 10);
229. //Final del JavaScript
230. </script>

```

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, 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.

```

1. <!DOCTYPE html>
2. <html>
3.   <head>
4.     <title>Juego 2D: JavaScript VOID</title>
5.     <style>
6.       * {
7.         padding: 0;
8.         margin: 0;
9.       }
10.      canvas {
11.        background-image: url(img/FONDO.png); /* Imagen
de fondo para el CANVAS*/
12.        display: block;
13.        margin: 0 auto;
14.      }
15.    </style>
16.  </head>
17.  <body>
18.    <center>
19.      <p style="font-family:Comic Sans MS,
cursive;font-size: 40px;">JUEGO <span
style="color:#FF0000">ROMPER TODOS</span> LOS LADRILLOS</p>
20.    </center>
21.
22.    <canvas id="miCanvas" width="800"
height="600"></canvas> <!--Aqui se agrega el Elemeto CANVAS-->
23.    <script> //Inicio del JavaScript
24.
25.      ////////////VARIABLES//////////
26.      var canvas = document.getElementById("miCanvas");//Elemento
Canvas
27.      var ctx = canvas.getContext("2d");
28.      var img = new Image();
29.      //
30.      var x=canvas.width/2; //Coordenadas incio horizontal
31.      var y=canvas.height -78; //Coordenadas inicio vertical
32.      //
33.      var dx=2; //Dirección del cuadrado respecto a x
34.      var dy=-2; //Dirección del cuadrado respecto a y
35.      //
36.      var AlturaPaletaVoid = 35; //Altura paleta
37.      var AnchuraPaletaVoid = 150; //Anchura paleta
38.      var PaletaPosXVoid = (canvas.width - AnchuraPaletaVoid) / 2;
//Poscion paleta
39.      //
40.      var FlechaDerPVoid = false; //Estado de tecla presionada
41.      var FlechaIzqPVoid = false; //Estado de tecla preionada
42.      //
43.      var puntajeVoid = 0; //Estado inicial del puntaje
44.      //
45.      var nroFilasLadrillosVoid = 5; // Número de Filas
46.      var nroColumnasLadrillosVoid = 2; // Número de columnas
47.      var anchoLadrilloVoid = 140; // Ancho de los
ladrillos

```

```

48.   var alturaLadrilloVoid = 40;           // Altura de los
      ladrillos
49.   var rellenoLadrilloVoid = 10;          // Relleno de los
      ladrillos
50.   var vacioSuperiorLadrilloVoid = 30;    // Vavio Superior de
      los ladrillos
51.   var vacioIzquierdoLadrilloVoid = 30;   // vacio de la
      izquierda a los ladrillos
52.   //
53.   var nroFilasLadrillosVoid2 = 5;        // Número de Filas
54.   var nroColumnasLadrillosVoid2 = 2;     // Número de columnas
55.   var anchoLadrilloVoid2 = 140;          // Ancho de los
      ladrillos
56.   var alturaLadrilloVoid2 = 40;          // Altura de los
      ladrillos
57.   var rellenoLadrilloVoid2 = 10;          // Relleno de los
      ladrillos
58.   var vacioSuperiorLadrilloVoid2 = 140;   // Vavio Superior de
      los ladrillos
59.   var vacioIzquierdoLadrilloVoid2 = 30;   // vacio de la
      izquierda a los ladrillos
60.   //
61.   var ladrillosVoid = []; //Conjunto de ladrillos vacios
62.   //
63.   for(var c=0; c<nroColumnasLadrillosVoid; c++){
64.       //Define la primera columna. Lista vertical
65.       ladrillosVoid[c] = [];
66.       for(var f=0; f<nroFilasLadrillosVoid2; f++){
67.           ladrillosVoid[c][f] = {x: 0, y: 0, estado: 1};
68.       }
69.   }
70.   //
71.   var ladrillosVoid2 = []; //Conjunto de ladrillos vacios
72.   //
73.   for(var c2=0; c2<nroColumnasLadrillosVoid2; c2++){
74.       //Define la primera columna. Lista vertical
75.       ladrillosVoid2[c2] = [];
76.       for(var f2=0; f2<nroFilasLadrillosVoid2; f2++){
77.           ladrillosVoid2[c2][f2] = {x: 0, y: 0, estado2: 1};
78.       }
79.   }
80.   //
81.   document.addEventListener("keydown", ManejarTeclaPresionada,
      false); //Observador
82.   document.addEventListener("keyup", ManejarTeclaLiberada,
      false); //Observador
83.   //
84.   //Funciones
85.   //
86.   function ManejarTeclaPresionada(e) { //Funcion encargada de
      cambiar el estado de la tecla cuando se preiosna(true)
87.       if(e.keyCode == 39) {
88.           FlechaDerPVoid = true;
89.       }

```

```

90.     else if(e.keyCode == 37){
91.         FlechaIzqPVoid = true;
92.     }
93. }
94. //
95. function ManejarTeclaLiberada(e){ //Funcion encargada de
    cambiar el estado de la tecla cuando se preiosna(false)
96.     if(e.keyCode == 39){
97.         FlechaDerPVoid = false;
98.     }
99.     else if(e.keyCode == 37){
100.         FlechaIzqPVoid = false;
101.     }
102. }
103. //
104. function DeteccionColisionVoid(){
105.     for(var c=0; c<nroColumnasLadrillosVoid; c++){
106.         for(var f=0; f<nroFilasLadrillosVoid; f++){
107.             var b = ladrillosVoid[c][f];
108.             if(b.estado == 1){
109.                 if(x > b.x && x < b.x+anchoLadrilloVoid && y > b.y &&
    y < b.y+alturaLadrilloVoid){
110.                     dy = -dy;
111.                     b.estado = 0;
112.                     puntajeVoid++;
113.                     if(puntajeVoid ==
    nroFilasLadrillosVoid*nroColumnasLadrillosVoid+nroFilasLadrillos
    Void2*nroColumnasLadrillosVoid2){
114.                         alert("GANASTE!!! FELICITACIONES :3!!!");
115.                         document.location.reload();
116.                     }
117.                 }
118.             }
119.         }
120.     }
121. }
122. //
123. function DeteccionColisionVoid2(){
124.     for(var c2=0; c2<nroColumnasLadrillosVoid2; c2++){
125.         for(var f2=0; f2<nroFilasLadrillosVoid2; f2++){
126.             var b2 = ladrillosVoid2[c2][f2];
127.             if(b2.estado2 == 1){
128.                 if(x > b2.x && x < b2.x+anchoLadrilloVoid2 && y >
    b2.y && y < b2.y+alturaLadrilloVoid2){
129.                     dy = -dy;
130.                     b2.estado2 = 0;
131.                     puntajeVoid++;
132.                     if(puntajeVoid ==
    nroFilasLadrillosVoid*nroColumnasLadrillosVoid+nroFilasLadrillos
    Void2*nroColumnasLadrillosVoid2){
133.                         alert("GANASTE!!! FELICITACIONES :3!!!");
134.                         document.location.reload();
135.                     }
136.                 }

```



```

137.     }
138.   }
139. }
140. }
141. //
142. function DibujarPaletaVoid() { //Funcion que dibuja la paleta
143.   img.src = "img/Nether.png";
144.   ctx.drawImage(img,PaletaPosXVoid, canvas.height-
    AlturaPaletaVoid, AnchuraPaletaVoid, AlturaPaletaVoid);
145.   img.onload = function() {
146.     ctx.drawImage(img,PaletaPosXVoid, canvas.height-
    AlturaPaletaVoid, AnchuraPaletaVoid, AlturaPaletaVoid);
147.   }
148. }
149. //
150. function DibujarpuntajeVoid() {
151.   ctx.font="16px Comic Sans MS";
152.   ctx.fillStyle = "#ff0000";
153.   ctx.fillText("Puntaje:"+puntajeVoid,8,20);
154. }
155. //
156. function ProyectilVoid() { //Función de creación del cuadrado
157.   var img = new Image();
158.   img.src = "img/PROYECTIL.png"; //Imagen utilizada
159.   ctx.drawImage(img, x, y); //Posición de aparición de la
    imagen
160.   img.onload = function() {
161.     ctx.drawImage(img, x, y);
162.   }
163. }
164. //
165.
166. function DibujarLadrillosVoid() {
167.   for(var c=0; c<nroColumnasLadrillosVoid; c++){
168.     for(var f=0; f<nroFilasLadrillosVoid; f++){
169.       if(ladrillosVoid[c][f].estado == 1){
170.         var brickX =
    (f*(anchoLadrilloVoid+rellenoLadrilloVoid))+vacioIzquierdoLadril
    loVoid;
171.         var brickY =
    (c*(alturaLadrilloVoid+rellenoLadrilloVoid))+vacioSuperiorLadril
    loVoid;
172.         ladrillosVoid[c][f].x = brickX;
173.         ladrillosVoid[c][f].y = brickY;
174.         var img = new Image();
175.         img.src = "img/Piedra.png";
176.         ctx.drawImage(img,brickX, brickY, anchoLadrilloVoid,
    alturaLadrilloVoid);
177.         ctx.onload= function() {
178.           ctx.drawImage(img,brickX, brickY, anchoLadrilloVoid,
    alturaLadrilloVoid);
179.         }
180.       }
181.     }

```

```

182.     }
183. }
184. //
185. function DibujarLadrillosVoid2() {
186.     for(var c2=0; c2<nroColumnasLadrillosVoid2; c2++){
187.         for(var f2 =0; f2<nroFilasLadrillosVoid2; f2++){
188.             if(ladrillosVoid2[c2][f2].estado2 == 1){
189.                 var brickX2 =
                    (f2*(anchoLadrilloVoid2+rellenoLadrilloVoid2))+vacioIzquierdoLad
                    rilloVoid2;
190.                 var brickY2 =
                    (c2*(alturaLadrilloVoid2+rellenoLadrilloVoid2))+vacioSuperiorLad
                    rilloVoid2;
191.                 ladrillosVoid2[c2][f2].x = brickX2;
192.                 ladrillosVoid2[c2][f2].y = brickY2;
193.                 var img = new Image();
194.                 img.src = "img/Madera.png";
195.                 ctx.drawImage(img,brickX2, brickY2,
                    anchoLadrilloVoid2, alturaLadrilloVoid2);
196.                 ctx.onload= function(){
197.                     ctx.drawImage(img,brickX2, brickY2,
                    anchoLadrilloVoid2, alturaLadrilloVoid2);
198.                 }
199.             }
200.         }
201.     }
202. }
203. //
204. function Dibujarvoid() { //Función de llamado a la anterior
    función
205.     ctx.clearRect(0, 0, canvas.width, canvas.height);
    //Función para limpiar el CANVAS
206.     //
207.     DibujarLadrillosVoid();
208.     //
209.     DibujarLadrillosVoid2();
210.     //
211.     ProyectilVoid(); //Llamado de función
212.     x = x + dx; //Nuevo valor para x
213.     y = y + dy; //Nuevo valor para y
214.     //
215.     DibujarPaletaVoid();
216.     DeteccionColisionVoid();
217.     DeteccionColisionVoid2();
218.     DibujarpuntajeVoid();
219.     if(y + dy > 522 || y + dy < 0) { //Condicion para que el
        cuadrado rebote en las coordenada y
220.         dy = -dy;
221.     }
222.     if(x + dx > 750 || x + dx < 0) { //Condicion para que el
        cuadrado rebote en las coordenada y
223.         dx = -dx;
224.     }

```

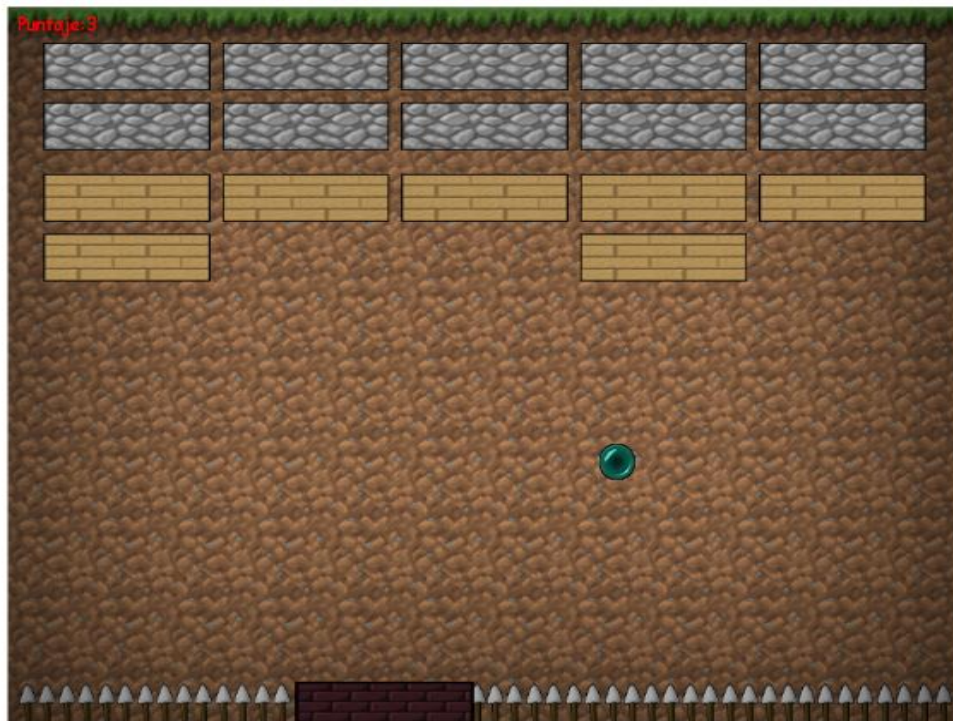
```

225.     else if(y + dy > 521) { //Condicion para que el
        cuadrado
226.         if(x > PaletaPosXVoid-45 && x < PaletaPosXVoid +
            AnchuraPaletaVoid-20) { //Se define una condición donde si al
            momento del cuadro pasar de cierta coordenada y y al estar en el
            mismo espacio de la paleta esat rebotara
227.             dy = -dy;
228.         }
229.         else { //De lo contrario se limpiara el intervalo de
            la variable juego que es la que llama la funcion de Dibujar todo
            cada 10 milisegundos
230.             clearInterval(juego);
231.             alert("Perdiste :c");
232.             document.location.reload();
233.         }
234.     }
235.
236.
237.     if(FlechaDerPVoid && PaletaPosXVoid < canvas.width-
        AnchuraPaletaVoid) {
238.         PaletaPosXVoid += 7;
239.     }
240.     else if(FlechaIzqPVoid && PaletaPosXVoid > 0) { // <-
        -----
241.         PaletaPosXVoid -= 7;
242.     }
243.     x += dx;
244.     y += dy;
245.
246. }
247. var juego = setInterval(Dibujarvoid, 10);
248. //Final del JavaScript
249. </script>

```

Al ejecutar este código se obtiene la siguiente interfaz visual:

JUEGO ROMPER TODOS LOS LADRILLOS



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: VIDAS

Al igual que en la anterior fase se creara una condición con las cuales se definirá un número de vidas y lo que pasa cuando estas se acaban.

```

1. <!DOCTYPE html>
2. <html>
3.   <head>
4.     <title>Juego 2D: JavaScript VOID</title>
5.     <style>
6.       * {
7.         padding: 0;
8.         margin: 0;
9.       }
10.      canvas {
11.        background-image: url(img/FONDO.png); /* Imagen
de fondo para el CANVAS*/
12.        display: block;
13.        margin: 0 auto;
14.      }
15.    </style>
16.  </head>
17.  <body>
18.    <center>
19.      <p style="font-family:Comic Sans MS,
cursive;font-size: 40px;">JUEGO <span
style="color:#FF0000">ROMPER TODOS</span> LOS LADRILLOS</p>
20.    </center>
21.
22.    <canvas id="miCanvas" width="800"
height="600"></canvas> <!--Aqui se agrega el Elemeto CANVAS-->
23.    <script> //Inicio del JavaScript
24.
25.      //////////VARIABLES//////////
26.      var canvas = document.getElementById("miCanvas");//Elemento
Canvas
27.      var ctx = canvas.getContext("2d");
28.      var img = new Image();
29.      //
30.      var x=canvas.width/2; //Coordenadas inicio horizontal
31.      var y=canvas.height -60; //Coordenadas inicio vertical
32.      //
33.      var dx=2; //Dirección del cuadrado respecto a x
34.      var dy=-2; //Dirección del cuadrado respecto a y
35.      //
36.      var AlturaPaletaVoid = 35; //Altura paleta
37.      var AnchuraPaletaVoid = 150; //Anchura paleta
38.      var PaletaPosXVoid = (canvas.width - AnchuraPaletaVoid) / 2;
//Poscion paleta
39.      //
40.      var FlechaDerPVoid = false; //Estado de tecla presionada

```

```

41.  var FlechaIzqPVoid = false; //Estado de tecla preionada
42.  //
43.  var puntajeVoid = 0; //Estado inicial del puntaje
44.  //
45.  var vidasVoid = 3; //Estado inicial de las vidas
46.  //
47.  var nroFilasLadrillosVoid = 5;           // Número de Filas
48.  var nroColumnasLadrillosVoid = 2;        // Número de columnas
49.  var anchoLadrilloVoid = 120;             // Ancho de los
    ladrillos
50.  var alturaLadrilloVoid = 30;             // Altura de los
    ladrillos
51.  var rellenoLadrilloVoid = 35;            // Relleno de los
    ladrillos
52.  var vacioSuperiorLadrilloVoid = 40;      // Vavio Superior de
    los ladrillos
53.  var vacioIzquierdoLadrilloVoid = 30;     // vacio de la
    izquierda a los ladrillos
54.  //
55.  var nroFilasLadrillosVoid2 = 5;          // Número de Filas
56.  var nroColumnasLadrillosVoid2 = 2;       // Número de columnas
57.  var anchoLadrilloVoid2 = 140;            // Ancho de los
    ladrillos
58.  var alturaLadrilloVoid2 = 40;            // Altura de los
    ladrillos
59.  var rellenoLadrilloVoid2 = 10;           // Relleno de los
    ladrillos
60.  var vacioSuperiorLadrilloVoid2 = 160;    // Vavio Superior de
    los ladrillos
61.  var vacioIzquierdoLadrilloVoid2 = 30;    // vacio de la
    izquierda a los ladrillos
62.  //
63.  var nroFilasLadrillosVoid3 = 5;          // Número de Filas
64.  var nroColumnasLadrillosVoid3 = 2;       // Número de columnas
65.  var anchoLadrilloVoid3 = 140;            // Ancho de los
    ladrillos
66.  var alturaLadrilloVoid3 = 50;            // Altura de los
    ladrillos
67.  var rellenoLadrilloVoid3 = 15;           // Relleno de los
    ladrillos
68.  var vacioSuperiorLadrilloVoid3 = 30;     // Vavio Superior de
    los ladrillos
69.  var vacioIzquierdoLadrilloVoid3 = 20;    // vacio de la
    izquierda a los ladrillos
70.  //
71.  var ladrillosVoid = []; //Conjunto de ladrillos vacios
72.  //
73.  for(var c=0; c<nroColumnasLadrillosVoid; c++){
74.      //Define la primera columna. Lista vertical
75.      ladrillosVoid[c] = [];
76.      for(var f=0; f<nroFilasLadrillosVoid; f++){
77.          ladrillosVoid[c][f] = {x: 0, y: 0, estado: 1};
78.      }
79.  }

```

```

80.  //
81.  var ladrillosVoid2 = []; //Conjunto de ladrillos vacios
82.  //
83.  for(var c2=0; c2<nroColumnasLadrillosVoid2; c2++){
84.    //Define la primera columna. Lista vertical
85.    ladrillosVoid2[c2] = [];
86.    for(var f2=0; f2<nroFilasLadrillosVoid2; f2++){
87.      ladrillosVoid2[c2][f2] = {x: 0, y: 0, estado2: 1};
88.    }
89.  }
90.  //
91.  var ladrillosVoid3 = []; //Conjunto de ladrillos vacios
92.  //
93.  for(var c3=0; c3<nroColumnasLadrillosVoid3; c3++){
94.    //Define la primera columna. Lista vertical
95.    ladrillosVoid3[c3] = [];
96.    for(var f3=0; f3<nroFilasLadrillosVoid3; f3++){
97.      ladrillosVoid3[c3][f3] = {x: 0, y: 0, estado3: 1};
98.    }
99.  }
100. //
101. document.addEventListener("keydown", ManejarTeclaPresionada,
    false); //Observador
102. document.addEventListener("keyup", ManejarTeclaLiberada,
    false); //Observador
103. //
104. //Funciones
105. //
106. function ManejarTeclaPresionada(e){ //Funcion encargada de
    cambiar el estado de la tecla cuando se preiosna(true)
107.   if(e.keyCode == 39){
108.     FlechaDerPVoid = true;
109.   }
110.   else if(e.keyCode == 37){
111.     FlechaIzqPVoid = true;
112.   }
113. }
114. //
115. function ManejarTeclaLiberada(e){ //Funcion encargada de
    cambiar el estado de la tecla cuando se preiosna(false)
116.   if(e.keyCode == 39){
117.     FlechaDerPVoid = false;
118.   }
119.   else if(e.keyCode == 37){
120.     FlechaIzqPVoid = false;
121.   }
122. }
123. //
124. function DeteccionColisionVoid(){
125.   for(var c=0; c<nroColumnasLadrillosVoid; c++){
126.     for(var f=0; f<nroFilasLadrillosVoid; f++){
127.       var b = ladrillosVoid[c][f];
128.       if(b.estado == 1){

```

```

129.         if(x > b.x-40 && x < b.x+anchoLadrilloVoid+5 && y >
           b.y-30 && y < b.y+alturaLadrilloVoid){
130.             dy = -dy;
131.             b.estado = 0;
132.             puntajeVoid++;
133.             if(puntajeVoid ==
               nroFilasLadrillosVoid*nroColumnasLadrillosVoid+nroFilasLadrillos
               Void2*nroColumnasLadrillosVoid2){ //Si el puntaje es igual a la
               operación realizada
134.                 alert("GANASTE!!! FELICITACIONES :3!!!");//Se
               muestra esta alerta
135.                 document.location.reload();//Se reinicia el juego
136.             }
137.         }
138.     }
139. }
140. }
141. }
142. //
143. function DeteccionColisionVoid2(){
144.     for(var c2=0; c2<nroColumnasLadrillosVoid2; c2++){
145.         for(var f2=0; f2<nroFilasLadrillosVoid2; f2++){
146.             var b2 = ladrillosVoid2[c2][f2];
147.             if(b2.estado2 == 1){
148.                 if(x > b2.x-37 && x < b2.x+anchoLadrilloVoid2+5 && y
                   > b2.y-30 && y < b2.y+alturaLadrilloVoid2){
149.                     dy = -dy;
150.                     b2.estado2 = 0;
151.                     puntajeVoid++;
152.                     if(puntajeVoid ==
                       nroFilasLadrillosVoid*nroColumnasLadrillosVoid+nroFilasLadrillos
                       Void2*nroColumnasLadrillosVoid2){
153.                         alert("GANASTE!!! FELICITACIONES :3!!!");
154.                         document.location.reload();
155.                     }
156.                 }
157.             }
158.         }
159.     }
160. }
161. //
162. function DeteccionColisionVoid3(){
163.     for(var c3=0; c3<nroColumnasLadrillosVoid3; c3++){
164.         for(var f3=0; f3<nroFilasLadrillosVoid3; f3++){
165.             var b3 = ladrillosVoid3[c3][f3];
166.             if(b3.estado3 == 1){
167.                 if(x > b3.x-36 && x < b3.x+anchoLadrilloVoid3+5 && y
                   > b3.y-30 && y < b3.y+alturaLadrilloVoid3){
168.                     dy = -dy;
169.                     b3.estado3 = 0;
170.                 }
171.             }
172.         }
173.     }

```



```

174. }
175. //
176. function DibujarPaletaVoid(){ //Funcion que dibuja la paleta
177.     img.src = "img/Nether.png";
178.     ctx.drawImage(img,PaletaPosXVoid, canvas.height-
        AlturaPaletaVoid, AnchuraPaletaVoid, AlturaPaletaVoid);
179.     img.onload = function(){
180.         ctx.drawImage(img,PaletaPosXVoid, canvas.height-
            AlturaPaletaVoid, AnchuraPaletaVoid, AlturaPaletaVoid);
181.     }
182. }
183. //
184. function DibujarpuntajeVoid() {
185.     ctx.font="16px Comic Sans MS";
186.     ctx.fillStyle = "#FF0000";
187.     ctx.fillText("Puntaje:"+puntajeVoid,8,20);
188. }
189. //
190. function DibujarvidasVoid(){
191.     ctx.font="16x Comic Sans MS";
192.     ctx.fillStyle = "#FF0000";
193.     ctx.fillText("Vidas:"+vidasVoid, canvas.width-65,20)
194. }
195. //
196. function ProyectilVoid() { //Función de creación del cuadrado
197.     var img = new Image();
198.     img.src = "img/PROYECTIL.png"; //Imagen utilizada
199.     ctx.drawImage(img, x, y); //Posición de aparición de la
        imagen
200.     img.onload = function(){
201.         ctx.drawImage(img, x, y);
202.     }
203. }
204. //
205.
206. function DibujarLadrillosVoid(){
207.     for(var c=0; c<nroColumnasLadrillosVoid; c++){
208.         for(var f =0; f<nroFilasLadrillosVoid; f++){
209.             if(ladrillosVoid[c][f].estado == 1){
210.                 var brickX =
                    (f*(anchoLadrilloVoid+rellenoLadrilloVoid))+vacioIzquierdoLadril
                    loVoid;
211.                 var brickY =
                    (c*(alturaLadrilloVoid+rellenoLadrilloVoid))+vacioSuperiorLadril
                    loVoid;
212.                 ladrillosVoid[c][f].x = brickX;
213.                 ladrillosVoid[c][f].y = brickY;
214.                 var img = new Image();
215.                 img.src = "img/Piedra.png";
216.                 ctx.drawImage(img,brickX, brickY, anchoLadrilloVoid,
                    alturaLadrilloVoid);
217.                 ctx.onload= function(){
218.                     ctx.drawImage(img,brickX, brickY, anchoLadrilloVoid,
                        alturaLadrilloVoid);

```

```

219.     }
220.   }
221. }
222. }
223. }
224. //
225. function DibujarLadrillosVoid2 () {
226.   for(var c2=0; c2<nroColumnasLadrillosVoid2; c2++){
227.     for(var f2 =0; f2<nroFilasLadrillosVoid2; f2++){
228.       if(ladrillosVoid2[c2][f2].estado2 == 1){
229.         var brickX2 =
          (f2*(anchoLadrilloVoid2+rellenoLadrilloVoid2))+vacioIzquierdoLad
          rilloVoid2;
230.         var brickY2 =
          (c2*(alturaLadrilloVoid2+rellenoLadrilloVoid2))+vacioSuperiorLad
          rilloVoid2;
231.         ladrillosVoid2[c2][f2].x = brickX2;
232.         ladrillosVoid2[c2][f2].y = brickY2;
233.         var img = new Image();
234.         img.src = "img/Madera.png";
235.         ctx.drawImage(img,brickX2, brickY2,
          anchoLadrilloVoid2, alturaLadrilloVoid2);
236.         ctx.onload= function() {
237.           ctx.drawImage(img,brickX2, brickY2,
          anchoLadrilloVoid2, alturaLadrilloVoid2);
238.         }
239.       }
240.     }
241.   }
242. }
243. //
244. function DibujarLadrillosVoid3 () {
245.   for(var c3=0; c3<nroColumnasLadrillosVoid3; c3++){
246.     for(var f3 =0; f3<nroFilasLadrillosVoid3; f3++){
247.       if(ladrillosVoid3[c3][f3].estado3 == 1){
248.         var brickX3 =
          (f3*(anchoLadrilloVoid3+rellenoLadrilloVoid3))+vacioIzquierdoLad
          rilloVoid3;
249.         var brickY3 =
          (c3*(alturaLadrilloVoid3+rellenoLadrilloVoid3))+vacioSuperiorLad
          rilloVoid3;
250.         ladrillosVoid3[c3][f3].x = brickX3;
251.         ladrillosVoid3[c3][f3].y = brickY3;
252.         var img = new Image();
253.         img.src = "img/Nether.png";
254.         ctx.drawImage(img,brickX3, brickY3,
          anchoLadrilloVoid3, alturaLadrilloVoid3);
255.         ctx.onload= function() {
256.           ctx.drawImage(img,brickX3, brickY3,
          anchoLadrilloVoid3, alturaLadrilloVoid3);
257.         }
258.       }
259.     }
260.   }

```

```

261. }
262. //
263. function Dibujarvoid() { //Función de llamado a la anterior
    función
264.     ctx.clearRect(0, 0, canvas.width, canvas.height);
    //Función para limpiar el CANVAS
265.     //
266.     DibujarLadrillosVoid();
267.     //
268.     DibujarLadrillosVoid2();
269.     //
270.     DibujarLadrillosVoid3();
271.     //
272.     ProyectilVoid(); //Llamado de función
273.     x = x + dx; //Nuevo valor para x
274.     y = y + dy; //Nuevo valor para y
275.     //
276.     DibujarPaletaVoid();
277.     DeteccionColisionVoid();
278.     DeteccionColisionVoid2();
279.     DeteccionColisionVoid3();
280.     DibujarpuntajeVoid();
281.     DibujarvidasVoid();
282.     //
283.     if(y + dy > 600 || y + dy < 0) { //Condicion para que el
        cuadrado rebote en las coordenada y
284.         dy = -dy;
285.     }
286.     if(x + dx > 775 || x + dx < 0) { //Condicion para que el
        cuadrado rebote en las coordenada y
287.         dx = -dx;
288.     }
289.     else if(y + dy > 540) { //Condicion para que el
        cuadrado
290.         if(x > PaletaPosXVoid-30 && x < PaletaPosXVoid +
            AnchuraPaletaVoid) { //Se define una condición donde si al
            momento del cuadro pasar de cierta coordenada y y al estar en el
            mismo espacio de la paleta esat rebotara
291.             dy = -dy;
292.         }
293.         else { //De lo contrario se limpiara el intervalo de
            la variable juego que es la que llama la funcion de Dibujar todo
            cada 10 milisegundos
294.             vidasVoid--;
295.             if(!vidasVoid){ //Si vidas es igual a cero
296.                 clearInterval(juego);
297.                 alert("Perdiste :c");
298.                 document.location.reload();
299.             }
300.             else{
301.                 x = canvas.width/2;
302.                 y = canvas.height-60;
303.                 dx = 2
304.                 dy = -2

```

```

305.   PaletaPosXVoid = (canvas.width-AnchuraPaletaVoid)/2;
306.   }
307. }
308. }
309.
310.   if(FlechaDerPVoid && PaletaPosXVoid < canvas.width-
    AnchuraPaletaVoid) {
311.       PaletaPosXVoid += 7;
312.   }
313.   else if(FlechaIzqPVoid && PaletaPosXVoid > 0) { // <-
    -----
314.       PaletaPosXVoid -= 7;
315.   }
316.   x += dx;
317.   y += dy;
318. }
319. }
320. var juego = setInterval(Dibujarvoid, 10);
321. //Final del JavaScript
322. </script>
323.

```

Al ejecutar este código se obtiene la siguiente interfaz visual:



En la figura 9 se puede observar el indicador de vidas



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 fase se terminan ciertos ajustes como cambios de texturas y el posicionamiento de bloques más grandes sobre otros más pequeños para dar la ilusión de deber de pegar más golpes para romperlos completamente.

```

1. <!DOCTYPE html>
2. <html>
3.   <head>
4.     <title>Juego 2D: JavaScript VOID</title>
5.     <style>
6.       * {
7.         padding: 0;
8.         margin: 0;
9.       }
10.      canvas {
11.        background-image: url(img/FONDO.png); /* Imagen
de fondo para el CANVAS*/
12.        display: block;
13.        margin: 0 auto;
14.      }
15.    </style>
16.  </head>
17.  <body>
18.    <center>
19.      <p style="font-family:Montserrat;font-size:
40px;">JUEGO <span style="color:#FF0000">ROMPER TODOS</span> LOS
LADRILLOS</p>
20.    </center>
21.
22.    <canvas id="miCanvas" width="800"
height="600"></canvas> <!--Aqui se agrega el Elemento CANVAS-->
23.    <script> //Inicio del JavaScript
24.
25.      ////////////VARIABLES//////////
26.      var canvas = document.getElementById("miCanvas");//Elemento
Canvas
27.      var ctx = canvas.getContext("2d");
28.      var img = new Image();
29.      //
30.      var x=canvas.width/2; //Coordenadas inicio horizontal
31.      var y=canvas.height -60; //Coordenadas inicio vertical
32.      //
33.      var dx=-3.2; //Dirección del cuadrado respecto a x
34.      var dy=-2; //Dirección del cuadrado respecto a y
35.      //
36.      var AlturaPaletaVoid = 35; //Altura paleta
37.      var AnchuraPaletaVoid = 150; //Anchura paleta

```

```

38.   var PaletaPosXVoid = (canvas.width - AnchuraPaletaVoid) / 2;
    //Poscion paleta
39.   //
40.   var FlechaDerPVoid = false; //Estado de tecla presionada
41.   var FlechaIzqPVoid = false; //Estado de tecla preionada
42.   //
43.   var puntajeVoid = 0; //Estado inicial del puntaje
44.   //
45.   var vidasVoid = 3;
46.   //
47.   var nroFilasLadrillosVoid = 5;           // Número de Filas
48.   var nroColumnasLadrillosVoid = 2;        // Número de columnas
49.   var anchoLadrilloVoid = 120;             // Ancho de los
    ladrillos
50.   var alturaLadrilloVoid = 40;             // Altura de los
    ladrillos
51.   var rellenoLadrilloVoid = 30;            // Relleno de los
    ladrillos
52.   var vacioSuperiorLadrilloVoid = 31;      // Vavio Superior de
    los ladrillos
53.   var vacioIzquierdoLadrilloVoid = 40;     // vacio de la
    izquierda a los ladrillos
54.   //
55.   var nroFilasLadrillosVoid2 = 5;          // Número de Filas
56.   var nroColumnasLadrillosVoid2 = 2;       // Número de columnas
57.   var anchoLadrilloVoid2 = 140;            // Ancho de los
    ladrillos
58.   var alturaLadrilloVoid2 = 40;            // Altura de los
    ladrillos
59.   var rellenoLadrilloVoid2 = 10;           // Relleno de los
    ladrillos
60.   var vacioSuperiorLadrilloVoid2 = 160;    // Vavio Superior de
    los ladrillos
61.   var vacioIzquierdoLadrilloVoid2 = 30;    // vacio de la
    izquierda a los ladrillos
62.   //
63.   var nroFilasLadrillosVoid3 = 5;          // Número de Filas
64.   var nroColumnasLadrillosVoid3 = 2;       // Número de columnas
65.   var anchoLadrilloVoid3 = 140;            // Ancho de los
    ladrillos
66.   var alturaLadrilloVoid3 = 50;            // Altura de los
    ladrillos
67.   var rellenoLadrilloVoid3 = 15;           // Relleno de los
    ladrillos
68.   var vacioSuperiorLadrilloVoid3 = 30;     // Vavio Superior de
    los ladrillos
69.   var vacioIzquierdoLadrilloVoid3 = 20;    // vacio de la
    izquierda a los ladrillos
70.   //
71.   var ladrillosVoid = []; //Conjunto de ladrillos vacios
72.   //
73.   for(var c=0; c<nroColumnasLadrillosVoid; c++){
74.       //Define la primera columna. Lista vertical
75.       ladrillosVoid[c] = [];

```

```

76.     for(var f=0; f<nroFilasLadrillosVoid; f++){
77.         ladrillosVoid[c][f] = {x: 0, y: 0, estado: 1};
78.     }
79. }
80. //
81. var ladrillosVoid2 = []; //Conjunto de ladrillos vacios
82. //
83. for(var c2=0; c2<nroColumnasLadrillosVoid2; c2++){
84.     //Define la primera columna. Lista vertical
85.     ladrillosVoid2[c2] = [];
86.     for(var f2=0; f2<nroFilasLadrillosVoid2; f2++){
87.         ladrillosVoid2[c2][f2] = {x: 0, y: 0, estado2: 1};
88.     }
89. }
90. //
91. var ladrillosVoid3 = []; //Conjunto de ladrillos vacios
92. //
93. for(var c3=0; c3<nroColumnasLadrillosVoid3; c3++){
94.     //Define la primera columna. Lista vertical
95.     ladrillosVoid3[c3] = [];
96.     for(var f3=0; f3<nroFilasLadrillosVoid3; f3++){
97.         ladrillosVoid3[c3][f3] = {x: 0, y: 0, estado3: 1};
98.     }
99. }
100. //
101. document.addEventListener("keydown", ManejarTeclaPresionada,
    false); //Observador
102. document.addEventListener("keyup", ManejarTeclaLiberada,
    false); //Observador
103. //
104. //Funciones
105. //
106. function ManejarTeclaPresionada(e){ //Funcion encargada de
    cambiar el estado de la tecla cuando se preiosna(true)
107.     if(e.keyCode == 39){
108.         FlechaDerPVoid = true;
109.     }
110.     else if(e.keyCode == 37){
111.         FlechaIzqPVoid = true;
112.     }
113. }
114. //
115. function ManejarTeclaLiberada(e){ //Funcion encargada de
    cambiar el estado de la tecla cuando se preiosna(false)
116.     if(e.keyCode == 39){
117.         FlechaDerPVoid = false;
118.     }
119.     else if(e.keyCode == 37){
120.         FlechaIzqPVoid = false;
121.     }
122. }
123. //
124. function DeteccionColisionVoid(){
125.     for(var c=0; c<nroColumnasLadrillosVoid; c++){

```



```

126.     for(var f=0; f<nroFilasLadrillosVoid; f++){
127.         var b = ladrillosVoid[c][f];
128.         if(b.estado == 1){
129.             if(x > b.x-40 && x < b.x+anchoLadrilloVoid+5 && y >
b.y-30 && y < b.y+alturaLadrilloVoid){
130.                 dy = -dy;
131.                 b.estado = 0;
132.                 puntajeVoid++;
133.                 if(puntajeVoid ==
nroFilasLadrillosVoid*nroColumnasLadrillosVoid+nroFilasLadrillos
Void2*nroColumnasLadrillosVoid2){
134.                     alert("GANASTE!!! FELICITACIONES :3!!!");
135.                     document.location.reload();
136.                 }
137.             }
138.         }
139.     }
140. }
141. }
142. //
143. function DeteccionColisionVoid2(){
144.     for(var c2=0; c2<nroColumnasLadrillosVoid2; c2++){
145.         for(var f2=0; f2<nroFilasLadrillosVoid2; f2++){
146.             var b2 = ladrillosVoid2[c2][f2];
147.             if(b2.estado2 == 1){
148.                 if(x > b2.x-37 && x < b2.x+anchoLadrilloVoid2+5 && y
> b2.y-30 && y < b2.y+alturaLadrilloVoid2){
149.                     dy = -dy;
150.                     b2.estado2 = 0;
151.                     puntajeVoid++;
152.                     if(puntajeVoid ==
nroFilasLadrillosVoid*nroColumnasLadrillosVoid+nroFilasLadrillos
Void2*nroColumnasLadrillosVoid2){
153.                         alert("GANASTE!!! FELICITACIONES :3!!!");
154.                         document.location.reload();
155.                     }
156.                 }
157.             }
158.         }
159.     }
160. }
161. //
162. function DeteccionColisionVoid3(){
163.     for(var c3=0; c3<nroColumnasLadrillosVoid3; c3++){
164.         for(var f3=0; f3<nroFilasLadrillosVoid3; f3++){
165.             var b3 = ladrillosVoid3[c3][f3];
166.             if(b3.estado3 == 1){
167.                 if(x > b3.x-36 && x < b3.x+anchoLadrilloVoid3+5 && y
> b3.y-30 && y < b3.y+alturaLadrilloVoid3){
168.                     dy = -dy;
169.                     b3.estado3 = 0;
170.                 }
171.             }
172.         }

```

```

173.     }
174. }
175. //
176. function DibujarPaletaVoid(){ //Funcion que dibuja la paleta
177.     img.src = "img/Nether.png";
178.     ctx.drawImage(img,PaletaPosXVoid, canvas.height-
        AlturaPaletaVoid, AnchuraPaletaVoid, AlturaPaletaVoid);
179.     img.onload = function(){
180.         ctx.drawImage(img,PaletaPosXVoid, canvas.height-
            AlturaPaletaVoid, AnchuraPaletaVoid, AlturaPaletaVoid);
181.     }
182. }
183. //
184. function DibujarpuntajeVoid() {
185.     ctx.font="18px Montserrat";
186.     ctx.fillStyle = "#FF0000";
187.     ctx.fillText("Puntaje:"+puntajeVoid,8,20);
188. }
189. //
190. function DibujarvidasVoid(){
191.     ctx.font="18x Montserrat";
192.     ctx.fillStyle = "#FF0000";
193.     ctx.fillText("VIDAS:"+vidasVoid, canvas.width-80,20)
194. }
195. //
196. function ProyectilVoid() { //Función de creación del cuadrado
197.     var img = new Image();
198.     img.src = "img/PROYECTIL.png"; //Imagen utilizada
199.     ctx.drawImage(img, x, y); //Posición de aparición de la
        imagen
200.     img.onload = function(){
201.         ctx.drawImage(img, x, y);
202.     }
203. }
204. //
205.
206. function DibujarLadrillosVoid(){
207.     for(var c=0; c<nroColumnasLadrillosVoid; c++){
208.         for(var f=0; f<nroFilasLadrillosVoid; f++){
209.             if(ladrillosVoid[c][f].estado == 1){
210.                 var brickX =
                    (f*(anchoLadrilloVoid+rellenoLadrilloVoid))+vacioIzquierdoLadril
                    loVoid;
211.                 var brickY =
                    (c*(alturaLadrilloVoid+rellenoLadrilloVoid))+vacioSuperiorLadril
                    loVoid;
212.                 ladrillosVoid[c][f].x = brickX;
213.                 ladrillosVoid[c][f].y = brickY;
214.                 var img = new Image();
215.                 img.src = "img/Piedra2.png";
216.                 ctx.drawImage(img,brickX, brickY, anchoLadrilloVoid,
                    alturaLadrilloVoid);
217.                 ctx.onload= function(){

```

```

218.         ctx.drawImage(img,brickX, brickY, anchoLadrilloVoid,
                alturaLadrilloVoid);
219.     }
220. }
221. }
222. }
223. }
224. //
225. function DibujarLadrillosVoid2() {
226.     for(var c2=0; c2<nroColumnasLadrillosVoid2; c2++){
227.         for(var f2 =0; f2<nroFilasLadrillosVoid2; f2++){
228.             if(ladrillosVoid2[c2][f2].estado2 == 1){
229.                 var brickX2 =
                    (f2*(anchoLadrilloVoid2+rellenoLadrilloVoid2))+vacioIzquierdoLad
                    rilloVoid2;
230.                 var brickY2 =
                    (c2*(alturaLadrilloVoid2+rellenoLadrilloVoid2))+vacioSuperiorLad
                    rilloVoid2;
231.                 ladrillosVoid2[c2][f2].x = brickX2;
232.                 ladrillosVoid2[c2][f2].y = brickY2;
233.                 var img = new Image();
234.                 img.src = "img/Madera.png";
235.                 ctx.drawImage(img,brickX2, brickY2,
                    anchoLadrilloVoid2, alturaLadrilloVoid2);
236.                 ctx.onload= function() {
237.                     ctx.drawImage(img,brickX2, brickY2,
                    anchoLadrilloVoid2, alturaLadrilloVoid2);
238.                 }
239.             }
240.         }
241.     }
242. }
243. //
244. function DibujarLadrillosVoid3() {
245.     for(var c3=0; c3<nroColumnasLadrillosVoid3; c3++){
246.         for(var f3 =0; f3<nroFilasLadrillosVoid3; f3++){
247.             if(ladrillosVoid3[c3][f3].estado3 == 1){
248.                 var brickX3 =
                    (f3*(anchoLadrilloVoid3+rellenoLadrilloVoid3))+vacioIzquierdoLad
                    rilloVoid3;
249.                 var brickY3 =
                    (c3*(alturaLadrilloVoid3+rellenoLadrilloVoid3))+vacioSuperiorLad
                    rilloVoid3;
250.                 ladrillosVoid3[c3][f3].x = brickX3;
251.                 ladrillosVoid3[c3][f3].y = brickY3;
252.                 var img = new Image();
253.                 img.src = "img/Piedra.png";
254.                 ctx.drawImage(img,brickX3, brickY3,
                    anchoLadrilloVoid3, alturaLadrilloVoid3);
255.                 ctx.onload= function() {
256.                     ctx.drawImage(img,brickX3, brickY3,
                    anchoLadrilloVoid3, alturaLadrilloVoid3);
257.                 }
258.             }

```

```

259.     }
260.   }
261. }
262. //
263. function DibujarVoid() { //Función de llamado a la anterior
    función
264.     //
265.     ctx.clearRect(0, 0, canvas.width, canvas.height);
    //Función para limpiar el CANVAS
266.     //
267.     ProyectilVoid(); //Llamado de función
268.     x = x + dx; //Nuevo valor para x
269.     y = y + dy; //Nuevo valor para y
270.     //
271.     DibujarLadrillosVoid();
272.     //
273.     DibujarLadrillosVoid2();
274.     //
275.     DibujarLadrillosVoid3();
276.     //
277.     DibujarPaletaVoid();
278.     DeteccionColisionVoid();
279.     DeteccionColisionVoid2();
280.     DeteccionColisionVoid3();
281.     DibujarpuntajeVoid();
282.     DibujarvidasVoid();
283.     //
284.     if(y + dy > 600 || y + dy < 0) { //Condicion para que el
        cuadrado rebote en las coordenada y
285.         dy = -dy;
286.     }
287.     if(x + dx > 775 || x + dx < 0) { //Condicion para que el
        cuadrado rebote en las coordenada y
288.         dx = -dx;
289.     }
290.     else if(y + dy > 540) { //Condicion para que el
        cuadrado
291.         if(x > PaletaPosXVoid-30 && x < PaletaPosXVoid +
            AnchuraPaletaVoid) { //Se define una condición donde si al
            momento del cuadro pasar de cierta coordenada y y al estar en el
            mismo espacio de la paleta esat rebotara
292.             dy = -dy;
293.         }
294.         else { //De lo contrario se limpiara el intervalo de
            la variable juego que es la que llama la funcion de Dibujar todo
            cada 10 milisegundos
295.             vidasVoid--;
296.             if(!vidasVoid){
297.                 clearInterval(juego);
298.                 alert("Perdiste :c");
299.                 document.location.reload();
300.             }
301.             else{
302.                 x = canvas.width/2;

```

```

303.         y = canvas.height-60;
304.         dx = -3.5
305.         dy = -2
306.         PaletaPosXVoid = (canvas.width-AnchuraPaletaVoid)/2;
307.     }
308. }
309. }
310.
311.     if(FlechaDerPVoid && PaletaPosXVoid < canvas.width-
AnchuraPaletaVoid) {
312.         PaletaPosXVoid += 7;
313.     }
314.     else if(FlechaIzqPVoid && PaletaPosXVoid > 0) { // <-
-----
315.         PaletaPosXVoid -= 7;
316.     }
317.     x += dx;
318.     y += dy;
319.
320. }
321. var juego = setInterval(Dibujarvoid, 10);
322. //Final del JavaScript
323. </script>
324.
325.

```

Al ejecutar este código se obtiene la siguiente interfaz visual:

JUEGO ROMPER TODOS LOS LADRILLOS



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.





12 CONCLUSIONES

Como conclusión por medio de HTML, CANVAS, CSS y JavaScript podemos desarrollar un juego en entorno 2D que sirve de manera práctica como medio de aprendizaje para la agsinatura.



13 BIBLIOGRAFÍA

[https://developer.mozilla.org/es/docs/Games/Workflows/Famoso juego 2D usando JavaScript puro/Construye grupo bloques](https://developer.mozilla.org/es/docs/Games/Workflows/Famoso_juego_2D_usando_JavaScript_puro/Construye_grupo_bloques)