

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

<JOHN ESTEBAN PERDOMO>

OCTUBRE DE 2020



# 1 CONTENIDO

---

1	CONTENIDO .....	1
2	PRESENTACIÓN .....	2
3	FASE 1: Dibujar y mover una bola .....	3
4	FASE 2: Rebotando en las paredes .....	7
5	FASE 3: Control de la pala y el teclado .....	11
6	FASE 4: Fin del juego .....	17
7	FASE 5: Muro de ladrillos.....	22
8	FASE 6: Detección de colisiones.....	28
9	FASE 7: Contar puntos y ganar .....	34
10	FASE 8: Controlando el ratón .....	40
11	FASE 9: Finalizando el juego.....	46
12	CONCLUSIONES .....	53
13	BIBLIOGRAFÍA.....	54

## 2 PRESENTACIÓN

---

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

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

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

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



**Gráfica 1. Juego 2D en JavaScript**

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

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

**AUTOR: <JOHN ESTEBAN PERDOMO >**



### 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.

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

La correcta visualización 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 y los valores de incremento para modificar su posición.

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, 200)`, llama a la función `dibujar` cada 200 milisegundos.

El código fuente del programa es el siguiente (para darle formato, se deben seguir las instrucciones disponibles en: <https://trabajonomada.com/insertar-codigo-word/> y seguidamente utilizar el enlace: <http://hilito.me/>

A continuación el código:

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

  <!-- Define los estilos de la interfaz visual
        padding es la distancia de un objeto en relación con el marco
que lo contiene
        margin es la distancia que separa a un objeto de otro
        background es el color de fondo
        display: block; Estos elementos fluyen hacia abajo
        margin: 0 auto; Centra el canvas en la pantalla -->
<style>
  * {
    padding: 0;
    margin: 0;
```



```
    }
    canvas {
        background: #02B630;
        display: block;
        margin: 0 auto;
    }
</style>

</head>
<body>

<canvas id="miCanvas" width="400" height="444"></canvas>

<script>
    var canvas = document.getElementById("miCanvas");
    var ctx = canvas.getContext("2d");

    // Coloca x en la mitad del ancho deL CANVAS
    var x = canvas.width/2;

    // Coloca y en la mitad de la altura del CANVAS (restando 30 a dicho
valor)
    var y = canvas.height/-30;

    /* DEFINE LOS INCREMENTOS EN X y en Y. El valor dy es negativo
    para que inicialmente el movimiento de la bola sea hacia arriba */
    var dx = -5;
    var dy = 5;

    function dibujarBola() {
        // Inicia el dibujo
        ctx.beginPath();

        /* Define un círculo en las coordenadas (x, y) con radio 10
        El ángulo va desde 0 hasta 2*PI (360 grados) */
        ctx.arc(x, y, 25, 0, Math.PI*2);

        // Color de llenado
        ctx.fillStyle = "#080A08";

        // Se llena el círculo con el color indicado
        ctx.fill();

        // Finaliza el dibujo
        ctx.closePath();
    }

    /* LA FUNCIÓN dibujar REALIZA TRES TAREAS:
    1) Limpia el CANVAS. Inicio= (0,0) Ancho=canvas.width
    Altura=canvas.height
```

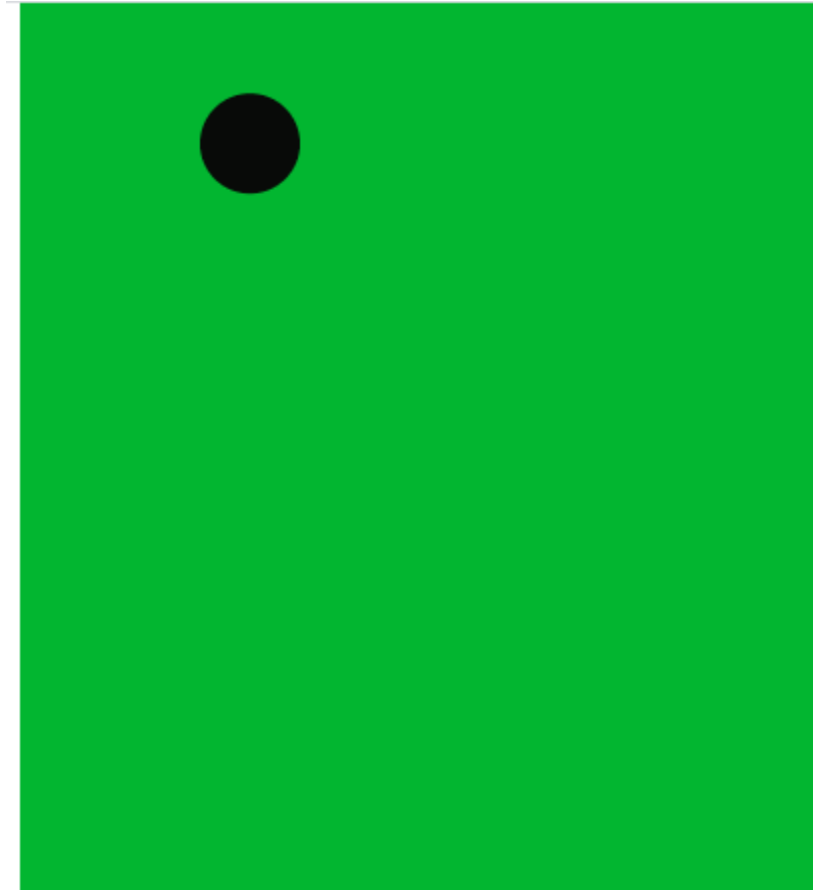


2) Dibuja una bola en las coordenadas (x, y)  
3) Cambiar las coordenadas (x, y) agregando los valores dx, dy  
Con este cambio cada vez que se dibuja la bola, está en una nueva posición \*/

```
function dibujar() {  
  
    // Limpia el CANVAS  
    ctx.clearRect(0, 0, canvas.width, canvas.height);  
  
    // Dibuja la bola  
    dibujarBola();  
  
    // Se incrementa x en el valor dx  
    x = x + dx;  
  
    // Se incrementa y en el valor dy  
    y = y + dy;  
}  
  
/* EJECUTA LA FUNCIÓN dibujar CADA 10 MILISEGUNDOS  
Este es el mecanismo utilizado para construir un sistema que  
ejecuta acciones de manera permanente y periódica */  
setInterval(dibujar, 200);  
</script>  
  
</body>  
</html>
```

Grafica 2;

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 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 rebotará 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.

A continuación el código:

```
<!DOCTYPE html>

<html>

<head>

  <meta charset="utf-8" />
  <title>Juego 2D - lección 02</title>

  <style>

    * { padding: 2; margin: 2; }

    canvas {background: #604A49; display: block; margin: auto; }

  </style>

</head>

<body>

  <canvas id="miCanvas" width="800" height="600"> </canvas>

  <script>
```





```
var canvas = document.getElementById("miCanvas");

var ctx = canvas.getContext("2d");

/* Se agrega la variable ballRadius, la cual define el tamaño
de la bola */
var ballRadius = 35; // <-----

var x = canvas.width / 2;

var y = canvas.height - 30;

var dx = -4;

var dy = -8;
function dibujarBola() {

    ctx.beginPath();

    /* En lugar de un número fijo, se coloca la variable
ballRadius */
    ctx.arc(x, y, ballRadius, 0, Math.PI*2); // <-----
    ---

    ctx.fillStyle = "#eee";

    ctx.fill();

    ctx.closePath();

}

function dibujar() {

    ctx.clearRect(0, 0, canvas.width, canvas.height);

    dibujarBola();
```



```
/* IMPORTANTE:

EL OPERADOR || es el operador lógico OR
Este operador se utiliza para indicar la condición de
conjunción
SI SE CUMPLE UNA CONDICIÓN, O SE CUMPLE OTRA CONDICIÓN,
ENTONCES
SE CUMPLE LA CONDICIÓN

EL OPERADOR && es el operador lógico AND
Este operador se utiliza para indicar la condición de
disyunción
SI SE CUMPLE UNA CONDICIÓN, Y SE CUMPLE OTRA CONDICIÓN
(simultánea), ENTONCES
SE CUMPLE LA CONDICIÓN

*/

/* DESPUÉS DE DIBUJAR LA BOLA, SE DEBEN CAMBIAR LAS
COORDENADAS
DE LA CAJA
-----
TAMAÑO DEL
izquierda)

SI x + dx ES MAYOR AL ANCHO DEL CANVAS O MENOR AL
RADIO DE LA BOLA (caso en el cual se encuentra hacia la
SE CAMBIA LA DIRECCIÓN DE AVANCE HORIZONTAL.
ESTO SE LOGRA CAMBIANDO EL SIGNO DE LA VARIABLE dx
ESTO HACE QUE SE CAMBIE EL SENTIDO DEL MOVIMIENTO
HORIZONTAL */
if (x + dx > canvas.width-ballRadius || x + dx <
ballRadius) { // <-----
    dx = dx * (-1);
}

/* SI y + dy ES MAYOR A LA ALTURA DEL CANVAS O MENOR AL
TAMAÑO DEL
VERTICAL.

RADIO DE LA BOLA, SE CAMBIA LA DIRECCIÓN DEL AVANCE
ESTO SE LOGRA CAMBIANDO EL SIGNO DE LA VARIABLE dy
ESTE CAMBIO EN dy HACE QUE SE MUEVA VERTICALMENTE EN
SENTIDO
OPUESTO */
if (y + dy > canvas.height - ballRadius || y + dy <
ballRadius) { // <-----
    dy = dy * (-1);
}
```

```

/* AQUÍ SE CAMBIA LA POSICIÓN DE LA BOLA. SE TOMA EN
Cuenta las
MODIFICACIONES A dx y dy, EN CASO DE QUE SE HUBIERAN
PRODUCIDO */
    x = x + dx; // EQUIVALE A: x = x + dx; <-----
    y = y + dy; // EQUIVALE A: y = y + dy; <-----
}

setInterval(dibujar, 60);

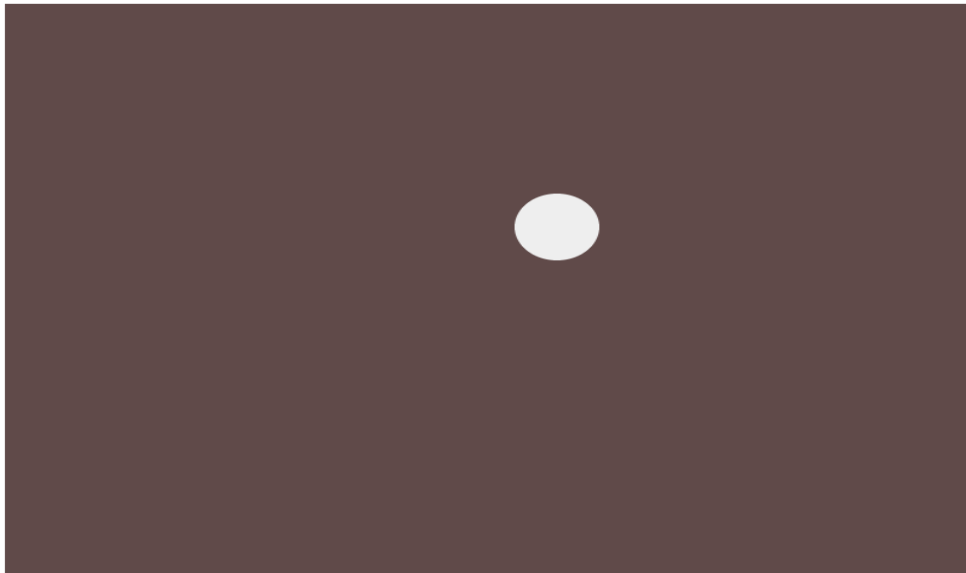
</script>

</body>

</html>

```

Al ejecutar este código se obtiene la siguiente interfaz visual:



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 TECLAD

---

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 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 flechaDerechaPulsada y flechalzquierdaPulsada. Luego de esto se crea una función la cual maneja el movimiento de la tecla presionada y otro de la tecla liberada para esa función utilizamos la teclas a y d que se identifican como 68 y 65.

A continuación el código:

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8" />
  <title>Juego 2D - #03 - Paleta y Control por Teclado</title>
  <style>
    * { padding: 0; margin: 0; }
    canvas { background: #5405D7; display: block; margin: 0 auto; }
  </style>
</head>
<body>

<canvas id="miCanvas" width="800" height="320"></canvas>

<script>
  var canvas = document.getElementById("miCanvas");
  var ctx = canvas.getContext("2d");

  /*
  /* Se cambia el nombre de la variable */
  var radioBola = 20; // <-----

  var x = canvas.width/2;
  var y = canvas.height-30;
  var dx = 2;
  var dy = -2;
```



```
*/

var alturaPaleta = 50; // <-----
var anchuraPaleta = 20; // <-----
var paletaPosX = (canvas.width - anchuraPaleta) / 2; // <-----
-----

var flechaDerechaPulsada = false; // <-----
var flechaIzquierdaPulsada = false; // <-----

/*
*/
document.addEventListener("keydown", manejadorTeclaPresionada,
false); // <-----
document.addEventListener("keyup", manejadorTeclaLiberada, false); //
<-----

// Función que maneja tecla presionada
function manejadorTeclaPresionada(e) { // <-----
    if(e.keyCode == 68) {
        /* e: Es el evento que se produce, en este caso
        tecla presionada. La propiedad: keyCode
permite descubrir de qué tecla se trata. Si el
código es 39, se ha presionado la flecha derecha. En
este caso se coloca la variable:
flechaDerechaPulsada a true
        */
        flechaDerechaPulsada = true;
    }
    else if(e.keyCode == 65) {
        /* e: Es el evento que se produce, en este caso
        tecla presionada. La propiedad: keyCode
permite descubrir de qué tecla se trata. Si el
código es 37, se ha presionado la flecha izquierda. En
este caso se coloca la variable:
flechaIzquierdaPulsada a true
        */
        flechaIzquierdaPulsada = true;
    }
}

// Función que maneja tecla liberada
```



```
function manejadorTeclaLiberada(e) { // <-----
    if(e.keyCode == 68) {
        /* Si la tecla liberada es la 39, se ha dejado de
           presionar la flecha derecha. En este caso, la variable
           se pone en: false
        */
        flechaDerechaPulsada = false;
    }
    else if(e.keyCode == 65) { // <-----
        /* Si la tecla liberada es la 37, se ha dejado de
           presionar la flecha izquierda. En este caso, la
variable
           se pone en: false
        */
        flechaIzquierdaPulsada = false;
    }
}

// Dibuja la bola. Código explicado en anteriores programas
function dibujarBola() {
    ctx.beginPath();
    ctx.arc(x, y, radioBola, 0, Math.PI*2);
    ctx.fillStyle = "#D70541";
    ctx.fill();
    ctx.closePath();
}

function dibujarPaleta() { // <-----
    // Se inicia el dibujo de la paleta
    ctx.beginPath();
    /* Se crea un rectángulo utilizando la posición en X
       El valor de Y está en la base de la pantalla menos la
       altura de la paleta
       Y a continuación se indica la anchura y la altura de la paleta
    */
    ctx.rect(paletaPosX, canvas.height-alturaPaleta, anchuraPaleta,
alturaPaleta);
    ctx.fillStyle = "#D70541";
    ctx.fill();
    // Se "cierra" la paleta, terminando su dibujo
    ctx.closePath();
}

// Función principal. A partir de aquí se origina el proceso
// general del juego
function dibujar() {
    ctx.clearRect(0, 0, canvas.width, canvas.height);

    // En primer lugar, dibuja la bola
    dibujarBola();
}
```



```
// Seguidamente, dibuja la paleta
dibujarPaleta(); // <-----

/* IMPORTANTE:

    EL OPERADOR || es el operador lógico OR
    Este operador se utiliza para indicar la condición de
conjunción
    SI SE CUMPLE UNA CONDICIÓN, O SE CUMPLE OTRA CONDICIÓN,
ENTONCES
    SE CUMPLE LA CONDICIÓN

    EL OPERADOR && es el operador lógico AND
    Este operador se utiliza para indicar la condición de
disyunción
    SI SE CUMPLE UNA CONDICIÓN, Y SE CUMPLE OTRA CONDICIÓN
(simultánea), ENTONCES
    SE CUMPLE LA CONDICIÓN

*/

// Aquí se controla los límites a los que puede llegar la bola
// En caso de intentar sobrepasar dichos límites, se cambia
// el sentido del movimiento
// Este código se explicó en el anterior programa
if(x + dx > canvas.width-radioBola || x + dx < radioBola) { // <-
-----
    dx = -dx;
}
if(y + dy > canvas.height-radioBola || y + dy < radioBola) { //
<-----
    dy = -dy;
}

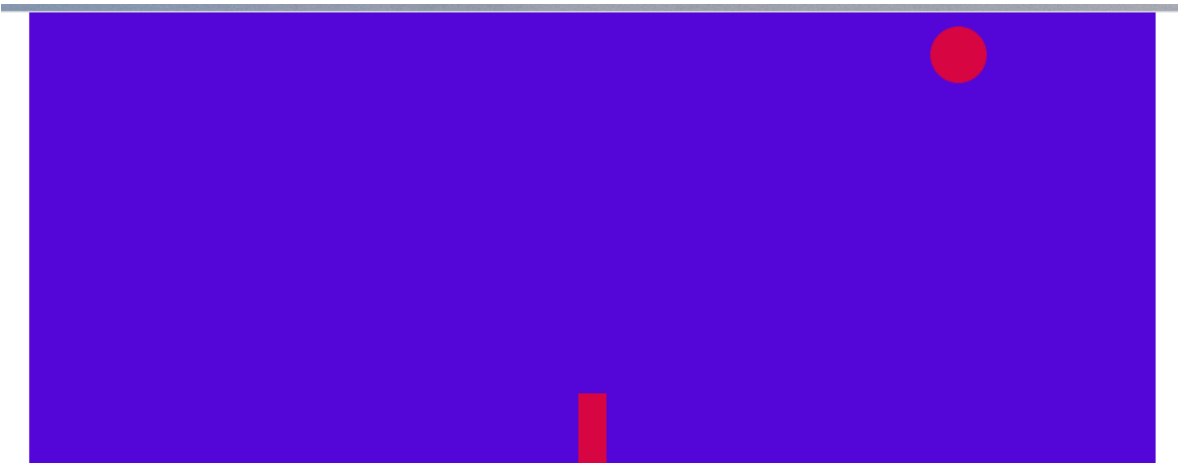
/* Si se ha pulsado la flecha derecha, y la paleta aún puede
desplazarse hacia la derecha sin que se sobrepase el límite de
la
pantalla, entonces se procede a cambiar su posición
En este caso, la función: dibujarPaleta (la cual se ejecuta de
manera cíclica) redibujará la paleta en la nueva posición
*/
if(flechaDerechaPulsada && paletaPosX < canvas.width-
anchuraPaleta) { // <-----
    // Se desplaza la paleta hacia la derecha
    // Aquí, paletaPosX += 7 equivale a: paletaPosX =
paletaPosX + 7
    paletaPosX += 7;
}
```



```
    else if(flechaIzquierdaPulsada && paletaPosX > 0) { // <-----  
---  
        // Se desplaza la paleta hacia la izquierda  
        // Aquí, paletaPosX -= 7 equivale a: paletaPosX =  
paletaPosX - 7  
        paletaPosX -= 7;  
    }  
  
    x += dx;  
    y += dy;  
}  
  
/* Con esta instrucción se crea un ciclo. Cada 10 milisegundos se  
ejecuta la función: dibujar(). Esto genera el ciclo que permitirá  
actualizar el juego, detectar eventos y cambiar el estado  
de los objetos según las nuevas posiciones que ocupen los  
elementos del juego  
  
NOTA: La función que se ejecuta es: dibujar  
Por tanto, dicha función es la encargada de "lanzar" el  
juego  
y dentro de ella se realizarán las acciones que desencadenan  
el juego como tal  
*/  
setInterval(dibujar, 20);  
</script>  
  
</body>  
</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 numero 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 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 sería:  $(y + dy > \text{canvas.height} - \text{radioBola})$  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:  $(x > \text{paletaPosX} \ \&\& \ x < \text{paletaPosX} + \text{anchuraPaleta})$  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.

A continuación el código:

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8" />
  <title>Juego 2D - #04 - Game Over</title>
  <style>* { padding: 3; margin: 0; } canvas { background: #000000;
display: block; margin: 0 auto; }</style>
</head>
<body>

<canvas id="miCanvas" width="500" height="500"></canvas>

<script>
  /* Este programa detecta cuando la bola toca la base de la
pantalla
    Lo anterior significa que la paleta está en otra posición
distinta
    al punto de toque de la bola con la base de la pantalla
    En este caso, se considera que el jugador ha perdido una vida
    El sistema lo informa generando una alerta
    El código se encuentra dentro de la función dibujar
  */

  var canvas = document.getElementById("miCanvas");
  var ctx = canvas.getContext("2d");

  var radioBola = 5;
  var x = canvas.width/2;
  var y = canvas.height-30;
  var dx = 2;
  var dy = -2;
```



```
var alturaPaleta = 5;
var anchuraPaleta = 100;
var paletaPosX = (canvas.width-anchuraPaleta)/2;

var flechaDerechaPresionada = false;
var flechaIzquierdaPresionada = false;

document.addEventListener("keydown", manejadorTeclaPresionada,
false);
document.addEventListener("keyup", manejadorTeclaLiberada, false);

function manejadorTeclaPresionada(e) {
    if(e.keyCode == 68) {
        flechaDerechaPresionada = true;
    }
    else if(e.keyCode == 65) {
        flechaIzquierdaPresionada = true;
    }
}

function manejadorTeclaLiberada(e) {
    if(e.keyCode == 68) {
        flechaDerechaPresionada = false;
    }
    else if(e.keyCode == 65) {
        flechaIzquierdaPresionada = false;
    }
}

function dibujarBola() {
    ctx.beginPath();
    ctx.arc(x, y, radioBola, 0, Math.PI*2);
    ctx.fillStyle = "#0095DD";
    ctx.fill();
    ctx.closePath();
}

function dibujarPaleta() {
    ctx.beginPath();
    ctx.rect(paletaPosX, canvas.height-alturaPaleta, anchuraPaleta,
alturaPaleta);
    ctx.fillStyle = "#0095DD";
    ctx.fill();
    ctx.closePath();
}

function dibujar() {
    ctx.clearRect(0, 0, canvas.width, canvas.height);

    dibujarBola();
    dibujarPaleta();
}
```

```

    if(x + dx > canvas.width-radioBola || x + dx < radioBola) {
        dx = -dx;
    }
    if(y + dy < radioBola) {
        dy = -dy;
    }

    /* Si y + dy alcanza la frontera inferior de la pantalla
       (y + dy > canvas.height - radioBola)
       existe la posibilidad de que el jugador pierda el juego
       Para ello debe evaluarse una segunda opción:
       La variable x determina la posición de la bola
       Lo que debe hacerse es mirar si x está DENTRO de la paleta:
       (x > paletaPosX && x < paletaPosX + anchuraPaleta)
       -----
       Si x está dentro de la paleta, todo va bien y se
       incrementa y
       -----
       Si x NO ESTÁ dentro de la paleta (else), la bola ha
       llegado
       a la frontera inferior, y no encuentra la paleta en su
       camino
       En este caso, SE DETIENE EL CICLO DE ANIMACIÓN, y se
       genera
       un ALERT indicando que el jugador ha perdido (GAME
       OVER)
       */
    else if(y + dy > canvas.height-radioBola) {
        if(x > paletaPosX && x < paletaPosX + anchuraPaleta) {
            dy = -dy;
        }
        else {
            clearInterval(juego);
            alert("PAILA SOCIO");
            document.location.reload();
        }
    }

    if(flechaDerechaPresionada && paletaPosX < canvas.width-
    anchuraPaleta) {
        paletaPosX += 6;
    }
    else if(flechaIzquierdaPresionada && paletaPosX > 0) {
        paletaPosX -= 6;
    }

    x += dx;

```

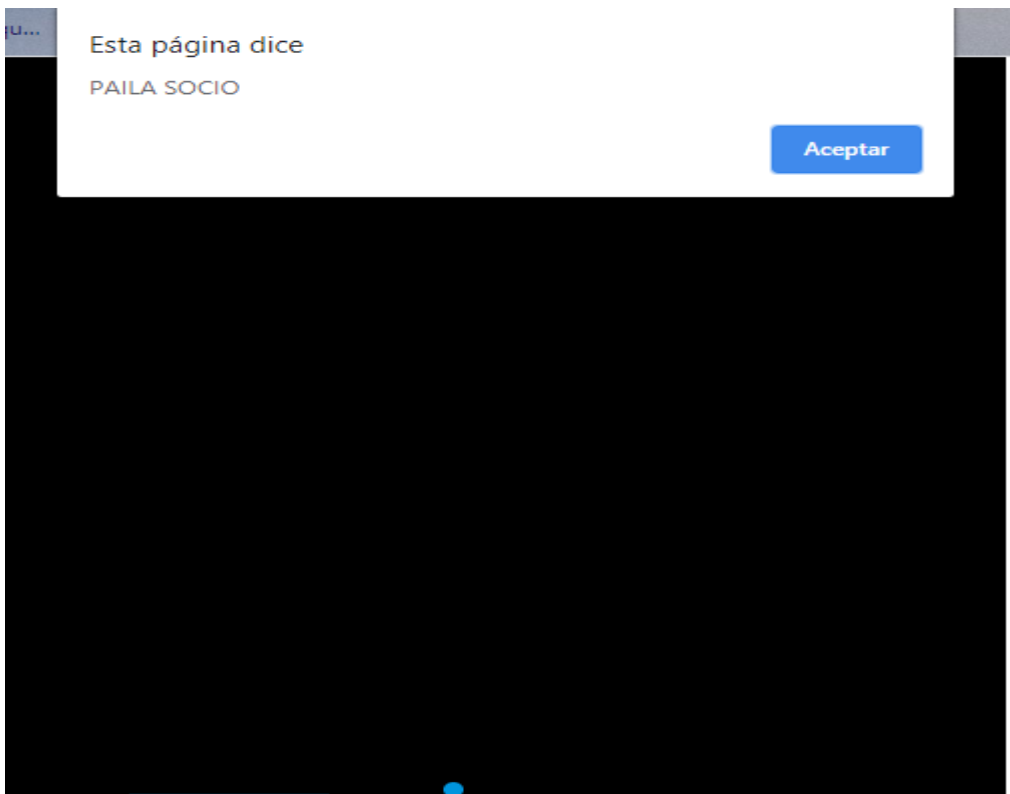


```
        y += dy;
    }

    /* En este programa se asigna a una variable el proceso cíclico
       Esto tiene mucha importancia, porque si en algún momento se
       requiere
       eliminar el ciclo, se utilizará la variable asignada
    */
    var juego = setInterval(dibujar, 5);
</script>

</body>
</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

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 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.

A continuación el código:

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8" />
  <title>Juego 2D: #05 - Construcción de los ladrillos</title>
  <style>* { padding: 5; margin: 0; } canvas { background: #B4C23F;
display: block; margin: 0 auto; }</style>
</head>
<body>

<canvas id="miCanvas" width="400" height="500"></canvas>

<script>
  var canvas = document.getElementById("miCanvas");
  var ctx = canvas.getContext("2d");
  var radioBola = 20;
  var x = canvas.width/6;
  var y = canvas.height/30;
  var dx = 2;
  var dy = -2;
  var alturaPaleta = 25;
  var anchuraPaleta = 30;
  var paletaPosX = (canvas.width-anchuraPaleta)/2;
  var flechaDerechaPresionada = false;
  var flechaIzquierdaPresionada = false;

  /* NUEVAS VARIABLES asociadas a los ladrillos
  */
  var nroFilasLadrillos = 6;
  var nroColumnasLadrillos = 7;
  var anchoLadrillo = 50;
  var alturaLadrillo = 10;
  var rellenoLadrillo = 10;
  var vacioSuperiorLadrillo = 30;
  var vacioIzquierdoLadrillo = 30;

  // Crea el conjunto de ladrillos. Inicialmente, vacío
```

```

var ladrillos = [];

// Recorre cinco columnas
for(var columna=0; columna<nroColumnasLadrillos; columna++) {
    // Define la primera columna. Es una lista vertical
    ladrillos[columna] = [];

    // Para la columna, recorre las tres filas, una después de otra
    for(var fila=0; fila<nroFilasLadrillos; fila++) {
        // Para cada (columna, fila) se define un ladrillo

        /* IMPORTANTE:
           Como se puede observar, cada ladrillo está definido como:
==> ladrillos[c][f]
           Los valores c y f, se corresponden con la fila y la
columna, DENTRO
           DE LA MATRIZ DE LADRILLOS
           -----
           A cada ladrillo en la posición (c, f), se le asignan tres
valores:

           x: Su coordenada horizontal EN LA PANTALLA
           y: Su coordenada vertical EN LA PANTALLA
           -----
           Los valores x y y valen originalmente cero (0)
           Esto cambia cuando se dibujan (más adelante, en la
función: dibujarLadrillos())
           */
        ladrillos[columna][fila] = { x: 0, y: 0 };
    }
}

document.addEventListener("keydown", manejadorTeclaPresionada,
false);
document.addEventListener("keyup", manejadorTeclaLiberada, false);

function manejadorTeclaPresionada(e) {
    if(e.keyCode == 68) {
        flechaDerechaPresionada = true;
    }
    else if(e.keyCode == 65) {
        flechaIzquierdaPresionada = true;
    }
}

function manejadorTeclaLiberada(e) {
    if(e.keyCode == 68) {

```





```
        flechaDerechaPresionada = false;
    }
    else if(e.keyCode == 65) {
        flechaIzquierdaPresionada = false;
    }
}

function dibujarBola() {
    ctx.beginPath();
    ctx.arc(x, y, radioBola, 0, Math.PI*2);
    ctx.fillStyle = "#0D0D0D";
    ctx.fill();
    ctx.closePath();
}

function dibujarPaleta() {
    ctx.beginPath();
    ctx.rect(paletaPosX, canvas.height-alturaPaleta, anchuraPaleta,
alturaPaleta);
    ctx.fillStyle = "#0D0D0D";
    ctx.fill();
    ctx.closePath();
}

/* FUNCIÓN QUE DIBUJA LOS LADRILLOS
-----
*/
function dibujarLadrillos() {
    // Recorre todas las columnas
    for(var columna=0; columna<nroColumnasLadrillos; columna++) {
        // Para cada columna, recorre sus filas
        for(var fila=0; fila<nroFilasLadrillos; fila++) {
            // Calcula la coordenada x del ladrillo, según en que
            // fila se encuentre
            // según el ancho del ladrillo, el valor de relleno
            // interno
            // y el espacio que debe dejar a la izquierda
            // NOTA: Se sugiere asignar valores y dibujar el esquema
            // a mano
            var brickX =
            (fila*(anchoLadrillo+rellenoLadrillo))+vacioIzquierdoLadrillo;

            // Repite el proceso para calcular la coordenada y del
            // ladrillo
            var brickY =
            (columna*(alturaLadrillo+rellenoLadrillo))+vacioSuperiorLadrillo;

            // ASIGNA AL LADRILLO EN LA columna, fila QUE LE
            // CORRESPONDE EN LA MATRIZ
            // EL VALOR CALCULADO (brickX) A SU COORDENADA x
```



```
ladrillos[columna][fila].x = brickX;

// IGUAL PARA EL VALOR y EN PANTALLA
ladrillos[columna][fila].y = brickY;

// DIBUJA EL LADRILLO CON LOS VALORES ASOCIADOS:
// Coordenada: (brickX, brickY)
// Anchura: anchoLadrillo
// Altrua: alturaLadrillo
ctx.beginPath();
ctx.rect(brickX, brickY, anchoLadrillo, alturaLadrillo);
ctx.fillStyle = "#F8F6F6";
ctx.fill();
ctx.closePath();

// COMO SE RECORRE TODO EL CICLO, SE DIBUJAN TODOS LOS
LADRILLOS
    }
}

function dibujar() {
    ctx.clearRect(0, 0, canvas.width, canvas.height);

    // DIBUJA EL CONJUNTO DE LADRILLOS
    dibujarLadrillos();

    dibujarBola();
    dibujarPaleta();

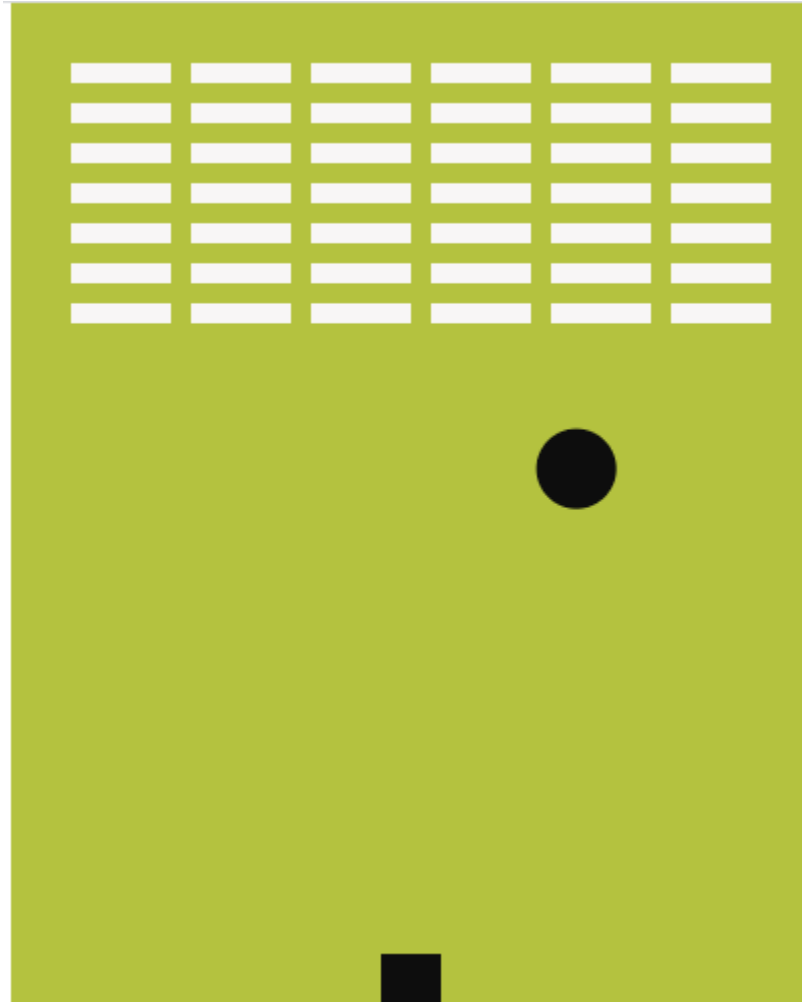
    if(x + dx > canvas.width-radioBola || x + dx < radioBola) {
        dx = -dx;
    }
    if(y + dy < radioBola) {
        dy = -dy;
    }
    else if(y + dy > canvas.height-radioBola) {
        if(x > paletaPosX && x < paletaPosX + anchuraPaleta) {
            dy = -dy;
        }
        else {
            clearInterval(juego);
            alert("uy casi mi rey");

            // RECARGA LA PÁGINA - El juego vuelve a empezar
            document.location.reload();
        }
    }
}
```



```
        if(flechaDerechaPresionada && paletaPosX < canvas.width-  
anchuraPaleta) {  
            paletaPosX += 8;  
        }  
        else if(flechaIzquierdaPresionada && paletaPosX > 0) {  
            paletaPosX -= 8;  
        }  
  
        x += dx;  
        y += dy;  
    }  
  
    var juego = setInterval(dibujar, 5);  
</script>  
  
</body>  
</html>
```

Al ejecutar este código se obtiene la siguiente interfaz visual:



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 deteccionColision()`, 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.

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8" />
  <title>Juego 2D - #06 - Detección de colisión</title>
  <style>* { padding: 6; margin: 0; } canvas { background: #F0FC00;
display: block; margin: 0 auto; }</style>
</head>
<body>

<canvas id="miCanvas" width="480" height="320"></canvas>

<script>
  var canvas = document.getElementById("miCanvas");
  var ctx = canvas.getContext("2d");

  var radioBola = 30;
  var x = canvas.width/2;
  var y = canvas.height-30;
  var dx = 2;
  var dy = -2;

  var alturaPaleta = 10;
  var anchuraPaleta = 90;
  var paletaPosX = (canvas.width-anchuraPaleta)/2;

  var flechaDerechaPresionada = false;
  var flechaIzquierdaPresionada = false;

  var nroFilasLadrillos = 10;
  var nroColumnasLadrillos = 4;
  var anchuraLadrillo = 30;
  var alturaLadrillo = 20;
```



```
var rellenoLadrillo = 10;
var vacioSuperiorLadrillo = 30;
var vacioIzquierdoLadrillo = 30;

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

        /* IMPORTANTE:
           Como se puede observar, cada ladrillo está
definido como: ==> ladrillos[c][f]
           Los valores c y f, se corresponden con la fila
y la columna, DENTRO
           DE LA MATRIZ DE LADRILLOS
           -----
           A cada ladrillo en la posicion (c, f), se le
asignan tres valores:

           x: Su coordenada horizontal EN LA PANTALLA
           y: Su coordenada vertical EN LA PANTALLA
           status: Indica si está visible o invisible.

1 = Visible, 0 = INVISIBLE

           Inicialmente el ladrillo debe esta visible. Si
la bola "toca" al ladrillo,
           el ladrillo se debe volver INVISIBLE (status =
0)
           -----
           Los valores x y y valen originalmente cero (0)
           Esto cambia cuando se dibujan (más adelante, en
la función: dibujarLadrillos())
           */
        ladrillos[c][f] = { x: 0, y: 0, status: 1 };
    }
}

document.addEventListener("keydown", manejadorTeclaPresionada,
false);
document.addEventListener("keyup", manejadorTeclaLiberada, false);

function manejadorTeclaPresionada(e) {
    if(e.keyCode == 68) {
        flechaDerechaPresionada = true;
    }
    else if(e.keyCode == 65) {
        flechaIzquierdaPresionada = true;
    }
}
```



```
}

function manejadorTeclaLiberada(e) {
    if(e.keyCode == 68) {
        flechaDerechaPresionada = false;
    }
    else if(e.keyCode == 65) {
        flechaIzquierdaPresionada = false;
    }
}

// EN ESTA FUNCIÓN SE DETECTA LA COLISIÓN DE LA BOLA CON EL LADRILLO

function deteccionColision() {

    // LOS DOS CICLOS SIGUIENTES RECORREN TODOS LOS LADRILLOS
    for(var c=0; c<nroColumnasLadrillos; c++) {
        for(var f=0; f<nroFilasLadrillos; f++) {

            // EN ESTE PUNTO SE TIENE EL LADRILLO SITUADO EN: (c, f)
            // SE CREA UNA VARIABLE TEMPORAL PARA EL LADRILLO
            var b = ladrillos[c][f];

            // SI EL LADRILLO ES VISIBLE, se debe verificar si entra
            en contacto con la bola
            if(b.status == 1) {

                /* SI LAS COORDENADAS x y y, SE ENCUENTRAN DENTRO
                DE LAS COORDENADAS
                DEL LADRILLO (aspecto que se verifica con las
                condiciones mostradas)
                LA BOLA HA IMPACTADO CONTRA EL LADRILLO
                En este caso, se modifica la coordenada y, PERO
                LÓ MÁS IMPORTANTE
                ES QUE SE COLOCA EL VALOR DE status A CERO,
                HACIENDO QUE EL LADRILLO
                SE VUELVA INVISIBLE
                */

                if(x > b.x && x < b.x+anchuraLadrillo && y > b.y && y
                < b.y+alturaLadrillo) {
                    dy = -dy;
                    b.status = 0;
                }
            }
        }
    }
}

function dibujarBola() {
```

```

        ctx.beginPath();
        ctx.arc(x, y, radioBola, 0, Math.PI*2);
        ctx.fillStyle = "#760C09";
        ctx.fill();
        ctx.closePath();
    }

    function dibujarPaleta() {
        ctx.beginPath();
        ctx.rect(paletaPosX, canvas.height-alturaPaleta, anchuraPaleta,
alturaPaleta);
        ctx.fillStyle = "#760C09";
        ctx.fill();
        ctx.closePath();
    }

    function dibujarLadrillos() {
        for(var c=0; c<nroColumnasLadrillos; c++) {
            for(var f=0; f<nroFilasLadrillos; f++) {

                /* IMPORTANTE:

                Solamente se dibujan los ladrillos que están VISIBLES
                Se sabe que el ladrillo es visible cuando: status == 1
                Los ladrillos INVISIBLES NO SE DIBUJAN

                */

                if(ladrillos[c][f].status == 1) {

                    // SE DIBUJA EL LADRILLO
                    var brickX =
(f*(anchuraLadrillo+rellenoLadrillo))+vacioIzquierdoLadrillo;
                    var brickY =
(c*(alturaLadrillo+rellenoLadrillo))+vacioSuperiorLadrillo;
                    ladrillos[c][f].x = brickX;
                    ladrillos[c][f].y = brickY;
                    ctx.beginPath();
                    ctx.rect(brickX, brickY, anchuraLadrillo,
alturaLadrillo);

                    ctx.fillStyle = "#000000";
                    ctx.fill();
                    ctx.closePath();

                }
            }
        }
    }

    function dibujar() {
        ctx.clearRect(0, 0, canvas.width, canvas.height);
    }

```





```
dibujarLadrillos();
dibujarBola();
dibujarPaleta();
deteccionColision();

if(x + dx > canvas.width-radioBola || x + dx < radioBola) {
    dx = -dx;
}
if(y + dy < radioBola) {
    dy = -dy;
}
else if(y + dy > canvas.height-radioBola) {
    if(x > paletaPosX && x < paletaPosX + anchuraPaleta) {
        dy = -dy;
    }
    else {
        // Detiene el ciclo del juego
        clearInterval(juego);
        // Genera mensaje, pues el jugador ha perdido
        alert("uy casi mi rey");
        // Recarga la página, para iniciar de nuevo el juego
        document.location.reload();
    }
}

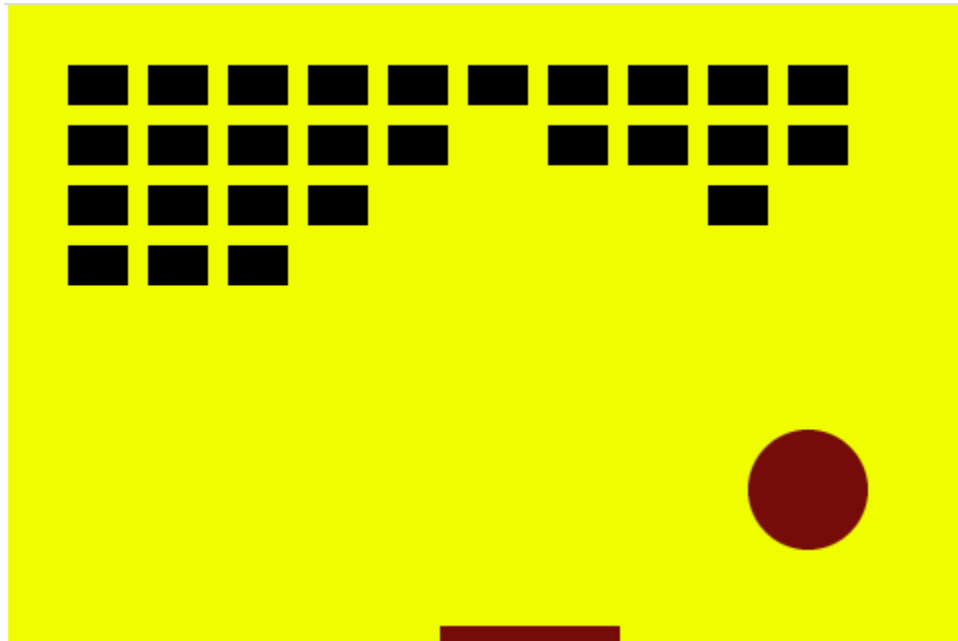
if(flechaDerechaPresionada && paletaPosX < canvas.width-
anchuraPaleta) {
    paletaPosX += 7;
}
else if(flechaIzquierdaPresionada && paletaPosX > 0) {
    paletaPosX -= 7;
}

x += dx;
y += dy;
}

var juego = setInterval(dibujar, 3);
</script>

</body>
</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 que nos dice 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 numero de ladrillos haciendo que el juego se gane.

A continuación el código:

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8" />
  <title>Juego 2D - #07 - Control de juego ganado</title>
  <!-- EN ESTE EJEMPLO SE CAMBIA LA ANCHURA DE LA PALETA
        ESTO ES CLAVE PARA PERMITIR QUE EL JUEGO SEA AUTOMÁTICO
        Y SE PUEDA VERIFICAR EL OBJETIVO DEL JUEGO Y EL JUGADOR GANE -->
  <style>* { padding: 0; margin: 0; } canvas { background: #0979B1;
display: block; margin: 0 auto; }</style>
</head>
<body>

<canvas id="miCanvas" width="600" height="420"></canvas>

<script>
  var canvas = document.getElementById("miCanvas");
  var ctx = canvas.getContext("2d");

  var radioBola = 3;
  var x = canvas.width/30;
  var y = canvas.height/30;
  var dx = 2;
  var dy = -2;
  var alturaPaleta = 10;

  // EL ANCHO DE LA PALETA ES 480. ESTE ES EL MISMO ANCHO DEL CANVAS
  // Con esto se garantiza que el juego termine
  var anchuraPaleta = 500;

  var paletaPosX = (canvas.width-anchuraPaleta)/2;
  var flechaDerechaPresionada = false;
  var flechaIzquierdaPresionada = false;

  var nroFilasLadrillos = 2;
  var nroColumnasLadrillos = 6;
```



```
var anchuraLadrillos = 75;
var alturaLadrillos = 20;
var rellenoLadrillos = 10;
var vacioSuperiorLadrillo = 30;
var vacioIzquierdoLadrillo = 30;

// LA VARIABLE puntaje ,
// la variable "puntaje" se CONTROLA EL NÚMERO DE LADRILLOS QUE HAN
SIDO
// IMPACTADOS POR LA BOLA. Cada vez que la bola golpee un
ladrilloincrementa en uno
var puntaje = 0;

var ladrillos = [];
for(var c=0; c<nroColumnasLadrillos; c++) {
    ladrillos[c] = [];
    for(var f=0; f<nroFilasLadrillos; f++) {
        ladrillos[c][f] = { x: 0, y: 0, estado: 1 };
    }
}

document.addEventListener("keydown", manejadorTeclaPresionada,
false);
document.addEventListener("keyup", manejadorTeclaLiberada, false);

function manejadorTeclaPresionada(e) {
    if(e.keyCode == 39) {
        flechaDerechaPresionada = true;
    }
    else if(e.keyCode == 37) {
        flechaIzquierdaPresionada = true;
    }
}

function manejadorTeclaLiberada(e) {
    if(e.keyCode == 39) {
        flechaDerechaPresionada = false;
    }
    else if(e.keyCode == 37) {
        flechaIzquierdaPresionada = false;
    }
}

function detectarColision() {
    for(var c=0; c<nroColumnasLadrillos; c++) {
        for(var f=0; f<nroFilasLadrillos; f++) {
            var b = ladrillos[c][f];
            if(b.estado == 1) {
                if(x > b.x && x < b.x+anchuraLadrillos && y > b.y &&
y < b.y+alturaLadrillos) {
                    dy = -dy;
                    b.estado = 0;
                }
            }
        }
    }
}
```

```

// LA INSTRUCCIÓN puntaje++ EQUIVALE A: puntaje =
puntaje + 1
-----
// EN ESTE PUNTO DEL CÓDIGO LA BOLA HA IMPACTADO
UN LADRILLO
// POR ESTE MOTIVO, SE INCREMENTA EL VALOR DE
puntaje
// Si el puntaje es igual al número total de
ladrillos (valor que
// se obtiene multiplicando el número de filas de
ladrillos por el
// número de columnas de ladrillos), entonces el
jugador ha ganado
    puntaje++;
    if(puntaje ==
nroFilasLadrillos*nroColumnasLadrillos) {
        alert("RE FACHERO MI PANA");
        document.location.reload();
    }
}
}
}
}

function dibujarBola() {
    ctx.beginPath();
    ctx.arc(x, y, radioBola, 0, Math.PI*2);
    ctx.fillStyle = "#080808";
    ctx.fill();
    ctx.closePath();
}

function dibujarPaleta() {
    ctx.beginPath();
    ctx.rect(paletaPosX, canvas.height-alturaPaleta, anchuraPaleta,
alturaPaleta);
    ctx.fillStyle = "#080808";
    ctx.fill();
    ctx.closePath();
}

function dibujarLadrillos() {
    for(var c=0; c<nroColumnasLadrillos; c++) {
        for(var r=0; r<nroFilasLadrillos; r++) {
            if(ladrillos[c][r].estado == 1) {

```

```

        var posXLadrillo =
(r*(anchuraLadrillos+rellenoLadrillos))+vacioIzquierdoLadrillo;
        var posYLadrillo =
(c*(alturaLadrillos+rellenoLadrillos))+vacioSuperiorLadrillo;
        ladrillos[c][r].x = posXLadrillo;
        ladrillos[c][r].y = posYLadrillo;
        ctx.beginPath();
        ctx.rect(posXLadrillo, posYLadrillo,
anchuraLadrillos, alturaLadrillos);
        ctx.fillStyle = "#9C0303";
        ctx.fill();
        ctx.closePath();
    }
}
}

function dibujarPuntaje() {
    ctx.font = "24px Arial";
    ctx.fillStyle = "#F3ECEC";
    ctx.fillText("puntaje: "+puntaje, 8, 20);
}

function dibujar() {
    ctx.clearRect(0, 0, canvas.width, canvas.height);
    dibujarLadrillos();
    dibujarBola();
    dibujarPaleta();
    dibujarPuntaje();
    detectarColision();

    if(x + dx > canvas.width-radioBola || x + dx < radioBola) {
        dx = -dx;
    }
    if(y + dy < radioBola) {
        dy = -dy;
    }
    else if(y + dy > canvas.height-radioBola) {
        if(x > paletaPosX && x < paletaPosX + anchuraPaleta) {
            dy = -dy;
        }
        else {
            clearInterval(juego);
            alert("malo con m de feo");
            document.location.reload();
        }
    }

    if(flechaDerechaPresionada && paletaPosX < canvas.width-
anchuraPaleta) {

```

```

        paletaPosX += 7;
    }
    else if(flechaIzquierdaPresionada && paletaPosX > 0) {
        paletaPosX -= 7;
    }

    x += dx;
    y += dy;
}

var juego = setInterval(dibujar, 7);
</script>

</body>
</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)` a la cual se le da una variable y una condición que al cumplirla hace que la paleta pueda ser desplazada mediante el mouse.

A continuación el código;

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8" />
  <title>Juego 2D - #08 - Utilizando el ratón</title>
  <style>* { padding: 9; margin: 1; } canvas { background: #11F5CC;
display: block; margin: 0 auto; }</style>
</head>
<body>

<canvas id="miCanvas" width="480" height="320"></canvas>

<script>
  var canvas = document.getElementById("miCanvas");
  var ctx = canvas.getContext("2d");

  var radioBola = 6;
  var x = canvas.width/2;
  var y = canvas.height-30;
  var dx = 2;
  var dy = -2;

  var alturaPaleta = 20;
  var anchuraPaleta = 100;
  var paletaPosX = (canvas.width-anchuraPaleta)/2;

  var flechaDerechaPresionada = false;
  var flechaIzquierdaPresionada = false;

  var nroFilasLadrillos = 10;
  var nroColumnasLadrillos = 7;
  var anchuraLadrillo = 30;
  var alturaLadrillo = 10;
  var rellenoLadrillo = 10;
  var vacioSuperiorLadrillo = 30;
  var vacioIzquierdoLadrillo = 30;
```



```
var puntaje = 0;

var ladrillos = [];
for(var c=0; c<nroColumnasLadrillos; c++) {
    ladrillos[c] = [];
    for(var f=0; f<nroFilasLadrillos; f++) {
        ladrillos[c][f] = { x: 0, y: 0, estado: 1 };
    }
}

document.addEventListener("keydown", manejadorTeclaPresionada,
false);
document.addEventListener("keyup", manejadorTeclaLiberada, false);

// PARA DETECTAR EL MOVIMIENTO DEL RATÓN, SE COLOCA UN ESCUCHADOR
(listener)
// AL EVENTO "mousemove"
document.addEventListener("mousemove", manejadorRaton, false);

function manejadorTeclaPresionada(e) {
    if(e.keyCode == 68) {
        flechaDerechaPresionada = true;
    }
    else if(e.keyCode == 65) {
        flechaIzquierdaPresionada = true;
    }
}

function manejadorTeclaLiberada(e) {
    if(e.keyCode == 68) {
        flechaDerechaPresionada = false;
    }
    else if(e.keyCode == 65) {
        flechaIzquierdaPresionada = false;
    }
}

// ESTE ES EL MANEJADOR DEL RATÓN
// -----
// La instrucción: "offsetLeft" calcula la distancia desde el borde
izquierdo
// de la pantalla hasta un componente html
// -----
// Por tanto, la instrucción: "canvas.offsetLeft" calcula el espacio
a la izquierda
// del objeto CANVAS
// -----
// Dentro del manejador del ratón, la instrucción: "e.clientX"
calcula la posición
```



```
// del ratón en la pantalla. Para calcular la posición del ratón
DENTRO del CANVAS
// debemos RESTAR a la posición X del ratón, el valor izquierdo del
CANVAS
// -----
// Es decir: "e.clientX - canvas.offsetLeft"
// -----
function manejadorRaton(e) {
    var posXRatonDentroDeCanvas = e.clientX - canvas.offsetLeft;
    // EL SIGUIENTE if DETERMINA SI LA POSICIÓN X DEL RATÓN ESTÁ
    // DENTRO DEL CANVAS
    if(posXRatonDentroDeCanvas > 0 && posXRatonDentroDeCanvas <
canvas.width) {
        // SI LA RESPUESTA ES POSITIVA, EL RATÓN ESTÁ DENTRO DEL
CANVAS

        // EN ESTE CASO, SE RECALCULA LA POSICIÓN DE LA PALETA
        // SU VALOR X ES AHORA LA POSICIÓN X DEL RATÓN
        // -----
--
        // PERO DEBE RECORDARSE QUE LA PALETA TIENE UN ANCHO. ESTA ES
LA RAZÓN
        // POR LA CUAL SE DEBE RESTAR A LA POSICIÓN X DE LA PALETA
LA MITAD DEL
        // ANCHO DE LA PALETA
        // -----
--
        // AL HACER ESTO, LA PALETA MODIFICA SU POSICIÓN CON BASE EN
EL

        // MOVIMIENTO DEL RATÓN
        paletaPosX = posXRatonDentroDeCanvas - anchuraPaleta/2;
    }
}

function detectarColision() {
    for(var c=0; c<nroColumnasLadrillos; c++) {
        for(var r=0; r<nroFilasLadrillos; r++) {
            var b = ladrillos[c][r];
            if(b.estado == 1) {
                if(x > b.x && x < b.x+anchuraLadrillo && y > b.y && y
< b.y+alturaLadrillo) {
                    dy = -dy;
                    b.estado = 0;
                    puntaje++;
                    if(puntaje ==
nroFilasLadrillos*nroColumnasLadrillos) {
                        alert("excelente mi rey!!!");
                        document.location.reload();
                    }
                }
            }
        }
    }
}
```



```
    }  
  }  
  
  function dibujarBola() {  
    ctx.beginPath();  
    ctx.arc(x, y, radioBola, 0, Math.PI*2);  
    ctx.fillStyle = "#138D75";  
    ctx.fill();  
    ctx.closePath();  
  }  
  function dibujarPaleta() {  
    ctx.beginPath();  
    ctx.rect(paletaPosX, canvas.height-alturaPaleta, anchuraPaleta,  
alturaPaleta);  
    ctx.fillStyle = "#138D75";  
    ctx.fill();  
    ctx.closePath();  
  }  
  function dibujarLadrillos() {  
    for(var c=0; c<nroColumnasLadrillos; c++) {  
      for(var r=0; r<nroFilasLadrillos; r++) {  
        if(ladrillos[c][r].estado == 1) {  
          var brickX =  
(r*(anchuraLadrillo+rellenoLadrillo))+vacioIzquierdoLadrillo;  
          var brickY =  
(c*(alturaLadrillo+rellenoLadrillo))+vacioSuperiorLadrillo;  
          ladrillos[c][r].x = brickX;  
          ladrillos[c][r].y = brickY;  
          ctx.beginPath();  
          ctx.rect(brickX, brickY, anchuraLadrillo,  
alturaLadrillo);  
          ctx.fillStyle = "#0095DD";  
          ctx.fill();  
          ctx.closePath();  
        }  
      }  
    }  
  }  
  
  function dibujarPuntaje() {  
    ctx.font = "44px Arial";  
    ctx.fillStyle = "#0095DD";  
    ctx.fillText("puntaje: "+puntaje, 8, 20);  
  }  
  
  function dibujar() {  
    ctx.clearRect(0, 0, canvas.width, canvas.height);  
    dibujarLadrillos();  
    dibujarBola();  
    dibujarPaleta();  
    dibujarPuntaje();  
  }  
}
```

```

    detectarColision();

    if(x + dx > canvas.width-radioBola || x + dx < radioBola) {
        dx = -dx;
    }
    if(y + dy < radioBola) {
        dy = -dy;
    }
    else if(y + dy > canvas.height-radioBola) {
        if(x > paletaPosX && x < paletaPosX + anchuraPaleta) {
            dy = -dy;
        }
        else {
            clearInterval(juego);
            alert("MAS MALO QUE VOMITAR SANGRE");
            document.location.reload();
        }
    }

    if(flechaDerechaPresionada && paletaPosX < canvas.width-
anchuraPaleta) {
        paletaPosX += 7;
    }
    else if(flechaIzquierdaPresionada && paletaPosX > 0) {
        paletaPosX -= 7;
    }

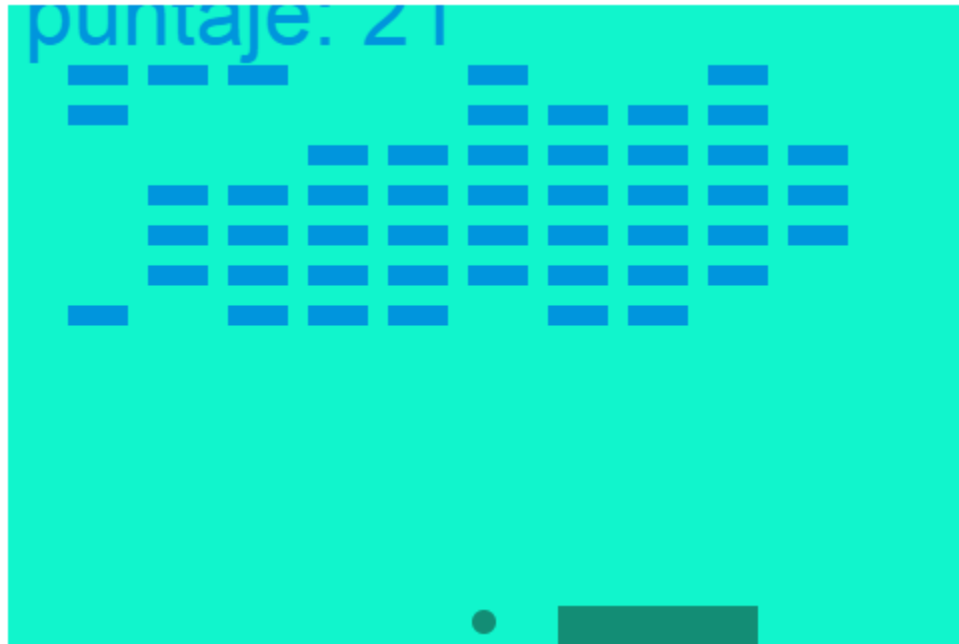
    x += dx;
    y += dy;
}

var juego = setInterval(dibujar, 3);
</script>

</body>
</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

---

Esta es la etapa final del juego aquí se incluye las vidas que tiene el jugador

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

A continuación el código:

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8" />
  <title>Juego 2D - #09 - Juego completo</title>
  <!-- 1. Se oculta el ratón
        2. Se agregan vidas al jugador
        3. Ya no se utiliza "setInterval" -->
  <style>* { padding: 20; margin: 0; } canvas { background: #0EA55A;
display: block; margin: 0 auto; * {cursor: none;} } </style>
</head>
<body>

<canvas id="miCanvas" width="800" height="700"></canvas>

<script>
  var canvas = document.getElementById("miCanvas");
  var ctx = canvas.getContext("2d");

  var bolaRadio = 10 ;
  var x = canvas.width/2;
  var y = canvas.height-30;
  var dx = 2;
  var dy = -2;

  var alturaPaleta = 10;
  var anchuraPaleta = 75;
  var paletaPosX = (canvas.width-anchuraPaleta)/2;

  var flechaDerechaPresionada = false;
  var flechaIzquierdaPresionada = false;

  var nroFilasLadrillos = 8;
  var nroColumnasLadrillos = 7;
  var anchuraLadrillo = 75;
  var alturaLadrillo = 20;
```



```
var rellenoLadrillo = 10;
var vacioSuperiorLadrillo = 30;
var vacioIzquierdoLadrillo = 30;

var puntaje = 0;

// ESTA INSTRUCCIÓN CONTROLA EL NÚMERO DE VIDAS DEL JUGADOR
// CUANDO LA INSTRUCCIÓN vidas DISMINUYE A CERO, EL JUGADOR PIERDE,
// PUESTO QUE HA PERDIDO TRES VECES
var vidas = 5;

// ESTA VARIABLE DEFINE UN COLOR
// Se pueden utilizar otros colores para los diferentes elementos del
juego
var colorFigura = "#060707";
var colorBola = "#00ffff";
var colorPaleta = "#0000ff";
var colorLadrillo = "#DFF100";
var colorTexto = "#00ffff";

// ESTA INSTRUCCIÓN OCULTA EL CURSOR DEL RATON (DENTRO DEL CANVAS)
canvas.style.cursor = 'none';

var ladrillos = [];
for(var c=0; c<nroColumnasLadrillos; c++) {
    ladrillos[c] = [];
    for(var f=0; f<nroFilasLadrillos; f++) {
        ladrillos[c][f] = { x: 0, y: 0, estado: 1 };
    }
}

document.addEventListener("keydown", manejadorTeclaPresionada,
false);
document.addEventListener("keyup", manejadorTeclaLiberada, false);
document.addEventListener("mousemove", manejadorRaton, false);

function manejadorTeclaPresionada(e) {
    if(e.keyCode == 68) {
        flechaDerechaPresionada = true;
    }
    else if(e.keyCode == 65) {
        flechaIzquierdaPresionada = true;
    }
}

function manejadorTeclaLiberada(e) {
    if(e.keyCode == 68) {
        flechaDerechaPresionada = false;
    }
    else if(e.keyCode == 65) {
```



```

        flechaIzquierdaPresionada = false;
    }
}

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

function detectarColision() {
    for(var c=0; c<nroColumnasLadrillos; c++) {
        for(var f=0; f<nroFilasLadrillos; f++) {
            var b = ladrillos[c][f];
            if(b.estado == 1) {
                if(x > b.x && x < b.x+anchuraLadrillo && y > b.y && y
< b.y+alturaLadrillo) {
                    dy = -dy;
                    b.estado = 0;
                    puntaje++;
                    if(puntaje ==
nroFilasLadrillos*nroColumnasLadrillos) {
                        alert("GANASTE UN IPHONE 11
FELICITACIONES!!!!");
                        document.location.reload();
                    }
                }
            }
        }
    }
}

function dibujarBola() {
    ctx.beginPath();
    ctx.arc(x, y, bolaRadio, 0, Math.PI*2);
    // SE UTILIZA EL COLOR PREVIAMENTE DEFINIDO
    ctx.fillStyle = colorBola;
    ctx.fill();
    ctx.closePath();
}

function dibujarPaleta() {
    ctx.beginPath();
    ctx.rect(paletaPosX, canvas.height-alturaPaleta, anchuraPaleta,
alturaPaleta);
    ctx.fillStyle = colorPaleta;
    ctx.fill();
    ctx.closePath();
}

```



```
function dibujarLadrillos() {
    for(var c=0; c<nroColumnasLadrillos; c++) {
        for(var f=0; f<nroFilasLadrillos; f++) {
            if(ladrillos[c][f].estado == 1) {
                var ladrilloX =
(f*(anchuraLadrillo+rellenoLadrillo))+vacioIzquierdoLadrillo;
                var ladrilloY =
(c*(alturaLadrillo+rellenoLadrillo))+vacioSuperiorLadrillo;
                ladrillos[c][f].x = ladrilloX;
                ladrillos[c][f].y = ladrilloY;
                ctx.beginPath();
                ctx.rect(ladrilloX, ladrilloY, anchuraLadrillo,
alturaLadrillo);
                ctx.fillStyle = colorLadrillo;
                ctx.fill();
                ctx.closePath();
            }
        }
    }
}

function dibujarPuntaje() {
    ctx.font = "20px time new roman";
    ctx.fillStyle = colorTexto;
    ctx.fillText("puntaje: "+puntaje, 8, 20);
}

function dibujarVidas() {
    ctx.font = "20px time new roman";
    ctx.fillStyle = colorTexto;
    // SE MUESTRA EL NÚMERO DE VIDAS DISPONIBLES
    ctx.fillText("vidas: "+vidas, canvas.width-65, 20);
}

function dibujar() {
    ctx.clearRect(0, 0, canvas.width, canvas.height);
    dibujarLadrillos();
    dibujarBola();
    dibujarPaleta();
    dibujarPuntaje();
    dibujarVidas();
    detectarColision();

    if(x + dx > canvas.width-bolaRadio || x + dx < bolaRadio) {
        dx = -dx;
    }
    if(y + dy < bolaRadio) {
        dy = -dy;
    }
    else if(y + dy > canvas.height-bolaRadio) {
```

```

        if(x > paletaPosX && x < paletaPosX + anchuraPaleta) {
            dy = -dy;
        }
        else {
            // 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
            vidas--;
            if(!vidas) {
                // SI vidas == 0 (lo cual también puede escribir:
!vidas)
                // EL JUGADOR HA PERDIDO
                alert("quieres jugar una vez mas? paga:10$");
                document.location.reload();
            }
            else {
                // SI vidas > 0 (diferente de CERO) EL JUEGO
CONTINUA
                x = canvas.width/2;
                y = canvas.height-30;
                dx = 3;
                dy = -3;
                paletaPosX = (canvas.width-anchuraPaleta)/2;
            }
        }
    }

    if(flechaDerechaPresionada && paletaPosX < canvas.width-
anchuraPaleta) {
        paletaPosX += 8;
    }
    else if(flechaIzquierdaPresionada && paletaPosX > 0) {
        paletaPosX -= 8;
    }

    x += dx;
    y += dy;

    // ESTE ES UN SEGUNDO MÉTODO PARA REALIZAR LA ANIMACIÓN DEL JUEGO
    // LA INSTRUCCIÓN: requestAnimationFrame SE EJECUTA 60 VECES POR
SEGUNDO
    // Y AL EJECUTARSE LLAMA A LA FUNCIÓN ENTRE PARÉNTESIS
    // POR TANTO, dibujar SE EJECUTA 60 VECES POR SEGUNDO
    // GENERANDO EL CICLO DEL JUEGO
    requestAnimationFrame(dibujar);
}

dibujar();
</script>

```



```
</body>  
</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.



## 12 CONCLUSIONES

---

Este juego realizado a través de un código html asignado a JavaScript, utilizando las herramientas y conocimientos aprendidos creamos y probamos códigos funciones, problemas que surgir y llegar a una solución todo esto con el fin de poder crear un juego en la red virtual

Este es un juego que nos ayuda para el aprendizaje dentro del campo de la programación tanto con el lenguaje html como con tantos lenguajes que existen el día de hoy en el campo de la programación.



## 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)