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

UTP | Pereira

**Juan diego velez sanchez**

septiembre DE 2020

2020

# CONTENIDO

[1 CONTENIDO 1](#_Toc42134755)

[2 PRESENTACIÓN 2](#_Toc42134756)

[3 FASE 1: Dibujar y mover una bola 3](#_Toc42134757)

[4 FASE 2: Rebotando en las paredes](#_Toc42134758) 7

[5 FASE 3: Control de la paleta con el teclado 10](#_Toc42134759)

[6 FASE 4: Fin de partes esenciales del juego 14](#_Toc42134760)

[7 FASE 5: Muro de ladrillos 18](#_Toc42134761)

[8 FASE 6: Detección de colisiones 23](file:///C:\Users\DIANA\Desktop\UTP%20Juan%20Diego\Introducción%20a%20la%20informatica\Monografia\MONOGRAFÍA%20Juego%202D%20en%20JavaScript%20puro%20-%20Informática.docx#_Toc42134762)

[9 FASE 7: Contar puntos, ganar, golpe y degradado de ladrillos 28](file:///C:\Users\DIANA\Desktop\UTP%20Juan%20Diego\Introducción%20a%20la%20informatica\Monografia\MONOGRAFÍA%20Juego%202D%20en%20JavaScript%20puro%20-%20Informática.docx#_Toc42134763)

[10 FASE 8: Controlando el ratón 36](file:///C:\Users\DIANA\Desktop\UTP%20Juan%20Diego\Introducción%20a%20la%20informatica\Monografia\MONOGRAFÍA%20Juego%202D%20en%20JavaScript%20puro%20-%20Informática.docx#_Toc42134764)

[11 FASE 9: Finalizando el juego 42](#_Toc42134765)

[12 FASE 10: Moviento de ladrillos, misil y explosión](#_FASE_10:_Movimiento) 49

[13 FASE 11: Fin, menú, ganar y perder](#_FASE_11:_Fin,) 57

[14 CONCLUSIONES 71](#_CONCLUSIONES)

[15 BIBLIOGRAFÍA 72](#_BIBLIOGRAFÍA)

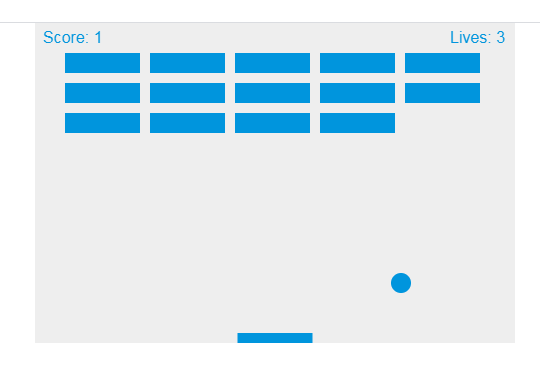
# PRESENTACIÓN

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: Juan Diego Vélez Sánchez**

# FASE 1: Dibujar y mover una bola

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

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

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

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

El código es el siguiente:

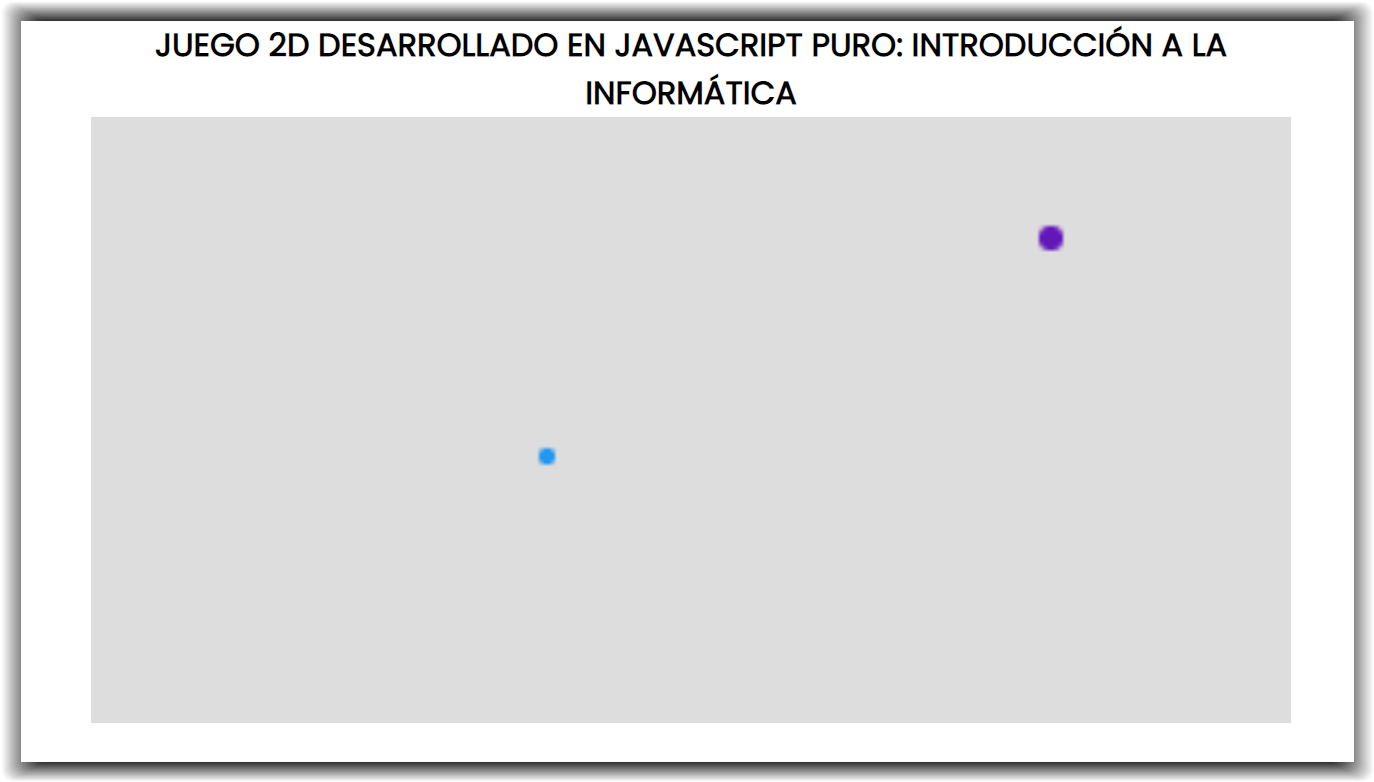
1. <!DOCTYPE html>
2. <html>
3. <head>
4. <meta charset=*"UTF-8"* />
5. <title>Juego 2D</title>
6. *<!-- Codigo Css para darle una mejor apariencia a la pagina*
7. *Define los estilos de la interfaz visual*
8. *padding es la distancia de un objeto en relación con el marco que lo contiene*
9. *margin es la distancia que separa a un objeto de otro*
10. *background es el color de fondo*
11. *display: block; Estos elementos fluyen hacia abajo*
12. *margin: 0 auto; Centra el canvas en la pantalla -->*
13. <style>
14. */\* Importa una fuente(tipo de letra) \*/*
15. @import url(*"https://fonts.googleapis.com/css2?family=Poppins&display=swap"*);
16. \* {
17. font-family: *"Poppins"*, sans-serif;
18. padding: 0;
19. margin: 0;
20. }
21. body {
22. margin: 0 200px 0 200px;
23. }
24. header h1 {
25. font-size: 2em;
26. text-transform: uppercase;
27. text-align: center;
28. }
29. #miCanvas {
30. width: 100%;
31. height: 80vh;
32. background-color: #dddddd;
33. }
34. </style>
35. </head>
36. *<!-- Comenzamos nuestra pagina -->*
37. <body>
38. *<!-- Intrducimos un titulo -->*
39. <header>
40. <h1>
41. Juego 2D desarrollado en JavaScript puro: Introducción a la Informática
42. </h1>
43. </header>
44. *<!-- Introducimos nuestro canvas con un identificador -->*
45. <canvas id=*"miCanvas"*> </canvas>
46. *<!-- Introducimos nuestro codigo JavaScript -->*
47. <script>
48. */\* Llamamos al canvas y le decimos que queremos dibujar en 2d \*/*
49. var canvas = document.getElementById(*"miCanvas"*);
50. var ctx = canvas.getContext(*"2d"*);
51. *//Variables para la bola*
52. var ballRadius = 3; *//Tamaño de la bola*
53. var x = canvas.width / 2; *// Coloca la bola en la mitad del ancho deL CANVAS*
54. var y = canvas.height - 30; *// Coloca la bola en la parte inferior del CANVAS*
55. */\* DEFINE LOS INCREMENTOS EN X y en Y. El valor dy es negativo*
56. *para que inicialmente el movimiento de la bola sea hacia arriba \*/*
57. var dx = 2;
58. var dy = -2;
59. *//Bola 2 el valor dx es negativo para que inicalmente se mueva a la izquierda*
60. var x2 = x;
61. var y2 = y;
62. var dx2 = -0.8;
63. var dy2 = -0.8;
64. function dibujarBola(cordernaX, cordenaY, tamaño, color) {
65. *// Inicia el dibujo*
66. ctx.beginPath();
67. ctx.arc(cordernaX, cordenaY, tamaño, 0, Math.PI \* 2);
68. *// Color de llenado*
69. ctx.fillStyle = color;
70. *// Se rellena el círculo con el color indicado*
71. ctx.fill();
72. *// Finaliza el dibujo*
73. ctx.closePath();
74. }
75. var azar = Math.random() \* 100; *// Nos dara un numero al azar entre 0 y 99*
76. function dibujar() {
77. *// Limpia el CANVAS*
78. ctx.clearRect(0, 0, canvas.width, canvas.height);
79. *// Dibuja la bola.*
80. dibujarBola(x, y, ballRadius, *"#6317B9"*);
81. */\* Si la variable azar que almacena un numero aleatorio*
82. *es mayor a 50 entonces dibujara una segunda bola,*
83. *no sabemos si es la primera vez que cargamos la pagina o n veces \*/*
84. if (azar > 50) {
85. dibujarBola(x2, y2, ballRadius - 1, *"#2196f3"*);
86. }
87. *// Se incrementa los valores para poder hacer el movimiento de las bolas*
88. x += dx;
89. y += dy;
90. x2 += dx2;
91. y2 += dy2;
92. }
93. */\* EJECUTA LA FUNCIÓN dibujar CADA 10 MILISEGUNDOS*
94. *Este es el mecanismo utilizado para construir un sistema que*
95. *ejecuta acciones de manera permanente y periódica \*/*
96. setInterval(dibujar, 20);
97. </script>
98. </body>
99. </html>

Al ejecutar este código se obtiene la siguiente interfaz visual:



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

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



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

# FASE 2: Rebotando en las paredes

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

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

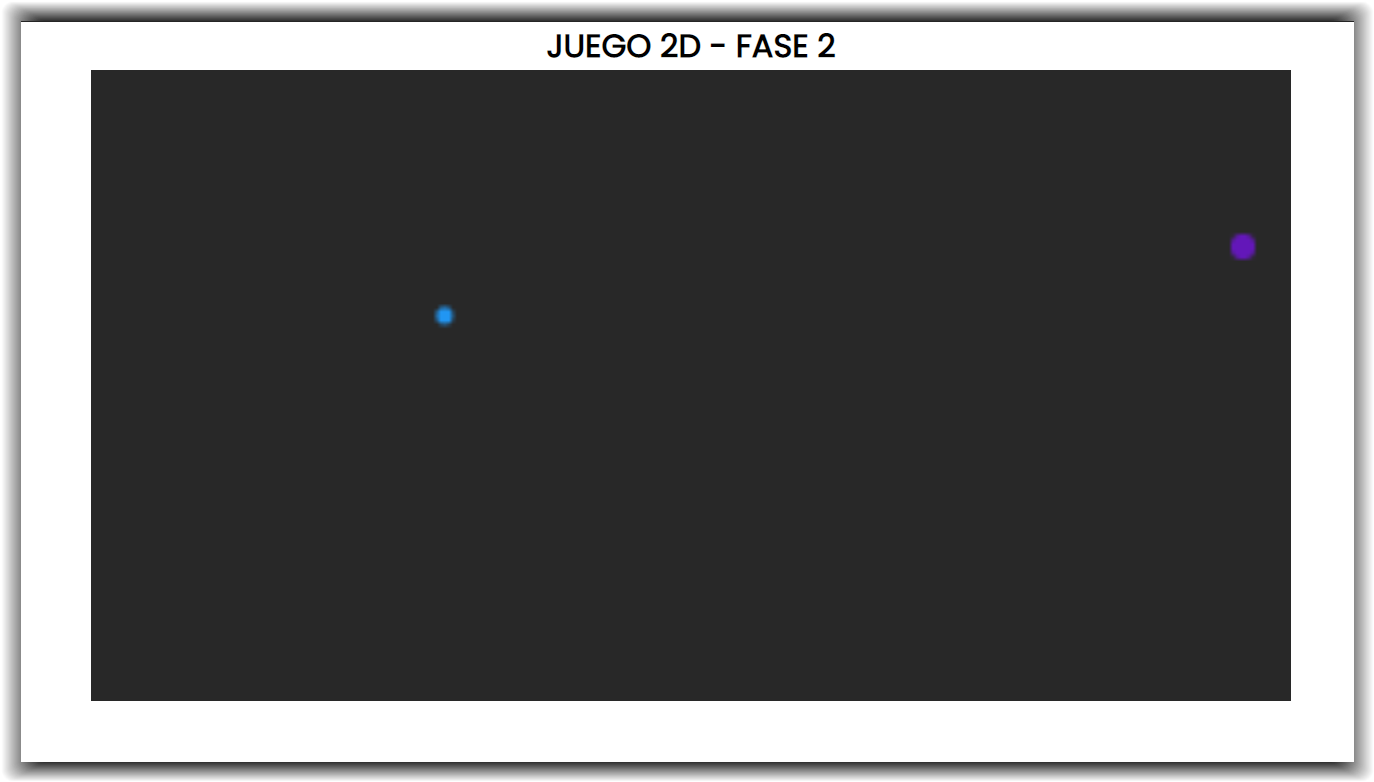
La primera condición es if(x + dx > canvas.width-ballRadius || x + dx < ballRadius) { dx = -dx;}, esta condición crea el rango horizontal al que la pelota se puede desplazar. Y el valor de dx pasara a valer negativo para poder que el movimiento se invierta, o sea, si se movía a la derecha, cuando llegue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

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

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

1. <!DOCTYPE html>
2. <html>
3. <head>
4. <meta charset=*"UTF-8"* />
5. <title>Juego 2D</title>
6. <style>
7. @import url(*"https://fonts.googleapis.com/css2?family=Poppins&display=swap"*);
8. \* {
9. font-family: *"Poppins"*, sans-serif;
10. padding: 0;
11. margin: 0;
12. }
13. body {
14. margin: 0 200px 0 200px;
15. }
16. h1 {
17. font-size: 2em;
18. text-transform: uppercase;
19. text-align: center;
20. }
21. canvas {
22. width: 100%;
23. height: 80vh;
24. background-color: #dddddd;
25. transition: 0.5s; */\* El retardo para cambiar de un estado a otro \*/*
26. }
27. */\* Se crea un nuevo estilo el cual sera activado cuando la bola principal toca el limite en la parte superior del canvas \*/*
28. canvas.activado{
29. background-color: #222222;
30. display: block;
31. margin: 0 auto;
32. }
33. </style>
34. </head>
35. <body>
36. <h1>Juego 2D - Fase 2</h1>
37. <canvas id=*"miCanvas"*> </canvas>
38. <script>
39. var canvas = document.getElementById(*"miCanvas"*);
40. var ctx = canvas.getContext(*"2d"*);
41. var ballRadius = 3;
42. var x = canvas.width / 2;
43. var y = canvas.height - 30;
44. var dx = 2;
45. var dy = -2;
46. var x2 = x;
47. var y2 = y;
48. var dx2 = -0.8;
49. var dy2 = -0.8;
50. function dibujarBola(cordernaX, cordenaY, tamaño, color) {
51. ctx.beginPath();
52. ctx.arc(cordernaX, cordenaY, tamaño, 0, Math.PI \* 2);
53. ctx.fillStyle = color;
54. ctx.fill();
55. ctx.closePath();
56. }
57. var azar = Math.random() \* 100;
58. function dibujar() {
59. ctx.clearRect(0, 0, canvas.width, canvas.height);
60. dibujarBola(x, y, ballRadius, *"#6317B9"*);
61. if (azar > 50) {
62. dibujarBola(x2, y2, ballRadius - 1, *"#2196f3"*);
63. }
64. *//Las condiciones nos permitira saber si la bola a llegado al limete de x o y, si es asi entonces cambiara la dirección, se logra cambiando el signo de las variable dx y dy, tenemos posiblemente dos bolas las cuales el movimento de una de ellas es mas lento, por lo cual no golpeara a la misma vez entonces utilizamos diferentes condiciones.*
65. *//CONDIONES PARA LA PRIMERA BOLA*
66. *//Cuando el valor de x + dx es mayor al ancho del canvas o menor al radio de la bola entonces se cambia la dirección horizontal*
67. if(x + dx > canvas.width-ballRadius || x + dx < ballRadius){
68. dx = -dx;
69. }
70. *//Cuando el valor de y + dy es mayor a la altura del canvas o menor al radio de la bola entonces se cambia la dirección vertical.*
71. if(y + dy < ballRadius){
72. dy = -dy;
73. *//Si llega al limite en Y entonces rebotara y habra un evento, el cual consiste que al elemento CANVAS en HTML le agregue o le quite una clase(identificador) la cual permitira en CSS cambiar el color del CANVAS*
74. var cambia = document.querySelector(*'canvas'*)
75. cambia.classList.toggle(*'activado'*)
76. }
77. *//CONDICONES PARA LA SEGUNDA BOLA similares a la bola 1 solo que diferentes coordenas X y Y*
78. if(x2 + dx2 > canvas.width-ballRadius || x2 + dx2 < ballRadius){
79. dx2 = -dx2;
80. }
81. if(y2 + dy2 < ballRadius){
82. dy2 = -dy2;
83. }
84. x += dx;
85. y += dy;
86. x2 += dx2;
87. y2 += dy2;
88. }
89. setInterval(dibujar, 20);
90. </script>
91. </body>
92. </html>

El resultado será el siguiente:



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

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

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

# FASE 3: Control de la paleta con el teclado

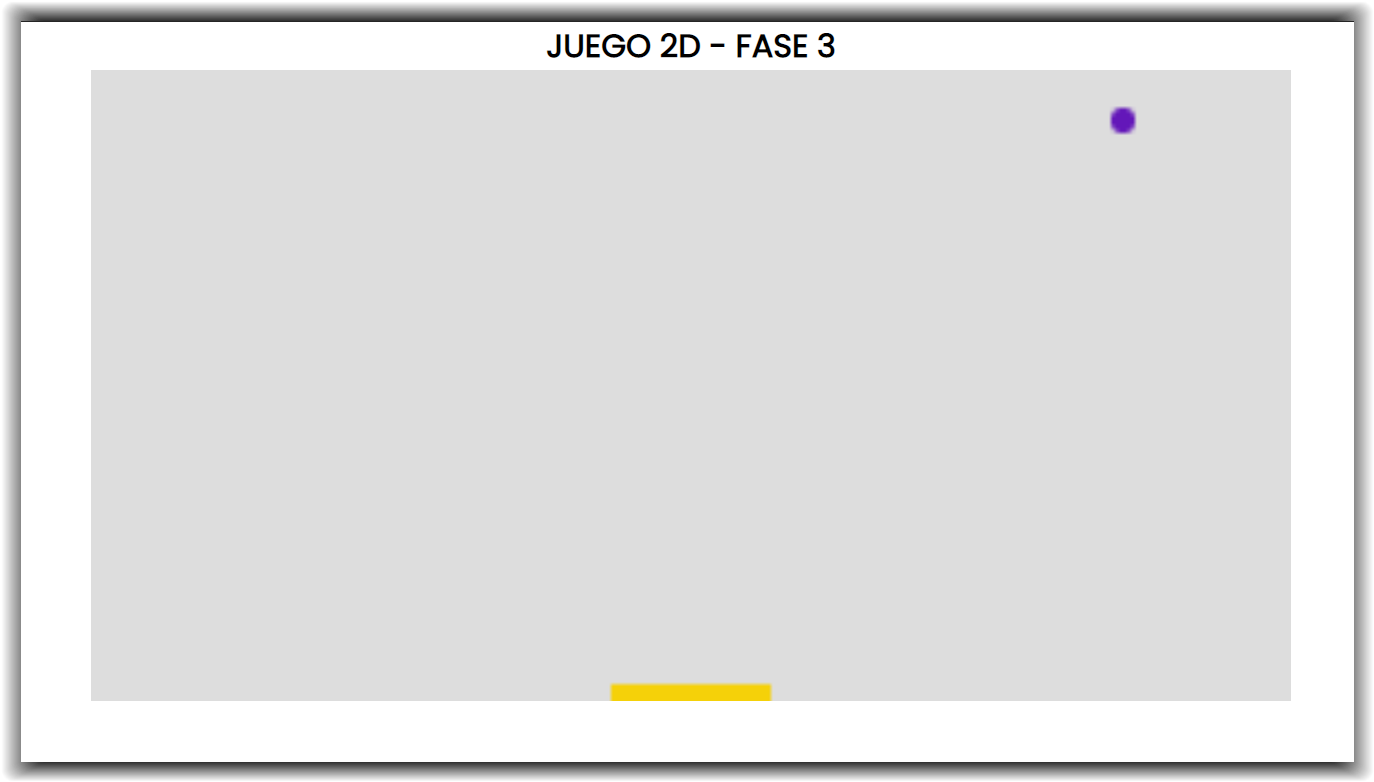
El paso número 3, consiste en crear una paleta, esta estará sujeta a unas variables la cuales serán utilizadas para su características, como el taño la anchura y el posicionamiento. Para la paleta tenemos dos variables que almacenan un dato booleano (true, false), esto nos permitirá saber si el jugador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

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

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

1. <!DOCTYPE html>
2. <html>
3. <head>
4. <meta charset=*"UTF-8"* />
5. <title>Juego 2D</title>
6. <style>
7. @import url(*"https://fonts.googleapis.com/css2?family=Poppins&display=swap"*);
8. \* {
9. font-family: *"Poppins"*, sans-serif;
10. padding: 0;
11. margin: 0;
12. }
13. body {
14. margin: 0 200px 0 200px;
15. }
16. h1 {
17. font-size: 2em;
18. text-transform: uppercase;
19. text-align: center;
20. }
21. canvas {
22. width: 100%;
23. height: 80vh;
24. background-color: #dddddd;
25. transition: 0.5s;
26. }
27. canvas.activado{
28. background-color: #222222;
29. display: block;
30. margin: 0 auto;
31. }
32. </style>
33. </head>
34. <body>
35. <h1>Juego 2D - Fase 2</h1>
36. <canvas id=*"miCanvas"*> </canvas>
37. <script>
38. var canvas = document.getElementById(*"miCanvas"*);
39. var ctx = canvas.getContext(*"2d"*);
40. var ballRadius = 3;
41. var x = canvas.width / 2; *//x: columna en la que se encuentra situada la bola*
42. var y = canvas.height - 30; *//y: fila en la que se encuentra situada la bola*
43. var dx = 2; *//dx: desplazamiento horizontal de la bola*
44. var dy = -2;*//dy: desplazamiento vertical de la bola*
45. var x2 = x;
46. var y2 = y;
47. var dx2 = -0.8;
48. var dy2 = -0.8;
49. */\* Las siguiente variables corresponden a las caracteristicas de la paleta \*/*
50. var alturaPaleta = 4; *// define la altura de lapaleta en pixeles*
51. var anchuraPaleta = 40; *// define El ancho de la paleta*
52. *//Posicion donde se encontrara inicialmente nuestra paleta en este caso en el centro del canvas y en la base de la pantalla, Para calcular la posición en X de la paleta, se debe tomar el ancho del CANVAS, restarle la anchura de la paleta, y el espacio que sobre debe dividirse entre dosEsto garantiza que originalmente la paleta estará centrada en la base de la pantalla*
53. var paletaPostX = (canvas.width-anchuraPaleta) / 2;
54. */\* Se define dos variables con tipos booleanos, para saber que tecla se preciona y permitir el movimiento, estas comienzan con el valor de false ya que al inicar el juego no se ha precionado ninguna tecla, cuando se pulse cualquiera de las dos teclas el estado correspondiente a la tecla cambiara a true, y permitira mover la paleta en su respectiva dirección   \*/*
55. var flechaDerechaPulsada = false; *// corresponde a la flecha derecha si es pulsada cambiara a true*
56. var flechaIzquierdaPulsada = false; *// corresponde a la flecha izquierda si es pulsada cambiara a true*
57. */\* La instruccion addEvenetListener, es un mecanismo para dar respuesta a los eventos que se producen en el juego, por ejemplo al precionar una de las dos teclas queremos que se mueva.*
58. *Este recibe tres parametros.*
59. *1) El evento que se va a detectar en este caso precionar una tecla*
60. *2) El nombre que le asignamos a la función que responde ante el evento*
61. *3) Valor true o false que determina la reacción ante el evento, o sea la propagación de este, cuando le damos a la tecla el evento sera asignado a toda la pagina, lo cual utilizamos el valor de false, este valor permite que el flujo del evento comience desde lo inferior hasta lo superior, o sea que al pulsar la tecla el evento se le asignara al CANVAS y luego al Body, en cambio si el valor es true el evento se le asignara al Body y luego al CANVAS.*
62. *\*/*
63. *// Evento si se pulsa una tecla*
64. document.addEventListener(*"keydown"*, teclaPresionada, false);
65. *// Evento si se suelta una tecla*
66. document.addEventListener(*"keyup"*, teclaLiberada, false);
67. *// Función que maneja la tecla presionada*
68. function teclaPresionada(e){
69. if(e.keyCode == 39){
70. */\* e: es el evento que se produce, keyCode permitir saber que codigo correspondiente a la tecla se ha presionado, 39 equivale a la tecla  flecha derecha por lo cual el valor de la variable la flechaDerechaPulsada cambia a true\*/*
71. flechaDerechaPulsada = true;
72. }
73. else if(e.keyCode == 37){
74. */\* e: es el evento que se produce, keyCode permitir saber que codigo correspondiente a la tecla se ha presionado, 37 equivale a la tecla  flecha izquierda por lo cual el valor de la variable la flechaIzquierdaPulsada cambia a true\*/*
75. flechaIzquierdaPulsada = true;
76. }
77. }
78. *// Función que maneja la tecla liberada*
79. function teclaLiberada(e){
80. if(e.keyCode == 39){
81. */\* Si la tecla liberada es la 39, se ha dejado de precionar la flecha derecha, la variable cambia a false\*/*
82. flechaDerechaPulsada = false;
83. }
84. else if(e.keyCode == 37){
85. */\* Si la tecla liberada es la 37, se ha dejado de precionar la flecha izquierda, la variable cambia a false\*/*
86. flechaIzquierdaPulsada = false;
87. }
88. }
90. function dibujarBola(cordernaX, cordenaY, tamaño, color) {
91. ctx.beginPath();
92. ctx.arc(cordernaX, cordenaY, tamaño, 0, Math.PI \* 2);
93. ctx.fillStyle = color;
94. ctx.fill();
95. ctx.closePath();
96. }
97. *//Función para dibujar la paleta*
98. function dibujarPaleta(){
99. ctx.beginPath(); *// Se incia el dibujo*
100. *//Se crea un rectangulo utlizando los valores de las variable correspondientes a la paleta, utilizamos la posición de x, y Y menos la altura de la paleta parasituarla en la base del CANVAS ,la altura, anchura*
101. ctx.rect(paletaPostX, canvas.height-alturaPaleta, anchuraPaleta, alturaPaleta);
102. ctx.fillStyle = *"#f5d109"* *// Color de ala paleta*
103. ctx.fill(); *// Se rellena la paleta*
104. ctx.closePath(); *// Finaliza el dibujo*
105. }
106. var azar = Math.random() \* 100;
107. *// Función principal para generar los procesos*
108. function dibujar() {
109. ctx.clearRect(0, 0, canvas.width, canvas.height);
110. dibujarBola(x, y, ballRadius, *"#6317B9"*); *// dibuja la bola*
111. dibujarPaleta(); *// Dibuja la paleta*
112. *// Dibuja una segunda bola*
113. if (azar > 50) {
114. dibujarBola(x2, y2, ballRadius - 1, *"#2196f3"*);
115. }
116. *// Rebote de la bola*
117. if(x + dx > canvas.width-ballRadius || x + dx < ballRadius){
118. dx = -dx;
119. }
121. if(y + dy < ballRadius){
122. dy = -dy;
123. var cambia = document.querySelector(*'canvas'*)
124. cambia.classList.toggle(*'activado'*)
125. }
127. if(x2 + dx2 > canvas.width-ballRadius || x2 + dx2 < ballRadius){
128. dx2 = -dx2;
129. }
130. if(y2 + dy2 < ballRadius){
131. dy2 = -dy2;
132. }
133. */\* Movimiento de la paleta*
134. *si se pulsa la flecha derecha y aun no ha llegado al borde de la anchura del CANVAS se puede carbiar su posición, la función dibujarPaleta() redibuja la paleta de manera ciclica en la nueva posición\*/*
135. if(flechaDerechaPulsada && paletaPostX < canvas.width-anchuraPaleta){
136. *// Desplaza la paleta hacia la derecha se logra sumando la poscion actual con la nueva*
137. paletaPostX += 7;
138. }
139. else if(flechaIzquierdaPulsada && paletaPostX > 0){
140. *// Desplaza la paleta hacia la izquierda se logra restando la poscion actual con la nueva*
141. paletaPostX -= 7;
142. }
143. *// Se incrementan los valores de la pocion de las bolas*
144. x += dx;
145. y += dy;
146. x2 += dx2;
147. y2 += dy2;
148. }
149. *// Redibuja, actualiza las posiciones en el juego cada 25 milisegundos,*
150. setInterval(dibujar, 25);
151. </script>
152. </body>
153. </html>

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



**Gráfica 5. Interfaz con la correspondiente paleta dibujada.**

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

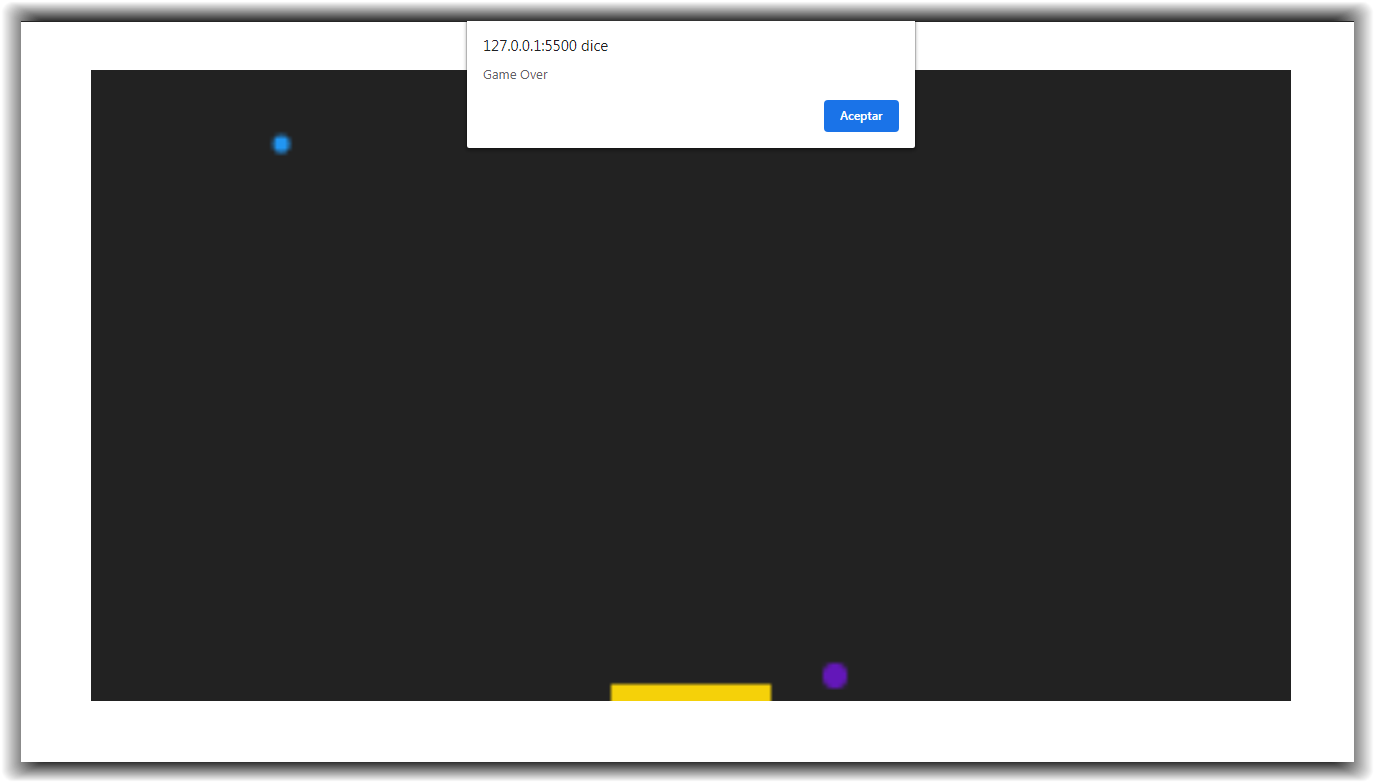
# FASE 4: Fin de partes esenciales del juego

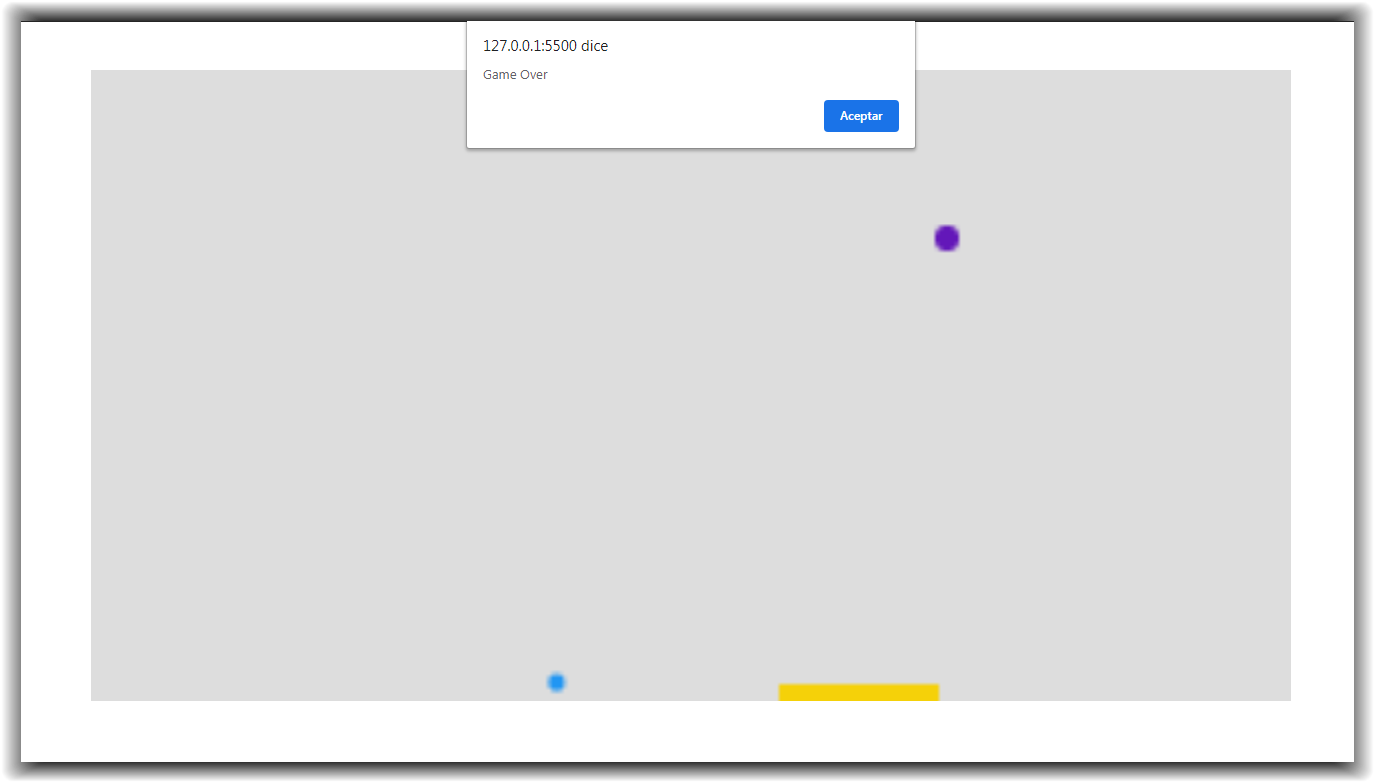
En esta fase del programa, vamos a programar el rebote de las bolas en la paleta ya que si recordamos nos puede salir una o dos bolas esto al azar, lo cual debemos evaluar cuando estas lleguen a la base del CANVAS si está en posición de la paleta o si toca directamente la base.

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

1. <!DOCTYPE html>
2. <html>
3. <head>
4. <meta charset=*"UTF-8"* />
5. <title>Juego 2D</title>
6. <style>
7. @import url(*"https://fonts.googleapis.com/css2?family=Poppins&display=swap"*);
8. \* { font-family: *"Poppins"*, sans-serif; padding: 0; margin: 0;}
9. body {margin: 0 200px 0 200px;}
10. h1 {font-size: 2em; text-transform: uppercase; text-align: center;}
11. canvas {width: 100%; height: 80vh; background-color: #dddddd; transition: 0.5s;}
12. canvas.activado {background-color: #222222; display: block; margin: 0 auto;}
13. </style>
14. </head>
15. <body>
16. <h1>Juego 2D - Fase 4</h1>
17. <canvas id=*"miCanvas"*> </canvas>
18. <script>
19. var canvas = document.getElementById(*"miCanvas"*);
20. var ctx = canvas.getContext(*"2d"*);
21. var ballRadius = 3;
22. var x = canvas.width / 2;
23. var y = canvas.height - 30;
24. var dx = 2;
25. var dy = -2;
26. var x2 = x;
27. var y2 = y;
28. var dx2 = -0.8;
29. var dy2 = -0.8;
30. *// Paleta*
31. var alturaPaleta = 4;
32. var anchuraPaleta = 40;
33. var paletaPosX = (canvas.width - anchuraPaleta) / 2;
34. var flechaDerechaPulsada = false;
35. var flechaIzquierdaPulsada = false;
36. document.addEventListener(*"keydown"*, teclaPresionada, false);
37. document.addEventListener(*"keyup"*, teclaLiberada, false);
38. function teclaPresionada(e) {
39. if (e.keyCode == 39) {
40. flechaDerechaPulsada = true;
41. } else if (e.keyCode == 37) {
42. flechaIzquierdaPulsada = true;
43. }
44. }
45. function teclaLiberada(e) {
46. if (e.keyCode == 39) {
47. flechaDerechaPulsada = false;
48. } else if (e.keyCode == 37) {
49. flechaIzquierdaPulsada = false;
50. }
51. }
52. function dibujarBola(cordernaX, cordenaY, tamaño, color) {
53. ctx.beginPath();
54. ctx.arc(cordernaX, cordenaY, tamaño, 0, Math.PI \* 2);
55. ctx.fillStyle = color;
56. ctx.fill();
57. ctx.closePath();
58. }
59. function dibujarPaleta() {
60. ctx.beginPath();
61. ctx.rect(paletaPosX, canvas.height-alturaPaleta, anchuraPaleta, alturaPaleta);
62. ctx.fillStyle = *"#f5d109"*;
63. ctx.fill();
64. ctx.closePath();
65. }
66. var azar = Math.random()\*100;
67. *// Función principal para generar los procesos*
68. function dibujar() {
69. ctx.clearRect(0, 0, canvas.width, canvas.height);
70. dibujarBola(x, y, ballRadius, *"#6317B9"*); *// dibuja la bola*
72. *// Dibuja una segunda bola*
73. if (azar > 50) {
74. dibujarBola(x2, y2, ballRadius - 1, *"#2196f3"*);
75. }
76. dibujarPaleta(); *// Dibuja la paleta*
77. *// Rebote de la bola*
78. if (x + dx > canvas.width-ballRadius || x + dx < ballRadius) {
79. dx = -dx;
80. }
81. if (y + dy < ballRadius) {
82. dy = -dy;
83. var cambia = document.querySelector(*"canvas"*);
84. cambia.classList.toggle(*"activado"*);
85. }
86. if (x2 + dx2 > canvas.width-ballRadius || x2 + dx2 < ballRadius) {
87. dx2 = -dx2;
88. }
89. if (y2 + dy2 < ballRadius) {
90. dy2 = -dy2;
91. }
92. *// Si la bola alcanza la base del CANVAS, se debe evaluar si la posiocn de la bola está dentro de la paleta, si esta entonces se incrementa y si no es asi, cuando la bola llegue a la base de la pantalla se detinene el ciclo del juego y se da a entender por medio de un mensaje con la funcionn alert que el juego ha terminado y posteriormente se recarga la pagina. Se definen dos veces estas condiciones ya que nos puede salir dos bolas y estas no tocan la paleta a la misma vez.*
93. *// Rebotar en Paleta o Game Over*
94. if (y + dy > canvas.height-ballRadius) {
95. if (x > paletaPosX && x < paletaPosX + anchuraPaleta) {
96. dy = -dy;
97. } else {
98. clearInterval(juego);
99. alert(*"Game Over"*);
100. document.location.reload();
101. }
102. }
103. *// Por si hay una segunda bola entonces esta también debe rebotar en la paleta o si no perdera*
104. if (azar > 50) {
105. if (y2 + dy2 > canvas.height - ballRadius) {
106. if (x2 > paletaPosX && x2 < paletaPosX + anchuraPaleta) {
107. dy2 = -dy2;
108. } else {
109. clearInterval(juego);
110. alert(*"Game Over"*);
111. document.location.reload();
112. }
113. }
114. }
115. *// Movimiento Paleta*
116. if (flechaDerechaPulsada && paletaPosX < canvas.width - anchuraPaleta) {
117. paletaPosX += 7;
118. } else if (flechaIzquierdaPulsada && paletaPosX > 0) {
119. paletaPosX -= 7;
120. }
121. *// Se incrementan los valores de la pocion de las bolas*
122. x += dx;
123. y += dy;
124. x2 += dx2;
125. y2 += dy2;
126. }
127. */\* Se asigna una variable al proceso del ciclo, si se quiere eliminar el ciclo se utiliza esta variable \*/*
128. var juego = setInterval(dibujar, 25);
129. </script>
130. </body>
131. </html>

El resultado del anterior código es:

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



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

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

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

# FASE 5: Muro de ladrillos

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

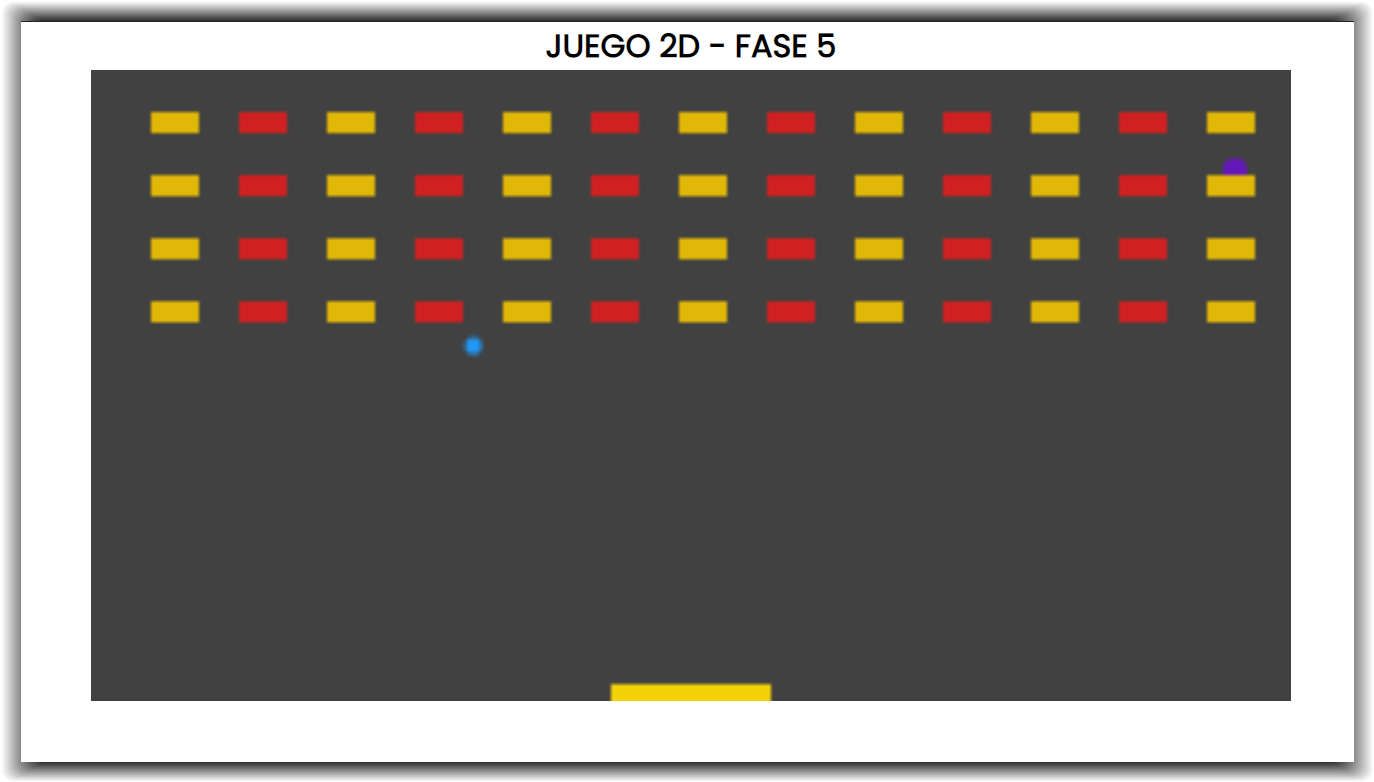
Las variables a utilizar para las características de estos son: el ancho de cada ladrillo, altura, espacio entre ellos, número de filas y columnas, estas dos últimas variables no serán constantes cada vez que se carga la página, ya que gracias al método Math.random(), anteriormente utilizado para dibujar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

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

1. <!DOCTYPE html>
2. <html>
3. <head>
4. <meta charset=*"UTF-8"* />
5. <title>Juego 2D</title>
6. <style>
7. @import url(*"https://fonts.googleapis.com/css2?family=Poppins&display=swap"*);
8. \* {font-family: *"Poppins"*, sans-serif; padding: 0; margin: 0; }
9. body {margin: 0 200px 0 200px;}
10. h1 {font-size: 2em; text-transform: uppercase; text-align: center;}
11. canvas {width: 100%; height: 80vh; background-color: #dddddd;transition: 0.5s;}
12. canvas.activado {background-color: #222222; display: block;margin: 0 auto;}
13. </style>
14. </head>
15. <body>
16. <h1>Juego 2D - Fase 5</h1>
17. <canvas id=*"miCanvas"*> </canvas>
18. <script>
19. var canvas = document.getElementById(*"miCanvas"*);
20. var ctx = canvas.getContext(*"2d"*);
21. var ballRadius = 3;
22. var x = canvas.width / 2;
23. var y = canvas.height - 30;
24. var dx = 2;
25. var dy = -2;
26. var x2 = x;
27. var y2 = y;
28. var dx2 = -0.8;
29. var dy2 = -0.8;
30. *// Paleta*
31. var alturaPaleta = 4;
32. var anchuraPaleta = 40;
33. var paletaPosX = (canvas.width - anchuraPaleta) / 2;
34. var flechaDerechaPulsada = false;
35. var flechaIzquierdaPulsada = false;
36. *// VARIABLES ASOCIADAS A LOS LADRILLOS*
37. var numFilasLadrillos = 0;
38. var numColumnasLadrillos = 0;
39. *// Se crea un bucle para poner un numero al zar de ladriilos y columnas*
40. do {
41. numFilasLadrillos = Math.round(Math.random() \* 100);
42. numColumnasLadrillos = Math.round(Math.random() \* 10);
43. } while (numFilasLadrillos < 1 || numFilasLadrillos > 19 ||numColumnasLadrillos < 1 || numColumnasLadrillos > 6);
44. var anchoLadrillo = 12;
45. var alturaLadrillo = 5;
46. var rellenoLadrillos = 2;
47. var vacioSuperiorLadrillos = 10;
48. var vacioIzquierdoLadrillos = 15;
49. *// Crea un conjunto de ladrillos, Inicalmente vacio*
50. var ladrillos = [];
51. *// Se recorren el numero de columnas*
52. for (var columna = 0; columna < numColumnasLadrillos; columna++) {
53. *// Se define la primera columna. es una lista vertical*
54. ladrillos[columna] = [];
55. *// Cada columna, recorre el numero de filas*
56. for (var fila = 0; fila < numFilasLadrillos; fila++) {
57. *// para cada columna y fila se define un ladrillo*
58. *// A cada ladrillo se le asignan la posicon de la cordena X y Y estos valores valen 0 esto cambia cuando se dibujan en la funcion dibujarladrillos()*
59. ladrillos[columna][fila] = { *x*: 0, *y*: 0 };
60. }
61. }
62. document.addEventListener(*"keydown"*, teclaPresionada, false);
63. document.addEventListener(*"keyup"*, teclaLiberada, false);
64. function teclaPresionada(e) {
65. if (e.keyCode == 39) {
66. flechaDerechaPulsada = true;
67. } else if (e.keyCode == 37) {
68. flechaIzquierdaPulsada = true;
69. }
70. }
71. function teclaLiberada(e) {
72. if (e.keyCode == 39) {
73. flechaDerechaPulsada = false;
74. } else if (e.keyCode == 37) {
75. flechaIzquierdaPulsada = false;
76. }
77. }
78. function dibujarBola(cordernaX, cordenaY, tamaño, color) {
79. ctx.beginPath();
80. ctx.arc(cordernaX, cordenaY, tamaño, 0, Math.PI \* 2);
81. ctx.fillStyle = color;
82. ctx.fill();
83. ctx.closePath();
84. }
85. function dibujarPaleta() {
86. ctx.beginPath();
87. ctx.rect(
88. paletaPosX,
89. canvas.height - alturaPaleta,
90. anchuraPaleta,
91. alturaPaleta
92. );
93. ctx.fillStyle = *"#f5d109"*;
94. ctx.fill();
95. ctx.closePath();
96. }
97. var colorLadrillos = false; *// La utlizaremos para cambiar el color de los ladrillos*
98. *// DIBUJAR LADRILLOS*
99. function dibujarLadrillos() {
100. *// Recorrer todas las columnas*
101. for (var columna = 0; columna < numColumnasLadrillos; columna++) {
102. *// Para cada columna, recorre sus filas*
103. for (var fila = 0; fila < numFilasLadrillos; fila++) {
104. *// Para calcular la posicion X que le corresponde a cada ladrillo, lo calculamos segun el ancho de ladrillo, el relleno, el espaci hacia a la zquierda*
105. var brickX =
106. fila \* (anchoLadrillo + rellenoLadrillos) +
107. vacioIzquierdoLadrillos;
108. *// Se repite pero para la posicion en Y*
109. var brickY =
110. columna \* (alturaLadrillo + rellenoLadrillos) +
111. vacioSuperiorLadrillos;
112. *// Se le asignan los valores anteriores a cada ladrillo*
113. ladrillos[columna][fila].x = brickX;
114. ladrillos[columna][fila].y = brickY;
115. *// Dibuja el ladrillo con los valores asociados*
116. *// Coordenada: (brickX, brickY)*
117. *// Anchura: anchoLadrillo*
118. *// Altrua: alturaLadrillo*
119. ctx.beginPath();
120. ctx.rect(brickX, brickY, anchoLadrillo, alturaLadrillo);
121. *// Dependiendo de la variable colorLadrillos si es true los ladrillos tendran un color diferente si es false, pero tambien si el numero correspondiente al darillo es par o impar cambiara de color*
122. if (colorLadrillos) {
123. if (fila % 2) {
124. ctx.fillStyle = *"#e61c1cdc"*;
125. } else {
126. ctx.fillStyle = *"#e4bb06fa"*;
127. }
128. } else {
129. if (fila % 2) {
130. ctx.fillStyle = *"#e4bb06fa"*;
131. } else {
132. ctx.fillStyle = *"#e61c1cdc"*;
133. }
134. }
135. ctx.fill();
136. ctx.closePath();
137. *// Se recore todo el ciclo se dibuja todos los ladrillos*
138. }
139. }
140. }
141. var azar = Math.random() \* 100;
142. *// Función principal para generar los procesos*
143. function dibujar() {
144. ctx.clearRect(0, 0, canvas.width, canvas.height);
145. dibujarBola(x, y, ballRadius, *"#6317B9"*); *// dibuja la bola*
146. *// Dibuja una segunda bola*
147. if (azar > 50) {
148. dibujarBola(x2, y2, ballRadius - 1, *"#2196f3"*);
149. }
150. dibujarPaleta(); *// Dibuja la paleta*
151. dibujarLadrillos(); *// Dibuja el conjunto de los ladrillos*
152. *// Rebote de la bola*
153. if (x + dx > canvas.width - ballRadius || x + dx < ballRadius) {
154. dx = -dx;
155. *// Cambia el color de los ladrillos si la bola principal rebota en x*
156. colorLadrillos = false;
157. }
158. if (y + dy < ballRadius) {
159. dy = -dy;
160. var cambia = document.querySelector(*"canvas"*);
161. cambia.classList.toggle(*"activado"*);
162. *// Cambiara el valor y color de los ladrillos si rebota la bola principal en Y*
163. colorLadrillos = true;
164. }
165. if (x2 + dx2 > canvas.width - ballRadius || x2 + dx2 < ballRadius) {
166. dx2 = -dx2;
167. }
168. if (y2 + dy2 < ballRadius) {
169. dy2 = -dy2;
170. }
171. *// Rebotar en Paleta o Game Over*
172. if (y + dy > canvas.height - ballRadius) {
173. if (x > paletaPosX && x < paletaPosX + anchuraPaleta) {
174. dy = -dy;
175. } else {
176. clearInterval(juego);
177. alert(*"Game Over"*);
178. document.location.reload();
179. }
180. }
181. if (azar > 50) {
182. if (y2 + dy2 > canvas.height - ballRadius) {
183. if (x2 > paletaPosX && x2 < paletaPosX + anchuraPaleta) {
184. dy2 = -dy2;
185. } else {
186. clearInterval(juego);
187. alert(*"Game Over"*);
188. document.location.reload();
189. }
190. }
191. }
192. *// Movimiento Paleta*
193. if (flechaDerechaPulsada && paletaPosX < canvas.width - anchuraPaleta) {
194. paletaPosX += 7;
195. } else if (flechaIzquierdaPulsada && paletaPosX > 0) {
196. paletaPosX -= 7;
197. }
198. *// Se incrementan los valores de la pocion de las bolas*
199. x += dx;
200. y += dy;
201. x2 += dx2;
202. y2 += dy2;
203. }
204. */\* Se asigna una variable al proceso del ciclo, si se quiere eliminar el ciclo se utiliza esta variable \*/*
205. var juego = setInterval(dibujar, 25);
206. </script>
207. </body>
208. </html>



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



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

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

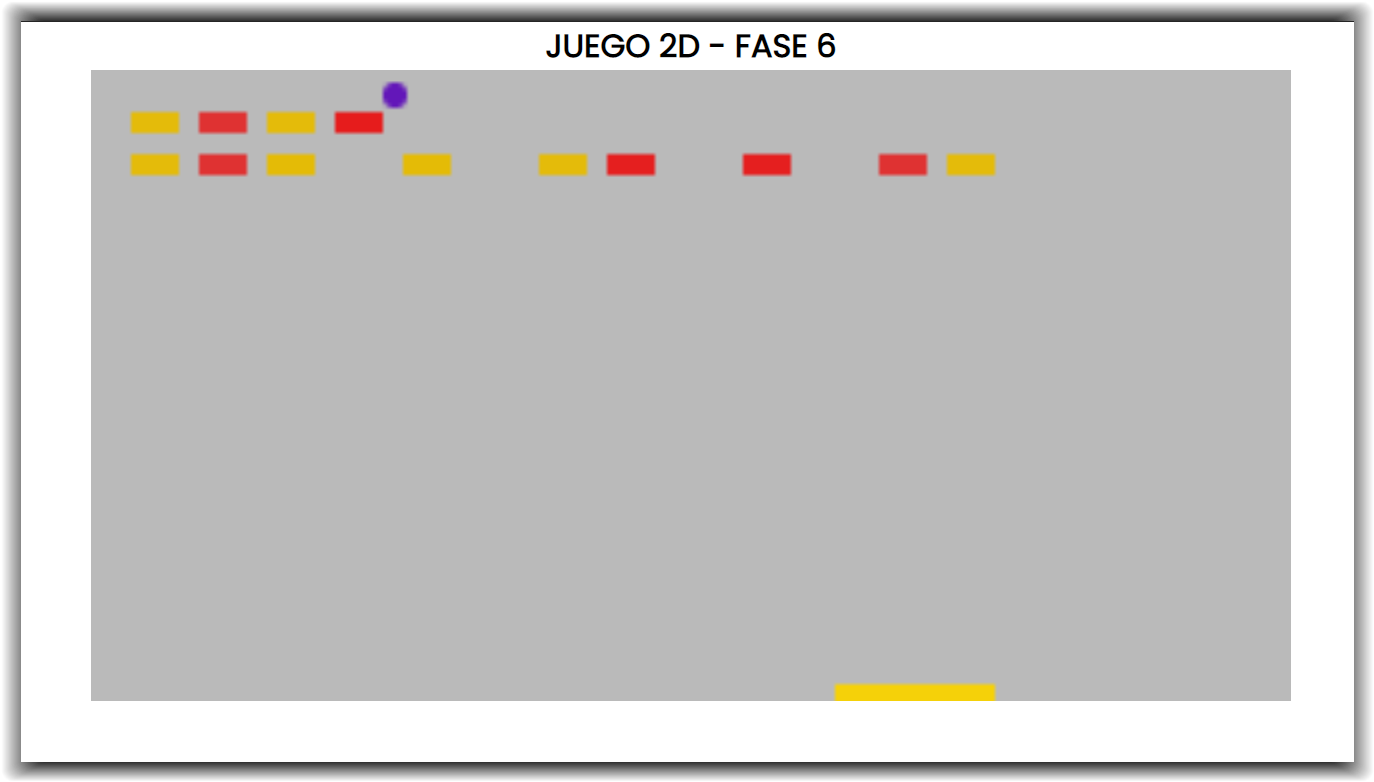
# FASE 6: Detección de colisiones

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

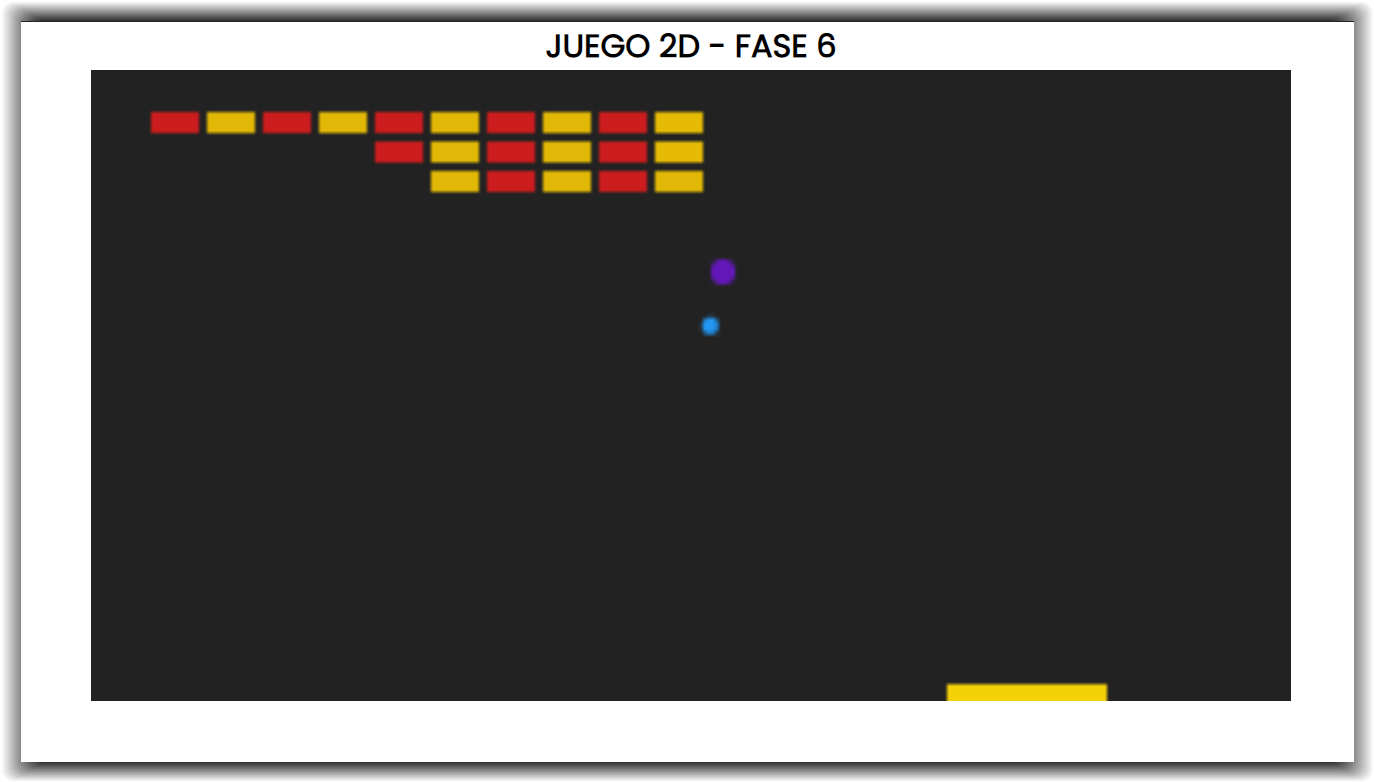
Procederemos a analizar la función que hace esto posible, la función será la siguiente: function deteccionColision(), cuando creamos los ladrillos se le asigna un status, este nos permitirá saber si el ladrillo es visible o no, en deteccionColision() recorremos las filas y columnas de los ladrillos para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

1. <!DOCTYPE html>
2. <html>
3. <head>
4. <meta charset=*"UTF-8"* />
5. <title>Juego 2D</title>
6. <style>
7. @import url(*"https://fonts.googleapis.com/css2?family=Poppins&display=swap"*);
8. \* {font-family: *"Poppins"*, sans-serif; padding: 0; margin: 0; }
9. body {margin: 0 200px 0 200px;}
10. h1 {font-size: 2em; text-transform: uppercase; text-align: center;}
11. canvas {width: 100%; height: 80vh; background-color: #dddddd;transition: 0.5s;}
12. canvas.activado {background-color: #222222; display: block;margin: 0 auto;}
13. </style>
14. </head>
15. <body>
16. <h1>Juego 2D - Fase 6</h1>
17. <canvas id=*"miCanvas"*> </canvas>
18. <script>
19. var canvas = document.getElementById(*"miCanvas"*);
20. var ctx = canvas.getContext(*"2d"*);
21. var ballRadius = 3;
22. var x = canvas.width / 2;
23. var y = canvas.height - 30;
24. var dx = 2;
25. var dy = -2;
26. var x2 = x;
27. var y2 = y;
28. var dx2 = -0.8;
29. var dy2 = -0.8;
30. *// Paleta*
31. var alturaPaleta = 4;
32. var anchuraPaleta = 40;
33. var paletaPosX = (canvas.width - anchuraPaleta) / 2;
34. var flechaDerechaPulsada = false;
35. var flechaIzquierdaPulsada = false;
36. *// VARIABLES ASOCIADAS A LOS LADRILLOS*
37. var numFilasLadrillos = 0;
38. var numColumnasLadrillos = 0;
39. *// Se crea un bucle para poner un numero al zar de ladriilos y columnas*
40. do {
41. numFilasLadrillos = Math.round(Math.random() \* 100);
42. numColumnasLadrillos = Math.round(Math.random() \* 10);
43. } while (numFilasLadrillos < 2 || numFilasLadrillos > 13 ||numColumnasLadrillos < 2 || numColumnasLadrillos > 6);
44. var anchoLadrillo = 12;
45. var alturaLadrillo = 5;
46. var rellenoLadrillos = 5;
47. var vacioSuperiorLadrillos = 10;
48. var vacioIzquierdoLadrillos = 10;
49. var ladrillos = [];
50. for (var columna = 0; columna < numColumnasLadrillos; columna++) {
51. ladrillos[columna] = [];
52. for (var fila = 0; fila < numFilasLadrillos; fila++) {
53. *// Status definira si el ladrillo es visible o no en 1 sera visible*
54. ladrillos[columna][fila] = { *x*: 0, *y*: 0, *status*: 1 };
55. }
56. }
57. document.addEventListener(*"keydown"*, teclaPresionada, false);
58. document.addEventListener(*"keyup"*, teclaLiberada, false);
59. function teclaPresionada(e) {
60. if (e.keyCode == 39) {
61. flechaDerechaPulsada = true;
62. } else if (e.keyCode == 37) {
63. flechaIzquierdaPulsada = true;
64. }
65. }
66. function teclaLiberada(e) {
67. if (e.keyCode == 39) {
68. flechaDerechaPulsada = false;
69. } else if (e.keyCode == 37) {
70. flechaIzquierdaPulsada = false;
71. }
72. }
73. function dibujarBola(cordernaX, cordenaY, tamaño, color) {
74. ctx.beginPath();
75. ctx.arc(cordernaX, cordenaY, tamaño, 0, Math.PI \* 2);
76. ctx.fillStyle = color;
77. ctx.fill();
78. ctx.closePath();
79. }
80. function dibujarPaleta() {
81. ctx.beginPath();
82. ctx.rect(paletaPosX, canvas.height - alturaPaleta, anchuraPaleta, alturaPaleta);
83. ctx.fillStyle = *"#f5d109"*;
84. ctx.fill();
85. ctx.closePath();
86. }
87. var colorLadrillos = false; *// La utlizaremos para cambiar el color de los ladrillos*
88. function dibujarLadrillos() {
89. *// Recorrer todas las columnas*
90. for (var columna = 0; columna < numColumnasLadrillos; columna++) {
91. for (var fila = 0; fila < numFilasLadrillos; fila++) {
92. var brickX = fila \* (anchoLadrillo + rellenoLadrillos) + vacioIzquierdoLadrillos;
93. var brickY = columna \* (alturaLadrillo + rellenoLadrillos) + vacioSuperiorLadrillos;
94. ladrillos[columna][fila].x = brickX;
95. ladrillos[columna][fila].y = brickY;
96. *// Se dibujan los ladrillos cuyo status vale 1*
97. if(ladrillos[columna][fila].status == 1){
98. ctx.beginPath();
99. ctx.rect(brickX, brickY, anchoLadrillo, alturaLadrillo);
100. if (colorLadrillos) {
101. if (fila % 2) {
102. ctx.fillStyle = *"#e61c1cdc"*;
103. } else {
104. ctx.fillStyle = *"#e4bb06fa"*;
105. }
106. } else {
107. if (fila % 2) {
108. ctx.fillStyle = *"#e4bb06fa"*;
109. } else {
110. ctx.fillStyle = *"#e61c1cdc"*;
111. }
112. }
113. }
114. ctx.fill();
115. ctx.closePath();
116. }
117. }
118. }
119. *// Detectar la colision*
120. function deteccionColision(){
121. *// Los ciclos recorre todos los ladrillos*
122. for(var c = 0; c < numColumnasLadrillos; c++){
123. for(var f = 0; f < numFilasLadrillos; f++){
124. *// Se define una variable para guardar el ladrillo en donde nos encontramos*
125. var b = ladrillos[c][f];
126. *// Si el ladrillo es visible, se debe verificar si entra en contacto con las bolas*
127. if(b.status == 1){
128. *// si hay contacto entre el ladrillo y las bolas entonces se modifica las cordenas y el valor de status valdra (0) para que no se muestre*
129. if(x > b.x && x < b.x + anchoLadrillo && y > b.y && y < b.y + alturaLadrillo){
130. dy = -dy;
131. b.status = 0;
132. }
133. *// Debemos tener en cuenta la segunda bola*
134. if(azarSegundaBola > 50){
135. if(x2 > b.x && x2 < b.x + anchoLadrillo && y2 > b.y && y2 < b.y + alturaLadrillo){
136. dy2 = -dy2;
137. b.status = 0;
138. }
139. }
140. }
141. }
142. }
143. }
144. var azarSegundaBola = Math.random() \* 100;
145. *// Función principal para generar los procesos*
146. function dibujar() {
147. ctx.clearRect(0, 0, canvas.width, canvas.height);
148. dibujarBola(x, y, ballRadius, *"#6317B9"*); *// dibuja la bola*
149. *// Dibuja una segunda bola*
150. if (azarSegundaBola > 50) {
151. dibujarBola(x2, y2, ballRadius - 1, *"#2196f3"*);
152. }
153. dibujarPaleta(); *// Dibuja la paleta*
154. dibujarLadrillos(); *// Dibuja el conjunto de los ladrillos*
155. *// SE LLAMA A LA FUNCION DETECTAR COLISION*
156. deteccionColision();
157. *// Rebote de la bola*
158. if (x + dx > canvas.width - ballRadius || x + dx < ballRadius) {
159. dx = -dx;
160. *// Cambia el color de los ladrillos si la bola principal rebota en x*
161. colorLadrillos = false;
162. }
163. if (y + dy < ballRadius) {
164. dy = -dy;
165. var cambia = document.querySelector(*"canvas"*);
166. cambia.classList.toggle(*"activado"*);
167. colorLadrillos = true;
168. }
169. if (x2 + dx2 > canvas.width - ballRadius || x2 + dx2 < ballRadius) {
170. dx2 = -dx2;
171. }
172. if (y2 + dy2 < ballRadius) {
173. dy2 = -dy2;
174. }
175. *// Rebotar en Paleta o Game Over*
176. if (y + dy > canvas.height - ballRadius) {
177. if (x > paletaPosX && x < paletaPosX + anchuraPaleta) {
178. dy = -dy;
179. } else {
180. clearInterval(juego);
181. alert(*"Game Over"*);
182. document.location.reload();
183. }
184. }
185. if (azarSegundaBola > 50) {
186. if (y2 + dy2 > canvas.height - ballRadius) {
187. if (x2 > paletaPosX && x2 < paletaPosX + anchuraPaleta) {
188. dy2 = -dy2;
189. } else {
190. clearInterval(juego);
191. alert(*"Game Over"*);
192. document.location.reload();
193. }
194. }
195. }
196. *// Movimiento Paleta*
197. if (flechaDerechaPulsada && paletaPosX < canvas.width - anchuraPaleta) {
198. paletaPosX += 7;
199. } else if (flechaIzquierdaPulsada && paletaPosX > 0) {
200. paletaPosX -= 7;
201. }
202. *// Se incrementan los valores de la pocion de las bolas*
203. x += dx;
204. y += dy;
205. x2 += dx2;
206. y2 += dy2;
207. }
209. var juego = setInterval(dibujar, 25);
210. </script>
211. </body>
212. </html>

El resultado tras ejecutar el anterior código es:



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

****

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

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

En los siguientes apartados se explicaran las fases finales.

# FASE 7: Contar puntos, ganar, golpe y degradado de ladrillos

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

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

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

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

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

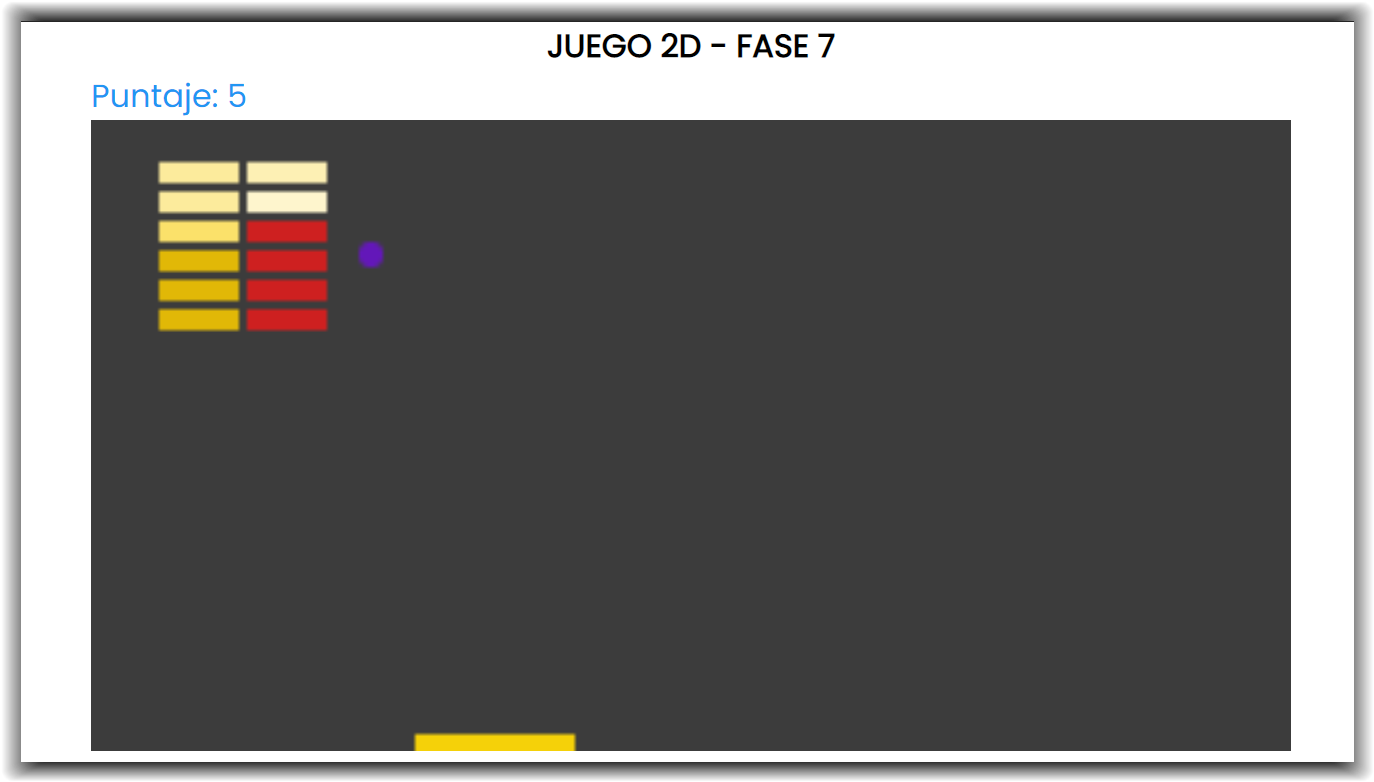
El cambio más grande se da en la función deteccionColision(), en donde creamos una función interna para disminuir el estado y el ciclo, además para sumarle uno al puntaje y ver si se ha ganado, hacemos esta función para evitar más adelante copiar el mimos código dos veces. También creamos una variable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

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

El código es el siguiente:

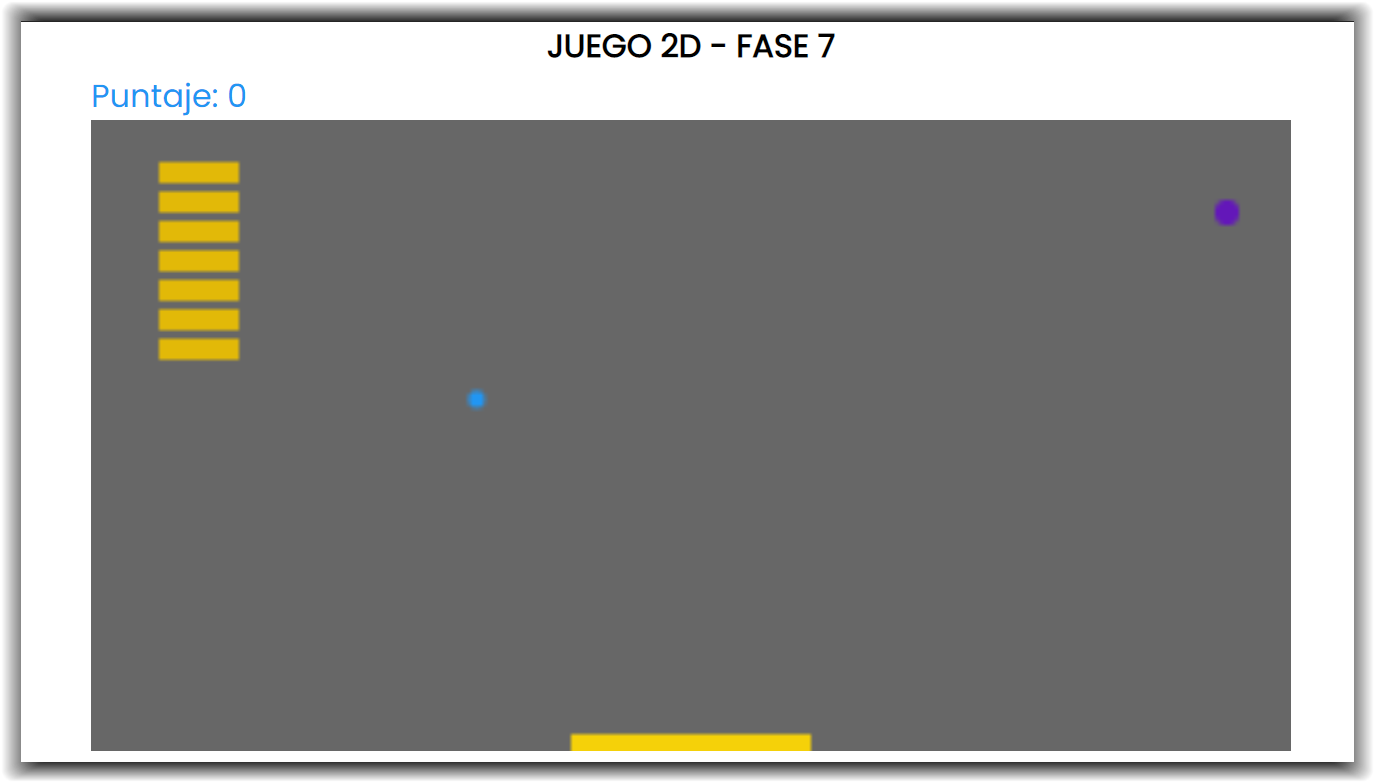
1. <!DOCTYPE html>
2. <html>
3. <head>
4. <meta charset=*"UTF-8"* />
5. <title>Juego 2D</title>
6. <style>
7. @import url(*"https://fonts.googleapis.com/css2?family=Poppins&display=swap"*);
8. \* {
9. font-family: *"Poppins"*, sans-serif;
10. padding: 0;
11. margin: 0;
12. }
13. body {
14. margin: 0 200px 0 200px;
15. }
16. h1 {
17. font-size: 2em;
18. text-transform: uppercase;
19. text-align: center;
20. }
21. canvas {
22. width: 100%;
23. height: 80vh;
24. background-color: #dddddd;
25. transition: 0.5s;
26. display: block;
27. margin: 50px auto;
28. }
29. canvas.activado {
30. background-color: #222222;
31. }
32. </style>
33. </head>
34. <body>
35. <h1>Juego 2D - Fase 7</h1>
36. <canvas id=*"miCanvas"*> </canvas>
37. <script>
38. var canvas = document.getElementById(*"miCanvas"*);
39. var ctx = canvas.getContext(*"2d"*);
41. var ballRadius = 3;
42. var x = canvas.width / 2;
43. var y = canvas.height - 30;
44. var dx = 2;
45. var dy = -2;
46. var x2 = x;
47. var y2 = y;
48. var dx2 = -0.8;
49. var dy2 = -0.8;
50. *// Variable para saber si se dibuja la segunda bola depende del numero arrojado al azar*
51. var segundaBola = false;
52. if(Math.random() \* 100 > 50){
53. segundaBola = true;
54. }
55. *// Paleta*
56. var alturaPaleta = 4;
57. var anchuraPaleta = 40;
58. *// Si el valor de la variable segundaBola es true, hara que la anchura de la paleta sea mas grande*
59. if(segundaBola){
60. anchuraPaleta = 60;
61. }
62. var paletaPosX = (canvas.width - anchuraPaleta) / 2;
63. var flechaDerechaPulsada = false;
64. var flechaIzquierdaPulsada = false;
65. *// VARIABLES ASOCIADAS A LOS LADRILLOS*
66. var numFilasLadrillos = 0;
67. var numColumnasLadrillos = 0;
68. *// Se crea un bucle para poner un numero al azar de ladriilos y columnas, esto para que nos arroje un numeros entre 1 y 12 en filas y entre 1 y 8 en columnas*
69. do {
70. numFilasLadrillos = Math.round(Math.random() \* 100);
71. numColumnasLadrillos = Math.round(Math.random() \* 10);
72. } while (numFilasLadrillos < 1 || numFilasLadrillos > 12 || numColumnasLadrillos < 1 || numColumnasLadrillos > 8);
73. var anchoLadrillo = 20;
74. var alturaLadrillo = 5;
75. var rellenoLadrillos = 2;
76. var vacioSuperiorLadrillos = 10;
77. var vacioIzquierdoLadrillos = 17;
78. var puntaje = 0;
79. var ladrillos = [];
80. for (var columna = 0; columna < numColumnasLadrillos; columna++) {
81. ladrillos[columna] = [];
82. for (var fila = 0; fila < numFilasLadrillos; fila++) {
84. *// estado definira si el ladrillo es visible o no en 10 sera visible, mas adelante permite hacer un degradado*
85. ladrillos[columna][fila] = { *x*: 0, *y*: 0, *estado*: 10, *ciclo*: 10 };
86. }
87. }
88. document.addEventListener(*"keydown"*, teclaPresionada, false);
89. document.addEventListener(*"keyup"*, teclaLiberada, false);
90. function teclaPresionada(e) {
91. if (e.keyCode == 39) {
92. flechaDerechaPulsada = true;
93. } else if (e.keyCode == 37) {
94. flechaIzquierdaPulsada = true;
95. }
96. }
97. function teclaLiberada(e) {
98. if (e.keyCode == 39) {
99. flechaDerechaPulsada = false;
100. } else if (e.keyCode == 37) {
101. flechaIzquierdaPulsada = false;
102. }
103. }
104. function dibujarBola(cordernaX, cordenaY, tamaño, color) {
105. ctx.beginPath();
106. ctx.arc(cordernaX, cordenaY, tamaño, 0, Math.PI \* 2);
107. ctx.fillStyle = color;
108. ctx.fill();
109. ctx.closePath();
110. }
111. function dibujarPaleta() {
112. ctx.beginPath();
113. ctx.rect(paletaPosX, canvas.height - alturaPaleta, anchuraPaleta, alturaPaleta);
114. ctx.fillStyle = *"#f5d109"*;
115. ctx.fill();
116. ctx.closePath();
117. }
118. var colorLadrillos = false; *// La utlizaremos para cambiar el color de los ladrillos*
119. function dibujarLadrillos() {
120. for (var columnas = 0; columnas < numColumnasLadrillos; columnas++) {
121. for (var filas = 0; filas < numFilasLadrillos; filas++) {
122. if (ladrillos[columnas][filas].estado > 0) {
123. var b = ladrillos[columnas][filas];
124. var brickX = filas \* (anchoLadrillo + rellenoLadrillos) + vacioIzquierdoLadrillos;
125. var brickY = columnas \* (alturaLadrillo + rellenoLadrillos) + vacioSuperiorLadrillos;
126. *// Se reduce el estado y el ciclo del ladrillo para posteriormente hacer un degradado y eliminar el ladrillo*
127. if (b.estado < 10 && b.estado > 1) {
128. b.ciclo--;
129. if (b.ciclo == 0) {
130. if (b.estado > 1) {
131. b.estado--;
132. b.ciclo = 10;
133. }
134. }
135. }
136. ladrillos[columnas][filas].x = brickX;
137. ladrillos[columnas][filas].y = brickY;
138. ctx.beginPath();
139. ctx.rect(brickX, brickY, anchoLadrillo, alturaLadrillo);
140. *// las filas cuyo numero es par, se pintaran diferentes a las impares*
141. if (b.estado == 10) {
142. if (colorLadrillos) {
143. if (filas % 2) {
144. ctx.fillStyle = *"#e61c1cdc"*;
145. } else {
146. ctx.fillStyle = *"#e4bb06fa"*;
147. }
148. } else {
149. if (filas % 2) {
150. ctx.fillStyle = *"#e4bb06fa"*;
151. } else {
152. ctx.fillStyle = *"#e61c1cdc"*;
153. }
154. }
155. *// Como anteriormente el estado se estara reduciendo los ladrillos que hayan sido golpeados, cambiaran de color hasta desaparecer.*
157. } else if (b.estado == 9) {
158. ctx.fillStyle = *"#fad638"*;
159. } else if (b.estado == 8) {
160. ctx.fillStyle = *"#fbdb51"*;
161. } else if (b.estado == 7) {
162. ctx.fillStyle = *"#fbe16a"*;
163. } else if (b.estado == 6) {
164. ctx.fillStyle = *"#fce683"*;
165. } else if (b.estado == 5) {
166. ctx.fillStyle = *"#fceb9c"*;
167. } else if (b.estado == 4) {
168. ctx.fillStyle = *"#fdf0b4"*;
169. } else if (b.estado == 3) {
170. ctx.fillStyle = *"#fef5cd"*;
171. } else if (b.estado == 2) {
172. ctx.fillStyle = *"#fefae6"*;
173. } else if (b.estado == 1) {
174. ctx.fillStyle = *"transparent"*;
175. }
176. ctx.fill();
177. ctx.closePath();
178. }
179. }
180. }
181. }
182. *// Detectar la colision*
183. function deteccionColision() {
184. *// Los ciclos recorre todos los ladrillos*
185. for (var c = 0; c < numColumnasLadrillos; c++) {
186. for (var f = 0; f < numFilasLadrillos; f++) {
187. var b = ladrillos[c][f];
188. *// Funcion para reducir el estado, el ciclo, sumar el puntaje y ver si se ha ganado, esto es para evitar copiar el mismo codigo mas adelante*
189. function sumarPuntaje(){
190. b.estado--;
191. b.ciclo--;
192. puntaje++;
193. if (puntaje == numFilasLadrillos \* numColumnasLadrillos) {
194. alert(*"FELICITACIONES: HAS GANADO"*);
195. document.location.reload();
196. }
197. }
198. */\**
199. *En la variable acontinuacion nos permitira obtener un numero al azar*
200. *Este si es par, y si el ladrillo con el cual ha chocado es visible entonces rebotara la bola, llamara a la funcion anteriormente hecha, la cual reducira el estado y hara que el ladrillo se degrade ademas sumara 1 al puntaje,*
202. *De lo contrario si el numero es impar la bola rebotara pero no llamara a la funcion anteriormente hecha, lo cual el estado no se reduce, o sea el ladrillo no desaparece y tambien el puntaje no se suma*
204. *Recordemos que podemos tener dos bolas, lo cual es el mismo proceso solo que cambiando los valores correspondientes a su posicion*
205. *\*/*
206. var num = (Math.round(Math.random() \* 100));
207. if (b.estado == 10) {
209. *// se modifica si la bola llega al ladrillo*
210. if (x > b.x && x < b.x + anchoLadrillo && y > b.y && y < b.y + alturaLadrillo) {
211. dy = -dy;
212. if(num % 2){
213. sumarPuntaje();*// Llama a la funcion para reducir el ciclo, el estado y sumar el puntaje*
214. }
215. }
216. if (segundaBola) {
217. if (x2 > b.x && x2 < b.x + anchoLadrillo && y2 > b.y && y2 < b.y + alturaLadrillo) {
218. dy2 = -dy2;
219. if(num % 2){
220. sumarPuntaje();*// Llama a la funcion para reducir el ciclo, el estado y sumar el puntaje*
221. }
222. }
223. }
224. }
225. }
226. }
227. }
228. *// Esta funcion permitira que la variable puntaje se pueda visualizar en la pantalla, dentro hay una variable que almacena de manera de texto la palabra puntaje con el valor del puntaje correspondientes, ejemplo "Puntaje: 5". Y gracias al elemento div en HTML se mostrara lo que devuelve esta funcion, esta es tambien llamada en la funcion principal dibujar()*
229. function dibujarPuntos(){
230. ctx.font = “24px Poppins”
231. ctx.fillStyle = #2496f3;
232. ctx.fillText(“puntaje: ” + puntaje, 8, 20);
233. }
234. *// Función principal para generar los procesos*
235. function dibujar() {
236. ctx.clearRect(0, 0, canvas.width, canvas.height);
237. dibujarBola(x, y, ballRadius, *"#6317B9"*); *// dibuja la bola*
238. *// Dibuja una segunda bola*
239. if (segundaBola) {
240. dibujarBola(x2, y2, ballRadius - 1, *"#2196f3"*);
241. }
242. dibujarPaleta(); *// Dibuja la paleta*
243. dibujarLadrillos(); *// Dibuja el conjunto de los ladrillos*
244. *// SE LLAMA A LA FUNCION DETECTAR COLISION*
245. deteccionColision();
246. *// SE LLAMA A LA FUNCION Dibujar puntos*
247. dibujarPuntos();
249. *// Rebote de la bola*
250. if (x + dx > canvas.width - ballRadius || x + dx < ballRadius) {
251. dx = -dx;
252. colorLadrillos = false;*// Cambia el color de los ladrillos si la bola principal rebota en x*
253. }
254. if (y + dy < ballRadius) {
255. dy = -dy;
256. var cambia = document.querySelector(*"canvas"*);
257. cambia.classList.toggle(*"activado"*);
258. colorLadrillos = true;
259. }
260. if (x2 + dx2 > canvas.width - ballRadius || x2 + dx2 < ballRadius) {
261. dx2 = -dx2;
262. }
263. if (y2 + dy2 < ballRadius) {
264. dy2 = -dy2;
265. }
266. *// Rebotar en Paleta o Game Over*
267. if (y + dy > canvas.height - ballRadius) {
268. if (x > paletaPosX && x < paletaPosX + anchuraPaleta) {
269. dy = -dy;
270. } else {
271. clearInterval(juego);
272. alert(*"Game Over"*);
273. document.location.reload();
274. }
275. }
276. if (segundaBola) {
277. if (y2 + dy2 > canvas.height - ballRadius) {
278. if (x2 > paletaPosX && x2 < paletaPosX + anchuraPaleta) {
279. dy2 = -dy2;
280. } else {
281. clearInterval(juego);
282. alert(*"Game Over"*);
283. document.location.reload();
284. }
285. }
286. }
287. *// Movimiento Paleta*
288. if (flechaDerechaPulsada && paletaPosX < canvas.width - anchuraPaleta) {
289. paletaPosX += 7;
290. } else if (flechaIzquierdaPulsada && paletaPosX > 0) {
291. paletaPosX -= 7;
292. }
293. *// Se incrementan los valores de la pocion de las bolas*
294. x += dx;
295. y += dy;
296. x2 += dx2;
297. y2 += dy2;
298. }
299. var juego = setInterval(dibujar, 20);
300. </script>
301. </body>
302. </html>

El resultado es el siguiente:



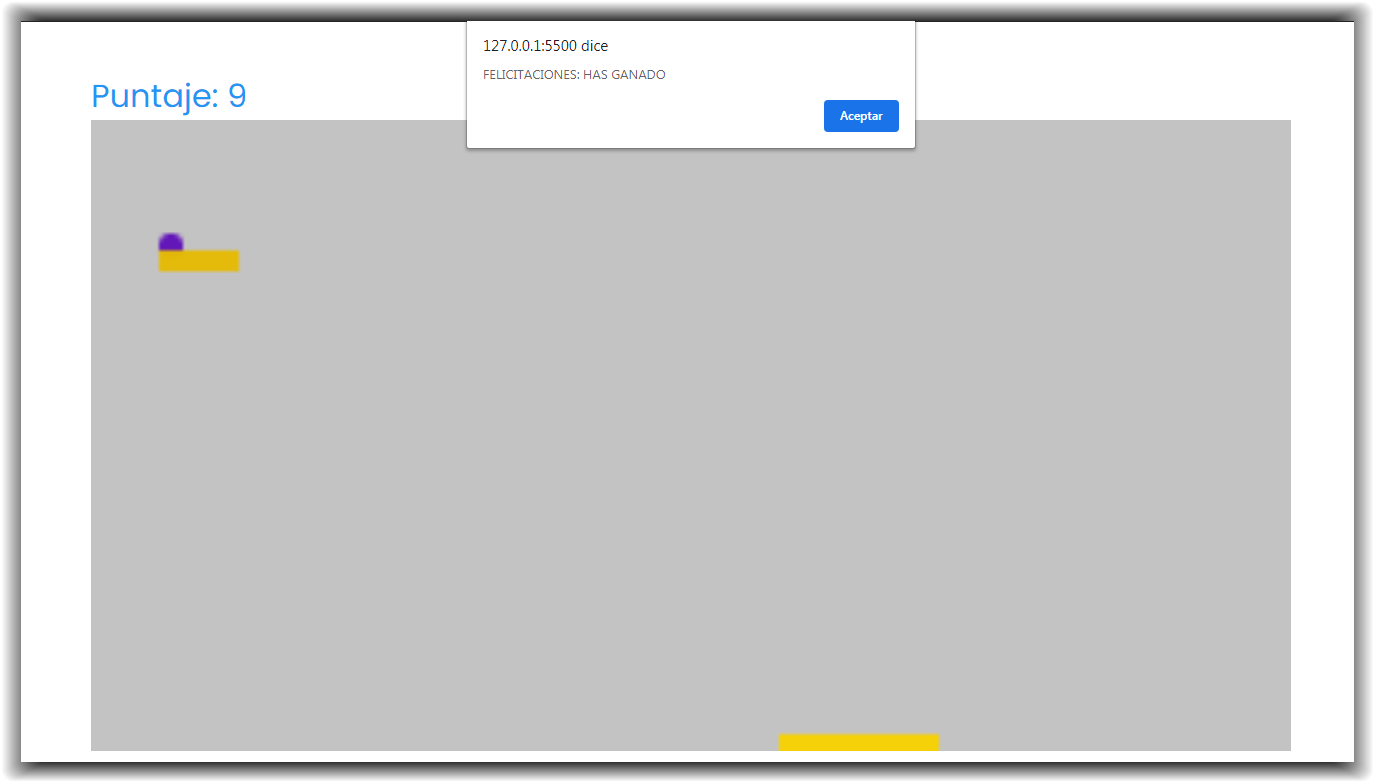
**Gráfica 12. Interfaz cuando la bola golpea los ladrillos.**

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



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

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



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

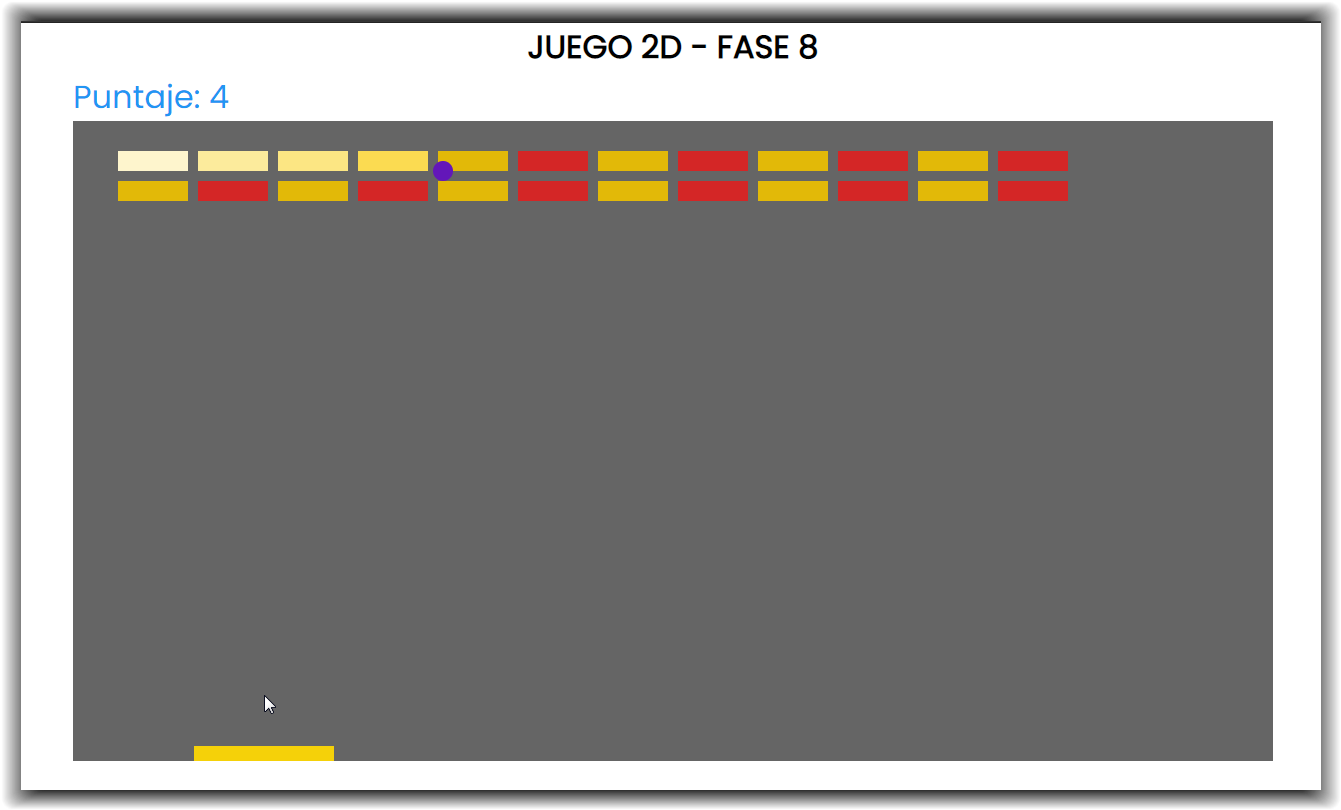
# FASE 8: Controlando el ratón

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

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

1. <!DOCTYPE html>
2. <html>
3. <head>
4. <meta charset=*"UTF-8"* />
5. <title>Juego 2D</title>
6. <style>
7. @import url(*"https://fonts.googleapis.com/css2?family=Poppins&display=swap"*);
8. \* {
9. font-family: *"Poppins"*, sans-serif;
10. padding: 0;
11. margin: 0;
12. }
13. body {
14. margin: 0 200px 0 200px;
15. }
16. h1 {
17. font-size: 2em;
18. text-transform: uppercase;
19. text-align: center;
20. }
21. canvas {
22. background-color: #dddddd;
23. transition: 0.5s;
24. display: block;
25. margin: 50px auto;
26. }
27. canvas.activado {
28. background-color: #222222;
29. }
30. #puntos {
31. font-size: 2em;
32. position: fixed;
33. top: 50px;
34. color: #2196f3;
35. }
36. </style>
37. </head>
38. <body>
39. <h1>Juego 2D - Fase 8</h1>
40. <canvas id=*"miCanvas"* width=*"1200"* height=*"640"*> </canvas>
41. <div id=*"puntos"*>
42. </div>
43. <script>
44. var canvas = document.getElementById(*"miCanvas"*);
45. var ctx = canvas.getContext(*"2d"*);
46. var ballRadius = 10;
47. var x = canvas.width / 2;
48. var y = canvas.height - 30;
49. var dx = 5;
50. var dy = -5;
51. var x2 = x;
52. var y2 = y;
53. var dx2 = -1.5;
54. var dy2 = -1.5;
55. var segundaBola = false;
56. if (Math.random() \* 100 > 50) {
57. segundaBola = true;
58. }
59. *// Paleta*
60. var alturaPaleta = 15;
61. var anchuraPaleta = 140;
62. if (segundaBola) {
63. anchuraPaleta = 180;
64. }
65. var paletaPosX = (canvas.width - anchuraPaleta) / 2;
66. var flechaDerechaPulsada = false;
67. var flechaIzquierdaPulsada = false;
68. var numFilasLadrillos = 0;
69. var numColumnasLadrillos = 0;
70. do {
71. numFilasLadrillos = Math.round(Math.random() \* 100);
72. numColumnasLadrillos = Math.round(Math.random() \* 10);
73. } while (numFilasLadrillos < 1 || numFilasLadrillos > 14 || numColumnasLadrillos < 1 || numColumnasLadrillos > 8);
74. var anchoLadrillo = 70;
75. var alturaLadrillo = 20;
76. var rellenoLadrillos = 10;
77. var vacioSuperiorLadrillos = 30;
78. var vacioIzquierdoLadrillos = 45;
79. var puntaje = 0;
80. var ladrillos = [];
81. for (var columna = 0; columna < numColumnasLadrillos; columna++) {
82. ladrillos[columna] = [];
83. for (var fila = 0; fila < numFilasLadrillos; fila++) {
84. ladrillos[columna][fila] = { *x*: 0, *y*: 0, *estado*: 10, *ciclo*: 10 };
85. }
86. }
87. document.addEventListener(*"keydown"*, teclaPresionada, false);
88. document.addEventListener(*"keyup"*, teclaLiberada, false);
89. document.addEventListener(*"mousemove"*, manejadorRaton, false);
90. function teclaPresionada(e) {
91. if (e.keyCode == 39) {
92. flechaDerechaPulsada = true;
93. } else if (e.keyCode == 37) {
94. flechaIzquierdaPulsada = true;
95. }
96. }
97. function teclaLiberada(e) {
98. if (e.keyCode == 39) {
99. flechaDerechaPulsada = false;
100. } else if (e.keyCode == 37) {
101. flechaIzquierdaPulsada = false;
102. }
103. }
104. *// ESTE ES EL MANEJADOR DEL RATÓN*
105. *// ------------------------------*
106. *// La instrucción: "offsetLeft" calcula la distancia desde el borde izquierdo*
107. *// de la pantalla hasta un componente html*
108. *// ------------------------------*
109. *// Por tanto, la instrucción: "canvas.offsetLeft" calcula el espacio a la izquierda*
110. *// del objeto CANVAS*
111. *// ------------------------------*
112. *// Dentro del manejador del ratón, la instrucción: "e.clientX" calcula la posición*
113. *// del ratón en la pantalla. Para calcular la posición del ratón DENTRO del CANVAS*
114. *// debemos RESTAR a la posición X del ratón, el valor izquierdo del CANVAS*
115. *// ------------------------------*
116. *// Es decir: "e.clientX - canvas.offsetLeft"*
117. *// ------------------------------*
119. function manejadorRaton(e){
120. var posXRatonDentroDeCanvas = e.clientX - canvas.offsetLeft;
121. *// La siguiente condicional determinara si el raton esta dentro del CANVAS*
122. if (posXRatonDentroDeCanvas > anchuraPaleta / 2 && posXRatonDentroDeCanvas < canvas.width - anchuraPaleta / 2) {
123. *// Si el raton esta dentro del canvas. Se recalcula la posicion de la paleta, la posicon de la paleta es ahora la mima posicion del ratón. Se resta ya que la paleta tiene un ancho, gracias a esto la paleta se mueve a basa del ratón*
124. paletaPosX = posXRatonDentroDeCanvas - anchuraPaleta / 2;
125. }
126. }
127. function dibujarBola(cordernaX, cordenaY, tamaño, color) {
128. ctx.beginPath();
129. ctx.arc(cordernaX, cordenaY, tamaño, 0, Math.PI \* 2);
130. ctx.fillStyle = color;
131. ctx.fill();
132. ctx.closePath();
133. }
134. function dibujarPaleta() {
135. ctx.beginPath();
136. ctx.rect(paletaPosX, canvas.height - alturaPaleta, anchuraPaleta, alturaPaleta);
137. ctx.fillStyle = *"#f5d109"*;
138. ctx.fill();
139. ctx.closePath();
140. }
141. var colorLadrillos = false; *// La utlizaremos para cambiar el color de los ladrillos*
142. function dibujarLadrillos() {
143. for (var columnas = 0; columnas < numColumnasLadrillos; columnas++) {
144. for (var filas = 0; filas < numFilasLadrillos; filas++) {
145. if (ladrillos[columnas][filas].estado > 0) {
146. var b = ladrillos[columnas][filas];
147. var brickX = filas \* (anchoLadrillo + rellenoLadrillos) + vacioIzquierdoLadrillos;
148. var brickY = columnas \* (alturaLadrillo + rellenoLadrillos) + vacioSuperiorLadrillos;
149. if (b.estado < 10 && b.estado > 1) {
150. b.ciclo--;
151. if (b.ciclo == 0) {
152. if (b.estado > 1) {
153. b.estado--;
154. b.ciclo = 10;
155. }
156. }
157. }
158. ladrillos[columnas][filas].x = brickX;
159. ladrillos[columnas][filas].y = brickY;
160. ctx.beginPath();
161. ctx.rect(brickX, brickY, anchoLadrillo, alturaLadrillo);
162. if (b.estado == 10) {
163. if (colorLadrillos) {
164. if (filas % 2) {
165. ctx.fillStyle = *"#e61c1cdc"*;
166. } else {
167. ctx.fillStyle = *"#e4bb06fa"*;
168. }
169. } else {
170. if (filas % 2) {
171. ctx.fillStyle = *"#e4bb06fa"*;
172. } else {
173. ctx.fillStyle = *"#e61c1cdc"*;
174. }
175. }
176. } else if (b.estado == 9) {
177. ctx.fillStyle = *"#fad638"*;
178. } else if (b.estado == 8) {
179. ctx.fillStyle = *"#fbdb51"*;
180. } else if (b.estado == 7) {
181. ctx.fillStyle = *"#fbe16a"*;
182. } else if (b.estado == 6) {
183. ctx.fillStyle = *"#fce683"*;
184. } else if (b.estado == 5) {
185. ctx.fillStyle = *"#fceb9c"*;
186. } else if (b.estado == 4) {
187. ctx.fillStyle = *"#fdf0b4"*;
188. } else if (b.estado == 3) {
189. ctx.fillStyle = *"#fef5cd"*;
190. } else if (b.estado == 2) {
191. ctx.fillStyle = *"#fefae6"*;
192. } else if (b.estado == 1) {
193. ctx.fillStyle = *"transparent"*;
194. }
195. ctx.fill();
196. ctx.closePath();
197. }
198. }
199. }
200. }
201. *// Detectar la colision*
202. function deteccionColision() {
203. for (var c = 0; c < numColumnasLadrillos; c++) {
204. for (var f = 0; f < numFilasLadrillos; f++) {
205. var b = ladrillos[c][f];
206. function sumarPuntaje() {
207. b.estado--;
208. b.ciclo--;
209. puntaje++;
210. if (puntaje == numFilasLadrillos \* numColumnasLadrillos) {
211. alert(*"FELICITACIONES: HAS GANADO"*);
212. document.location.reload();
213. }
214. }
215. if (b.estado == 10) {
216. *// se modifica si la bola llega al ladrillo*
217. if (x > b.x && x < b.x + anchoLadrillo + ballRadius/2 && y > b.y && y < b.y + alturaLadrillo + ballRadius/2) {
218. dy = -dy;
219. if ((Math.round(Math.random() \* 100)) % 2) {
220. sumarPuntaje();*// Llama a la funcion para reducir el ciclo, el estado y sumar el puntaje*
221. }
222. }
223. if (segundaBola) {
224. if (x2 > b.x && x2 < b.x + anchoLadrillo && y2 > b.y && y2 < b.y + alturaLadrillo) {
225. dy2 = -dy2;
226. if ((Math.round(Math.random() \* 100)) % 2) {
227. sumarPuntaje();*// Llama a la funcion para reducir el ciclo, el estado y sumar el puntaje*
228. }
229. }
230. }
231. }
232. }
233. }
234. }
235. function dibujarPuntos() {
236. var str = *'Puntaje:* *'* + puntaje.toString();
237. document.getElementById(*'puntos'*).innerText = str;
238. }
239. function dibujar() {
240. ctx.clearRect(0, 0, canvas.width, canvas.height);
241. *// SE LLAMA A LA FUNCION Dibujar puntos*
242. dibujarPuntos();
243. dibujarLadrillos();
244. dibujarBola(x, y, ballRadius, *"#6317B9"*);
245. if (segundaBola) {
246. dibujarBola(x2, y2, ballRadius - 2, *"#2196f3"*);
247. }
248. dibujarPaleta();
249. deteccionColision();
250. *// Rebote de la bola*
251. if (x + dx > canvas.width - ballRadius || x + dx < ballRadius) {
252. dx = -dx;
253. colorLadrillos = false;*// Cambia el color de los ladrillos si la bola principal rebota en x*
254. }
255. if (y + dy < ballRadius) {
256. dy = -dy;
257. var cambia = document.querySelector(*"canvas"*);
258. cambia.classList.toggle(*"activado"*);
259. colorLadrillos = true;
260. }
261. if (x2 + dx2 > canvas.width - ballRadius || x2 + dx2 < ballRadius) {
262. dx2 = -dx2;
263. }
264. if (y2 + dy2 < ballRadius) {
265. dy2 = -dy2;
266. }
267. *// Rebotar en Paleta o Game Over*
268. if (y + dy > canvas.height - ballRadius) {
269. if (x > paletaPosX && x < paletaPosX + anchuraPaleta) {
270. dy = -dy;
271. } else {
272. clearInterval(juego);
273. alert(*"Game Over"*);
274. document.location.reload();
275. }
276. }
277. if (segundaBola) {
278. if (y2 + dy2 > canvas.height - ballRadius) {
279. if (x2 > paletaPosX && x2 < paletaPosX + anchuraPaleta) {
280. dy2 = -dy2;
281. } else {
282. clearInterval(juego);
283. alert(*"Game Over"*);
284. document.location.reload();
285. }
286. }
287. }
288. *// Movimiento Paleta*
289. if (flechaDerechaPulsada && paletaPosX < canvas.width - anchuraPaleta) {
290. paletaPosX += 15;
291. } else if (flechaIzquierdaPulsada && paletaPosX > 0) {
292. paletaPosX -= 15;
293. }
294. *// Se incrementan los valores de la pocion de las bolas*
295. x += dx;
296. y += dy;
297. x2 += dx2;
298. y2 += dy2;
299. }
300. var juego = setInterval(dibujar, 15);
301. </script>
302. </body>
303. </html>

El resultado es:



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

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

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

# FASE 9: Finalizando el juego

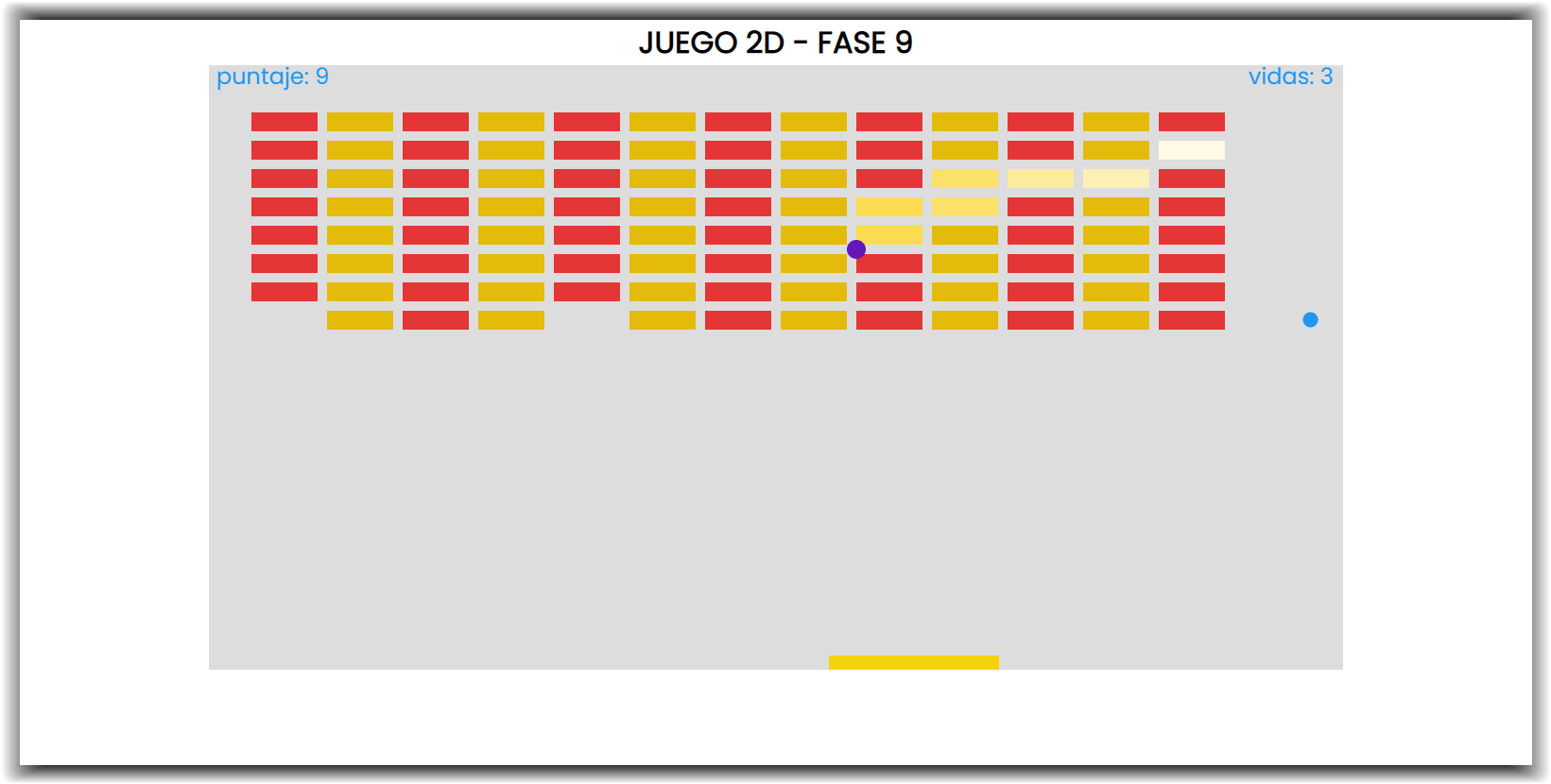
En esta parte del programa ya se agregan los toques finales tales como vidas para el jugador, ocultar el mouse entre otras.

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

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

1. <!DOCTYPE html>
2. <html>
3. <head>
4. <meta charset=*"UTF-8"* />
5. <title>Juego 2D</title>
6. <style>
7. @import url(*"https://fonts.googleapis.com/css2?family=Poppins&display=swap"*);
8. \* {
9. font-family: *"Poppins"*, sans-serif;
10. padding: 0;
11. margin: 0;
12. }
13. h1 {
14. font-size: 2em;
15. text-transform: uppercase;
16. text-align: center;
17. }
18. canvas {
19. background-color: #dddddd;
20. transition: 0.5s;
21. display: block;
22. margin: auto;
23. */\* OCULTA EL CURSOR EN EL CANVAS \*/*
24. cursor: none;
25. }
26. canvas.activado {
27. background-color: #222222;
28. }
29. </style>
30. </head>
31. <body>
32. <h1>Juego 2D - Fase 9</h1>
33. <canvas id=*"miCanvas"* width=*"1200"* height=*"640"*> </canvas>
34. <script>
35. var canvas = document.getElementById(*"miCanvas"*);
36. var ctx = canvas.getContext(*"2d"*);
37. var ballRadius = 10;
38. var x = canvas.width / 2;
39. var y = canvas.height - 30;
40. var dx = 5;
41. var dy = -5;
42. var x2 = x;
43. var y2 = y;
44. var dx2 = -1.5;
45. var dy2 = -1.5;
46. var segundaBola = false;
47. if (Math.random() \* 100 > 50) {
48. segundaBola = true;
49. }
50. *// Paleta*
51. var alturaPaleta = 15;
52. var anchuraPaleta = 140;
53. if (segundaBola) {
54. anchuraPaleta = 180;
55. }
56. var paletaPosX = (canvas.width - anchuraPaleta) / 2;
57. var flechaDerechaPulsada = false;
58. var flechaIzquierdaPulsada = false;
59. var numFilasLadrillos = 0;
60. var numColumnasLadrillos = 0;
61. do {
62. numFilasLadrillos = Math.round(Math.random() \* 100);
63. numColumnasLadrillos = Math.round(Math.random() \* 10);
64. } while (numFilasLadrillos < 1 || numFilasLadrillos > 14 || numColumnasLadrillos < 1 || numColumnasLadrillos > 8);
65. var anchoLadrillo = 70;
66. var alturaLadrillo = 20;
67. var rellenoLadrillos = 10;
68. var vacioSuperiorLadrillos = 50;
69. var vacioIzquierdoLadrillos = 45;
70. var puntaje = 0;
71. *// ESTA INSTRUCCIÓN CONTROLA EL NÚMERO DE VIDAS DEL JUGADOR*
72. *// CUANDO LA INSTRUCCIÓN vidas DISMINUYE A CERO, EL JUGADOR PIERDE,*
73. *// PUESTO QUE HA PERDIDO TRES VECES*
74. *// ESTA VARIABLE DEFINE UN COLOR*
75. *// Se pueden utilizar otros colores para los diferentes elementos del juego*
76. var colorFigura = *"#ff0000"*;
77. var colorBola = *"#6317B9"*;
78. var colorBola2 = *"#2196f3"*;
79. var colorPaleta = *"#f5d109"*;
80. var colorLadrillo = *"#e61c1cdc"*;
81. var colorLadrillo2 = *"#e4bb06fa"*;
82. var colorTexto = *"#2196f3"*;
83. var vidas = 3;
84. *// ESTA INSTRUCCIÓN OCULTA EL CURSOR DEL RATON (DENTRO DEL CANVAS)*
85. canvas.style.cursor = *'none'*;
86. var ladrillos = [];
87. for (var columna = 0; columna < numColumnasLadrillos; columna++) {
88. ladrillos[columna] = [];
89. for (var fila = 0; fila < numFilasLadrillos; fila++) {
90. ladrillos[columna][fila] = { *x*: 0, *y*: 0, *estado*: 10, *ciclo*: 10 };
91. }
92. }
93. document.addEventListener(*"keydown"*, teclaPresionada, false);
94. document.addEventListener(*"keyup"*, teclaLiberada, false);
95. document.addEventListener(*"mousemove"*, manejadorRaton, false);
96. function teclaPresionada(e) {
97. if (e.keyCode == 39) {
98. flechaDerechaPulsada = true;
99. } else if (e.keyCode == 37) {
100. flechaIzquierdaPulsada = true;
101. }
102. }
103. function teclaLiberada(e) {
104. if (e.keyCode == 39) {
105. flechaDerechaPulsada = false;
106. } else if (e.keyCode == 37) {
107. flechaIzquierdaPulsada = false;
108. }
109. }
110. function manejadorRaton(e) {
111. var posXRatonDentroDeCanvas = e.clientX - canvas.offsetLeft;
112. if (posXRatonDentroDeCanvas > anchuraPaleta / 2 && posXRatonDentroDeCanvas < canvas.width - anchuraPaleta / 2) {
113. paletaPosX = posXRatonDentroDeCanvas - anchuraPaleta / 2;
114. }
115. }
116. function dibujarBola(cordernaX, cordenaY, tamaño, color) {
117. ctx.beginPath();
118. ctx.arc(cordernaX, cordenaY, tamaño, 0, Math.PI \* 2);
119. ctx.fillStyle = color;
120. ctx.fill();
121. ctx.closePath();
122. }
123. function dibujarPaleta() {
124. ctx.beginPath();
125. ctx.rect(paletaPosX, canvas.height - alturaPaleta, anchuraPaleta, alturaPaleta);
126. ctx.fillStyle = colorPaleta;
127. ctx.fill();
128. ctx.closePath();
129. }
130. var colorLadrillos = false; *// La utlizaremos para cambiar el color de los ladrillos*
131. function dibujarLadrillos() {
132. for (var columnas = 0; columnas < numColumnasLadrillos; columnas++) {
133. for (var filas = 0; filas < numFilasLadrillos; filas++) {
134. if (ladrillos[columnas][filas].estado > 0) {
135. var b = ladrillos[columnas][filas];
136. var brickX = filas \* (anchoLadrillo + rellenoLadrillos) + vacioIzquierdoLadrillos;
137. var brickY = columnas \* (alturaLadrillo + rellenoLadrillos) + vacioSuperiorLadrillos;
138. if (b.estado < 10 && b.estado > 1) {
139. b.ciclo--;
140. if (b.ciclo == 0) {
141. if (b.estado > 1) {
142. b.estado--;
143. b.ciclo = 10;
144. }
145. }
146. }
147. ladrillos[columnas][filas].x = brickX;
148. ladrillos[columnas][filas].y = brickY;
149. ctx.beginPath();
150. ctx.rect(brickX, brickY, anchoLadrillo, alturaLadrillo);
151. if (b.estado == 10) {
152. if (colorLadrillos) {
153. if (filas % 2) {
154. ctx.fillStyle = colorLadrillo;
155. } else {
156. ctx.fillStyle = colorLadrillo2;
157. }
158. } else {
159. if (filas % 2) {
160. ctx.fillStyle = colorLadrillo2;
161. } else {
162. ctx.fillStyle = colorLadrillo;
163. }
164. }
165. } else if (b.estado == 9) {
166. ctx.fillStyle = *"#fad638"*;
167. } else if (b.estado == 8) {
168. ctx.fillStyle = *"#fbdb51"*;
169. } else if (b.estado == 7) {
170. ctx.fillStyle = *"#fbe16a"*;
171. } else if (b.estado == 6) {
172. ctx.fillStyle = *"#fce683"*;
173. } else if (b.estado == 5) {
174. ctx.fillStyle = *"#fceb9c"*;
175. } else if (b.estado == 4) {
176. ctx.fillStyle = *"#fdf0b4"*;
177. } else if (b.estado == 3) {
178. ctx.fillStyle = *"#fef5cd"*;
179. } else if (b.estado == 2) {
180. ctx.fillStyle = *"#fefae6"*;
181. } else if (b.estado == 1) {
182. ctx.fillStyle = *"transparent"*;
183. }
184. ctx.fill();
185. ctx.closePath();
186. }
187. }
188. }
189. }
190. *// Detectar la colision*
191. function deteccionColision() {
192. for (var c = 0; c < numColumnasLadrillos; c++) {
193. for (var f = 0; f < numFilasLadrillos; f++) {
194. var b = ladrillos[c][f];
195. function sumarPuntaje() {
196. b.estado--;
197. b.ciclo--;
198. puntaje++;
199. if (puntaje == numFilasLadrillos \* numColumnasLadrillos) {
200. alert(*"FELICITACIONES: HAS GANADO"*);
201. document.location.reload();
202. }
203. }
204. if (b.estado == 10) {
205. *// se modifica si la bola llega al ladrillo*
206. if (x > b.x && x < b.x + anchoLadrillo + ballRadius / 2 && y > b.y && y < b.y + alturaLadrillo + ballRadius / 2) {
207. dy = -dy;
208. if ((Math.round(Math.random() \* 100)) % 2) {
209. sumarPuntaje();*// Llama a la funcion para reducir el ciclo, el estado y sumar el puntaje*
210. }
211. }
212. if (segundaBola) {
213. if (x2 > b.x && x2 < b.x + anchoLadrillo && y2 > b.y && y2 < b.y + alturaLadrillo) {
214. dy2 = -dy2;
215. if ((Math.round(Math.random() \* 100)) % 2) {
216. sumarPuntaje();*// Llama a la funcion para reducir el ciclo, el estado y sumar el puntaje*
217. }
218. }
219. }
220. }
221. }
222. }
223. }
224. function dibujarPuntos() {
225. ctx.font = *"24px Poppins"*;
226. ctx.fillStyle = colorTexto;
227. ctx.fillText(*"puntaje:* *"* + puntaje, 8, 20);
228. }
229. function dibujarVidas() {
230. ctx.font = *"24px Poppins"*;
231. ctx.fillStyle = colorTexto;
232. *// SE MUESTRA EL NÚMERO DE VIDAS DISPONIBLES*
233. ctx.fillText(*"vidas:* *"* + vidas, canvas.width - 100, 20);
234. }
235. */\* La siguiente funcion permitira evitar escribir el codigo dos veces ya que tenemos posiblemente dos bolas \*/*
236. function quitarVidas() {
237. *// SI SE PRODUCE UN CONTACTO DE LA BOLA CON LA BASE DEL CANVAS*
238. *// SE PIERDE UNA VIDA. PARA ELLO, LA INSTRUCCIÓN vidas--;*
239. *// LO CUAL EQUIVALE A: vidas = vidas - 1*
240. vidas--;
241. if (!vidas) {
242. *// SI vidas == 0 (lo cual también puede escribir: !vidas)*
243. *// EL JUGADOR HA PERDIDO*
244. alert(*"GAME OVER"*);
245. document.location.reload();
246. } else {
247. *// SI vidas > 0 (diferente de CERO) EL JUEGO CONTINUA*
248. x = canvas.width / 2;
249. y = canvas.height - 30;
250. x2 = canvas.width / 2;
251. y2 = canvas.height - 30;
252. dx = 5;
253. dy = -5;
254. dx2 = -1.5;
255. dy2 = -1.5;
256. paletaPosX = (canvas.width - anchuraPaleta) / 2;
257. }
258. }
259. function dibujar() {
260. ctx.clearRect(0, 0, canvas.width, canvas.height);
261. *// SE LLAMA A LA FUNCION Dibujar puntos*
262. dibujarPuntos();
263. *// SE LLAMA A LA FUNCION Dibujar puntos*
264. dibujarVidas();
265. dibujarLadrillos();
266. dibujarBola(x, y, ballRadius, colorBola);
267. if (segundaBola) {
268. dibujarBola(x2, y2, ballRadius - 2, colorBola2);
269. }
270. dibujarPaleta();
271. deteccionColision();
272. *// Rebote de la bola*
273. if (x + dx > canvas.width - ballRadius || x + dx < ballRadius) {
274. dx = -dx;
275. colorLadrillos = false;*// Cambia el color de los ladrillos si la bola principal rebota en x*
276. }
277. if (y + dy < ballRadius) {
278. dy = -dy;
279. var cambia = document.querySelector(*"canvas"*);
280. cambia.classList.toggle(*"activado"*);
281. colorLadrillos = true;
282. }
283. if (x2 + dx2 > canvas.width - ballRadius || x2 + dx2 < ballRadius) {
284. dx2 = -dx2;
285. }
286. if (y2 + dy2 < ballRadius) {
287. dy2 = -dy2;
288. }
289. *// Rebotar en Paleta o Game Over*
290. if (y + dy > canvas.height - ballRadius) {
291. if (x > paletaPosX && x < paletaPosX + anchuraPaleta) {
292. dy = -dy;
293. } else {
294. quitarVidas();
295. }
296. }
297. if (segundaBola) {
298. if (y2 + dy2 > canvas.height - ballRadius) {
299. if (x2 > paletaPosX && x2 < paletaPosX + anchuraPaleta) {
300. dy2 = -dy2;
301. } else {
302. quitarVidas();
303. }
304. }
305. }
306. *// Movimiento Paleta*
307. if (flechaDerechaPulsada && paletaPosX < canvas.width - anchuraPaleta) {
308. paletaPosX += 15;
309. } else if (flechaIzquierdaPulsada && paletaPosX > 0) {
310. paletaPosX -= 15;
311. }
312. *// Se incrementan los valores de la pocion de las bolas*
313. x += dx;
314. y += dy;
315. x2 += dx2;
316. y2 += dy2;
317. *// ESTE ES UN SEGUNDO MÉTODO PARA REALIZAR LA ANIMACIÓN DEL JUEGO*
318. *// LA INSTRUCCIÓN: requestAnimationFrame SE EJECUTA 60 VECES POR SEGUNDO*
319. *// Y AL EJECUTARSE LLAMA A LA FUNCIÓN ENTRE PARÉNTESIS*
320. *// POR TANTO, dibujar SE EJECUTA 60 VECES POR SEGUNDO*
321. *// GENERANDO EL CICLO DEL JUEGO*
322. requestAnimationFrame(dibujar);
323. }
324. dibujar();
325. </script>
326. </body>
327. </html>

Al ejecutar este código se obtiene la siguiente interfaz visual:



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

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

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

# FASE 10: Movimiento de ladrillos, misil y explosión

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

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

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

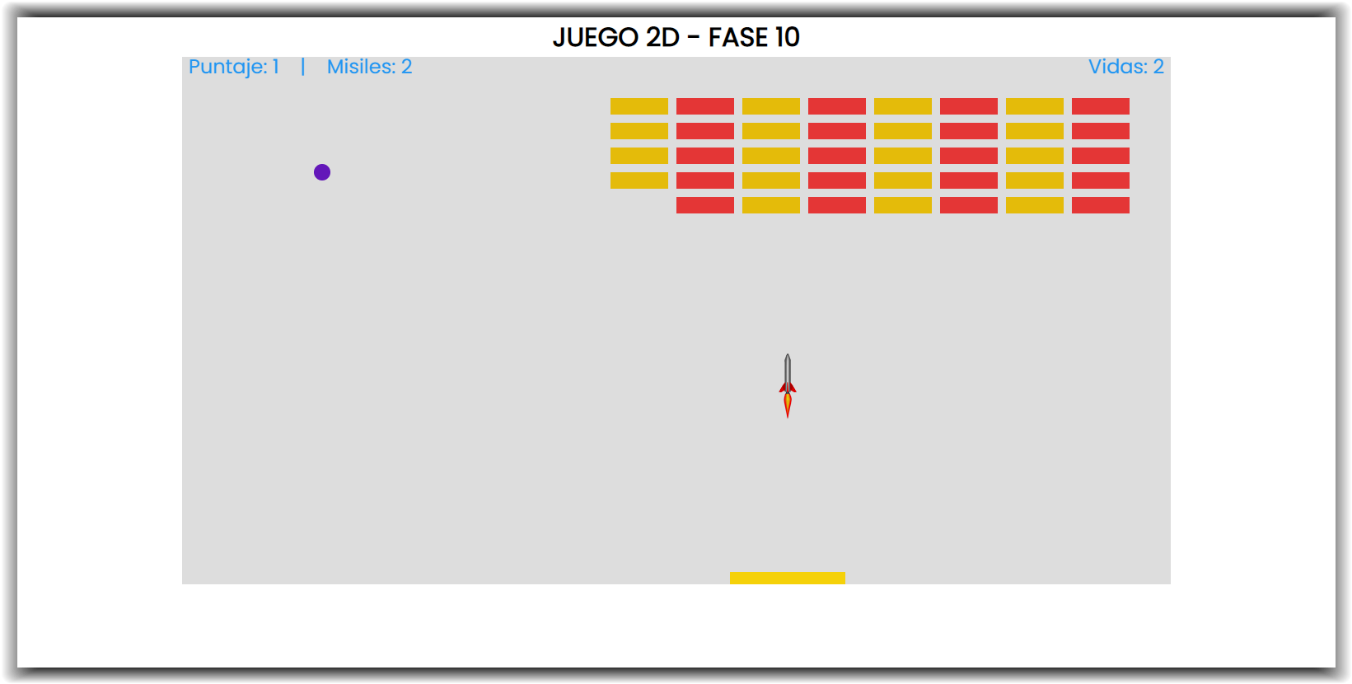
Se crea una variable booleana para saber si el misil ha sido disparado, esto para evitar dos misiles a la misma vez, también la cantidad de misiles, si recordamos anteriormente creamos un evento que llama una función, teclaPresionada(), la utilizaremos en nuestro caso para saber si se ha pulsado la tecla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

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

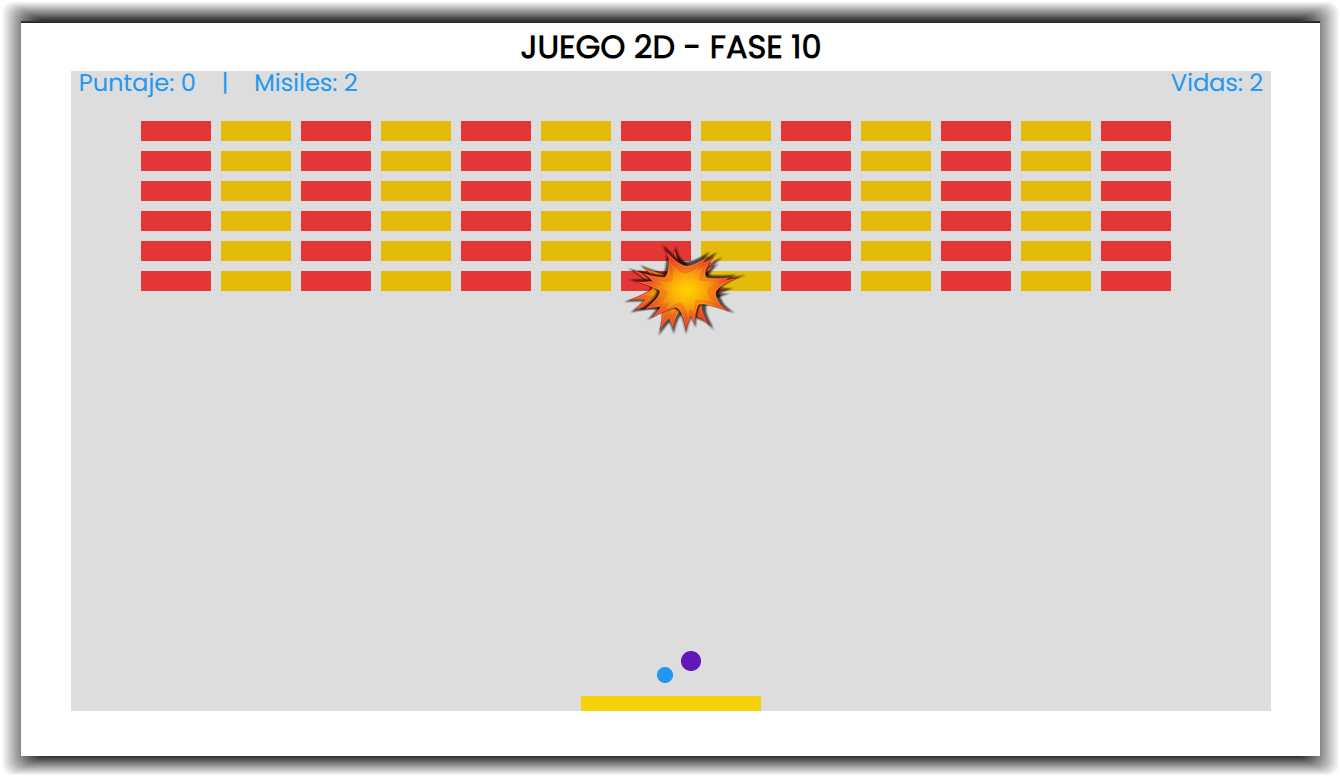
El código es el siguiente:

1. <!DOCTYPE html>
2. <html>
3. <head>
4. <meta charset=*"UTF-8"* />
5. <title>Juego 2D</title>
6. <style>
7. @import url(*"https://fonts.googleapis.com/css2?family=Poppins&display=swap"*);
8. \* {font-family: *"Poppins"*, sans-serif; padding: 0;margin: 0;}
9. h1 {font-size: 2em;text-transform: uppercase;text-align: center;}canvas {background-color: #dddddd;transition: 0.5s;display: block;margin: auto;cursor: none;}
10. canvas.activado {background-color: #222222;}
11. </style>
12. </head>
13. <body>
14. <h1>Juego 2D - Fase 10</h1>
15. <canvas id=*"miCanvas"* width=*"1200"* height=*"640"*> </canvas>
16. <script>
17. var canvas = document.getElementById(*"miCanvas"*);
18. var ctx = canvas.getContext(*"2d"*);
19. var ballRadius = 10;
20. var x = canvas.width / 2;
21. var y = canvas.height - 30;
22. var dx = 5;
23. var dy = -5;
24. var x2 = x;
25. var y2 = y;
26. var dx2 = -1.5;
27. var dy2 = -1.5;
28. var segundaBola = false;
29. if (Math.random() \* 100 > 50) {
30. segundaBola = true;
31. }
32. *// Paleta*
33. var alturaPaleta = 15;
34. var anchuraPaleta = 140;
35. if (segundaBola) {
36. anchuraPaleta = 180;
37. }
38. var paletaPosX = (canvas.width - anchuraPaleta) / 2;
39. var flechaDerechaPulsada = false;
40. var flechaIzquierdaPulsada = false;
41. var numFilasLadrillos = 0;
42. var numColumnasLadrillos = 0;
43. do {
44. numFilasLadrillos = Math.round(Math.random() \* 100);
45. numColumnasLadrillos = Math.round(Math.random() \* 10);
46. } while (numFilasLadrillos < 1 || numFilasLadrillos > 14 || numColumnasLadrillos < 1 || numColumnasLadrillos > 8);
47. var anchoLadrillo = 70;
48. var alturaLadrillo = 20;
49. var rellenoLadrillos = 10;
50. var vacioSuperiorLadrillos = 50;
51. var vacioIzquierdoLadrillos = canvas.width;
52. var puntaje = 0;
53. *// Misil*
54. */\* Se crean las variables para las caracteristicas del misil*
55. *la variable imgMisil nos permitira almacenar una imagen que se encuentra en la carpeta principal*
56. *creamos la posicon y del misil que sera la altura del canvas esto hace que el misil no se vea*
57. *la posiocion x la inicilizamos en (0) para que mas adelante le podamos asignar un valor*
58. *Creamos una segunda imagen para la explosion no tiene coordenadas*
59. *se crea una variable booleana para saber cuando se debe visualizar el misil*
60. *la cantidad de misilis nos permitira ejecurlo tres veces*
61. *Se crea una funcion para dibujar el misil que sera llamada luego*
62. *\*/*
63. var imgMisil = new Image();
64. imgMisil.src = *'misil.png'*;
65. var posY\_Misil = canvas.height;
66. var posX\_Misil = 0;
67. var explosion = new Image();
68. explosion.src = *'explosion.png'*
69. function misil(x) {
70. ctx.drawImage(imgMisil,  x, posY\_Misil);
71. }
72. var dispararMisil = false;
73. var cantidadMisiles = 3;
74. *// Caracteristicas de los elementos*
75. var colorFigura = *"#ff0000"*;
76. var colorBola = *"#6317B9"*;
77. var colorBola2 = *"#2196f3"*;
78. var colorPaleta = *"#f5d109"*;
79. var colorLadrillo = *"#e61c1cdc"*;
80. var colorLadrillo2 = *"#e4bb06fa"*;
81. var colorTexto = *"#2196f3"*;
82. var vidas = 3;
83. var ladrillos = [];
84. for (var columna = 0; columna < numColumnasLadrillos; columna++) {
85. ladrillos[columna] = [];
86. for (var fila = 0; fila < numFilasLadrillos; fila++) {
87. ladrillos[columna][fila] = { *x*: 0, *y*: 0, *estado*: 10, *ciclo*: 10 };
88. }
89. }
90. document.addEventListener(*"keydown"*, teclaPresionada, false);
91. document.addEventListener(*"keyup"*, teclaLiberada, false);
92. document.addEventListener(*"mousemove"*, manejadorRaton, false);
93. function teclaPresionada(e) {
94. if (e.keyCode == 39) {
95. flechaDerechaPulsada = true;
96. } else if (e.keyCode == 37) {
97. flechaIzquierdaPulsada = true;
98. } else if(e.keyCode == 32){
99. */\* Si se pulsa la tecla numero 32 que corresponde al la tecla espacio, entonces se evaluara si disparaMisil es falso ademas y cantidad de misiles es mayor a 1, de ser asi la coordenada x del misil sera la cordena de la paleta sumandole la mitad de la anchura de esta, esto para que el misil salga desde la mitad de la paleta no importa donde esta este situada en el CANVAS, ademas se restan cantidadMisiles en (1) \*/*
100. if(!dispararMisil && cantidadMisiles > 0){
101. posX\_Misil = paletaPosX + anchuraPaleta/2 - 20;
102. dispararMisil = true;
103. cantidadMisiles--;
104. }
105. }
106. }
107. function teclaLiberada(e) {
108. if (e.keyCode == 39) {
109. flechaDerechaPulsada = false;
110. } else if (e.keyCode == 37) {
111. flechaIzquierdaPulsada = false;
112. }
113. }
114. function manejadorRaton(e) {
115. var posXRatonDentroDeCanvas = e.clientX - canvas.offsetLeft;
116. if (posXRatonDentroDeCanvas > anchuraPaleta / 2 && posXRatonDentroDeCanvas < canvas.width - anchuraPaleta / 2) {
117. paletaPosX = posXRatonDentroDeCanvas - anchuraPaleta / 2;
118. }
119. }
120. function dibujarBola(cordernaX, cordenaY, tamaño, color) {
121. ctx.beginPath();
122. ctx.arc(cordernaX, cordenaY, tamaño, 0, Math.PI \* 2);
123. ctx.fillStyle = color;
124. ctx.fill();
125. ctx.closePath();
126. }
127. function dibujarPaleta() {
128. ctx.beginPath();
129. ctx.rect(paletaPosX, canvas.height - alturaPaleta, anchuraPaleta, alturaPaleta);
130. ctx.fillStyle = colorPaleta;
131. ctx.fill();
132. ctx.closePath();
133. }
134. var colorLadrillos = false;
135. */\* Nos permitira mover los ladrillos mas adelante \*/*
136. var moverDerecha = false;
137. function dibujarLadrillos() {
138. for (var columnas = 0; columnas < numColumnasLadrillos; columnas++) {
139. for (var filas = 0; filas < numFilasLadrillos; filas++) {
140. if (ladrillos[columnas][filas].estado > 0) {
141. var b = ladrillos[columnas][filas];
142. var brickX = filas \* (anchoLadrillo + rellenoLadrillos) + vacioIzquierdoLadrillos;
143. var brickY = columnas \* (alturaLadrillo + rellenoLadrillos) + vacioSuperiorLadrillos;
144. if (b.estado < 10 && b.estado > 1) {
145. b.ciclo--;
146. if (b.ciclo == 0) {
147. if (b.estado > 1) {
148. b.estado--;
149. b.ciclo = 10;
150. }
151. }
152. }
153. ladrillos[columnas][filas].x = brickX;
154. ladrillos[columnas][filas].y = brickY;
155. ctx.beginPath();
156. ctx.rect(brickX, brickY, anchoLadrillo, alturaLadrillo);
157. if (b.estado == 10) {
158. if (colorLadrillos) {
159. if (filas % 2) {
160. ctx.fillStyle = colorLadrillo;
161. } else {
162. ctx.fillStyle = colorLadrillo2;
163. }
164. } else {
165. if (filas % 2) {
166. ctx.fillStyle = colorLadrillo2;
167. } else {
168. ctx.fillStyle = colorLadrillo;
169. }
170. }
171. } else if (b.estado == 9) {
172. ctx.fillStyle = *"#fad638"*;
173. } else if (b.estado == 8) {
174. ctx.fillStyle = *"#fbdb51"*;
175. } else if (b.estado == 7) {
176. ctx.fillStyle = *"#fbe16a"*;
177. } else if (b.estado == 6) {
178. ctx.fillStyle = *"#fce683"*;
179. } else if (b.estado == 5) {
180. ctx.fillStyle = *"#fceb9c"*;
181. } else if (b.estado == 4) {
182. ctx.fillStyle = *"#fdf0b4"*;
183. } else if (b.estado == 3) {
184. ctx.fillStyle = *"#fef5cd"*;
185. } else if (b.estado == 2) {
186. ctx.fillStyle = *"#fefae6"*;
187. } else if (b.estado == 1) {
188. ctx.fillStyle = *"transparent"*;
189. }
190. ctx.fill();
191. ctx.closePath();
192. }
193. }
194. }
195. */\* Si mover deracha es falso o las vidas son impares entonces el vacio izquierdo de los ladrillos se disminuira hasta que esta variable tenga un valor de 50, cuando llegue a 50 la variable booleana valdra true, cuando las vidas sean pares entonces los ladrillos se moveran a la derecha hasta que tengan una distancia de 50 del margen derecho del CANVAS\*/*
196. if (!moverDerecha || vidas % 2 !== 0) {
197. if (vacioIzquierdoLadrillos >= 50) {
198. vacioIzquierdoLadrillos -= 10;
199. }
200. else {
201. moverDerecha = true;
202. }
203. } else {
204. if (vacioIzquierdoLadrillos + filas \* (anchoLadrillo + rellenoLadrillos) - 50 <= canvas.width - 100) {
205. vacioIzquierdoLadrillos += 10;
206. }
207. }
208. }
209. *// Detectar la colision*
210. function detectarColision() {
211. for (var c = 0; c < numColumnasLadrillos; c++) {
212. for (var f = 0; f < numFilasLadrillos; f++) {
213. var b = ladrillos[c][f];
214. function sumarPuntaje() {
215. b.estado--;
216. b.ciclo--;
217. puntaje++;
218. if (puntaje == numFilasLadrillos \* numColumnasLadrillos) {
219. alert(*"FELICITACIONES: HAS GANADO"*);
220. document.location.reload();
221. }
222. }
223. if (b.estado == 10) {
224. *// se modifica si la bola llega al ladrillo*
225. if (x > b.x && x < b.x + anchoLadrillo + ballRadius / 2 && y > b.y && y < b.y + alturaLadrillo + ballRadius / 2) {
226. dy = -dy;
227. if ((Math.round(Math.random() \* 100)) % 2) {
228. sumarPuntaje();*// Llama a la funcion para reducir el ciclo, el estado y sumar el puntaje*
229. }
230. }
231. if (segundaBola) {
232. if (x2 > b.x && x2 < b.x + anchoLadrillo && y2 > b.y && y2 < b.y + alturaLadrillo) {
233. dy2 = -dy2;
234. if ((Math.round(Math.random() \* 100)) % 2) {
235. sumarPuntaje();*// Llama a la funcion para reducir el ciclo, el estado y sumar el puntaje*
236. }
237. }
238. }
239. */\* Si se dispara el misil entonces se debe evaluar si este toca un ladrillo, de ser asi entonces se mostrara en la pantalla la imagen de la explosion con las coordenas del ladrillo, ademas la poscion y la variable bool correspondiente al misil volveran a sus estados originales para que el misil pueda ser lanzado de nuevo lanzado de nuevo \*/*
240. if(posX\_Misil+20 > b.x && posX\_Misil+20 < b.x + anchoLadrillo && posY\_Misil > b.y && posY\_Misil < b.y + alturaLadrillo){
241. ctx.drawImage(explosion,  b.x, b.y);
242. dispararMisil = false;
243. posY\_Misil = canvas.width/2;
244. sumarPuntaje();
245. }
246. }
247. }
248. }
249. }
250. */\* La funcion permitira dibujar el puntaje y la cantidad de misiles \*/*
251. function dibujarPuntos\_Misiles() {
252. ctx.font = *"24px Poppins"*;
253. ctx.fillStyle = colorTexto;
254. ctx.fillText(*"Puntaje:* *"* + puntaje + *"    |    Misiles:* *"* + cantidadMisiles, 8, 20);
255. }
256. function dibujarVidas() {
257. ctx.font = *"24px Poppins"*;
258. ctx.fillStyle = colorTexto;
259. *// SE MUESTRA EL NÚMERO DE VIDAS DISPONIBLES*
260. ctx.fillText(*"Vidas:* *"* + vidas, canvas.width - 100, 20);
261. }
262. function quitarVidas() {
263. vidas--;
264. if (!vidas) {
265. alert(*"GAME OVER"*);
266. document.location.reload();
267. } else {
268. x = canvas.width / 2;
269. y = canvas.height - 30;
270. x2 = canvas.width / 2;
271. y2 = canvas.height - 30;
272. dx = 5;
273. dy = -5;
274. dx2 = -1.5;
275. dy2 = -1.5;
276. paletaPosX = (canvas.width - anchuraPaleta) / 2;
277. }
278. }
279. function dibujar() {
280. ctx.clearRect(0, 0, canvas.width, canvas.height);
281. dibujarPuntos\_Misiles();
282. dibujarVidas();
283. dibujarLadrillos();
284. dibujarBola(x, y, ballRadius, colorBola);
285. if (segundaBola) {
286. dibujarBola(x2, y2, ballRadius - 2, colorBola2);
287. }
288. dibujarPaleta();
289. detectarColision();
290. *// Rebote de la bola*
291. if (x + dx > canvas.width - ballRadius || x + dx < ballRadius) {
292. dx = -dx;
293. colorLadrillos = false;*// Cambia el color de los ladrillos si la bola principal rebota en x*
294. }
295. if (y + dy < ballRadius) {
296. dy = -dy;
297. var cambia = document.querySelector(*"canvas"*);
298. cambia.classList.toggle(*"activado"*);
299. colorLadrillos = true;
300. }
301. if (x2 + dx2 > canvas.width - ballRadius || x2 + dx2 < ballRadius) {
302. dx2 = -dx2;
303. }
304. if (y2 + dy2 < ballRadius) {
305. dy2 = -dy2;
306. }
307. *// Rebotar en Paleta o Game Over*
308. if (y + dy > canvas.height - ballRadius) {
309. if (x > paletaPosX && x < paletaPosX + anchuraPaleta + ballRadius / 2) {
310. dy = -dy;
311. } else {
312. quitarVidas();
313. }
314. }
315. if (segundaBola) {
316. if (y2 + dy2 > canvas.height - ballRadius) {
317. if (x2 > paletaPosX && x2 < paletaPosX + anchuraPaleta) {
318. dy2 = -dy2;
319. } else {
320. quitarVidas();
321. }
322. }
323. }
324. *// Movimiento Paleta*
325. if (flechaDerechaPulsada && paletaPosX < canvas.width - anchuraPaleta) {
326. paletaPosX += 15;
327. } else if (flechaIzquierdaPulsada && paletaPosX > 0) {
328. paletaPosX -= 15;
329. }
330. *// Movimiento del misil*
331. */\* Si se pulsa la tecla espacio entonces la dispararMisil vale true, por lo tanto se llama a la funcion que dibuja el misli y como parametro que le dejamos su coordena x, si recordamos cuando se pulsa la tecla escpacio la posicion de x del misil varldra a la posicion de la mitad del canvas*
333. *Restamos Y para que la imagen se mueva hacia arriba*
334. *si la imagen alcanza la parte superior del CANVAS entonces se dibuja la explosion y se actualizan las variables del misil para que pueda ser disparado de nuevo*
335. *\*/*
336. if(dispararMisil){
337. misil(posX\_Misil);
338. posY\_Misil -= 5;
339. }
340. if(posY\_Misil == 0){
341. ctx.drawImage(explosion, posX\_Misil-50, posY\_Misil)
342. dispararMisil = false;
343. posY\_Misil = canvas.height;
344. }
345. *// Se incrementan los valores de la pocion de las bolas*
346. x += dx;
347. y += dy;
348. x2 += dx2;
349. y2 += dy2;
350. requestAnimationFrame(dibujar);
351. }
352. dibujar();
353. </script>
354. </body>
355. </html>

El resultado es el siguiente:



**Gráfica 16. Interfaz cuando se presiona la tecla espacio.**

****

**Gráfica 17. Interfaz cuando el misil golpea un ladrillo.**

# FASE 11: Fin, menú, ganar y perder

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

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

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

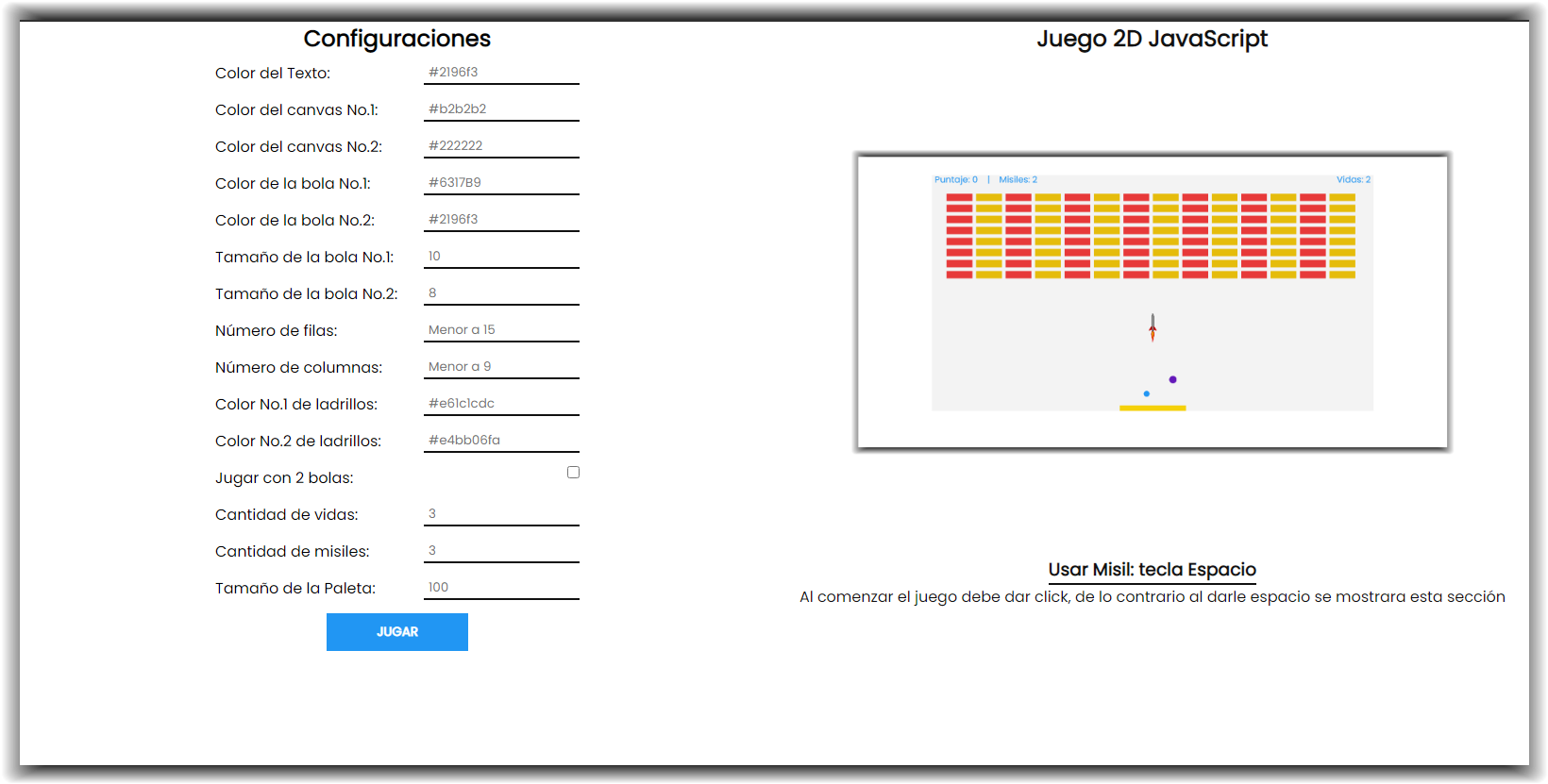
En el código CSS hacemos todas las configuraciones y como podemos observar el canvas no se ve ya que lo movemos a la iquierda esto se consigue con left:-100%, esto permite no verse en la página, también lo hacemos con los otros elementos. En el código JavaScript veremos como se mueven los lementos esto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

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

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

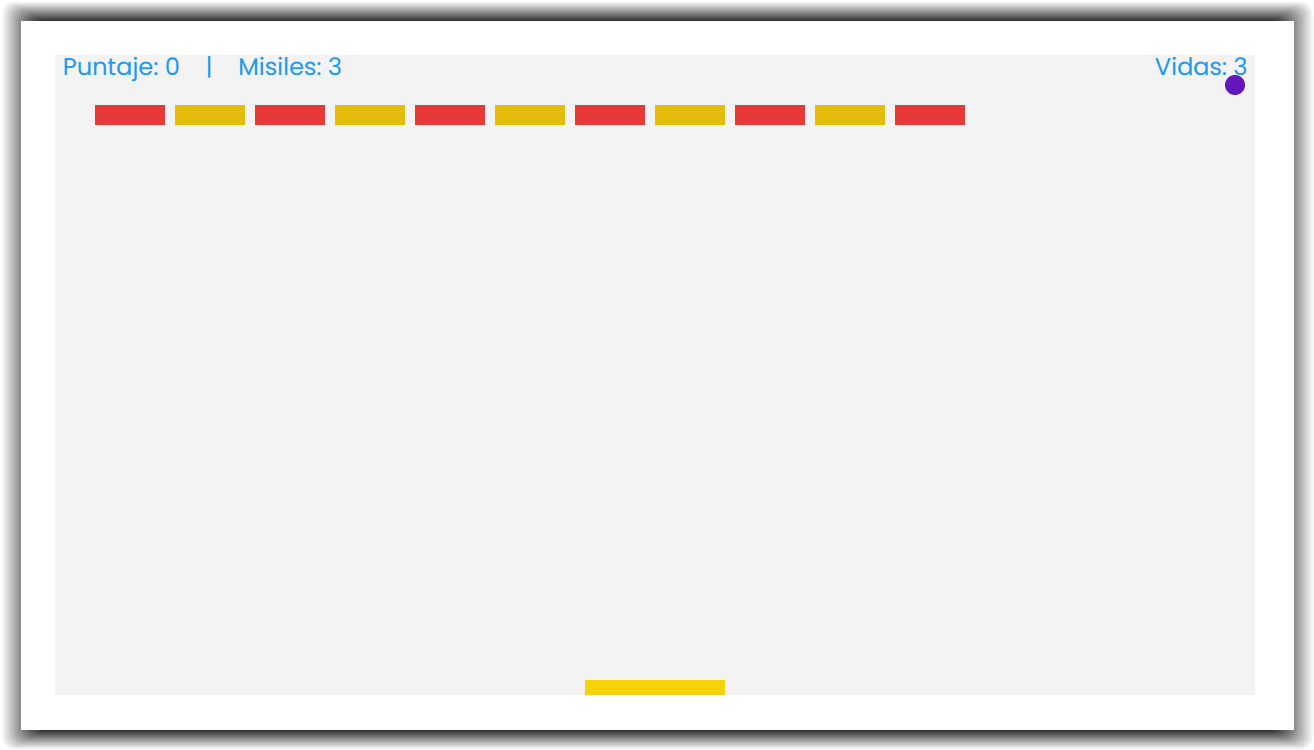
1. <!DOCTYPE html>
2. <html>
3. <head>
4. <meta charset=*"UTF-8"* />
5. <title>Juego 2D</title>
6. <link rel=*"stylesheet"* href=*"style.css"*>
7. </head>
8. <body>
9. <div class=*"inicio"*>
10. <div class=*"menu"*>
11. <h2>Configuraciones</h2>
12. <form action=*""*>
13. <ul>
14. <li>
15. Color del Texto:
16. <input id=*"cT"* type=*"text"* placeholder=*"#2196f3"* />
17. </li>
18. </ul>
19. <ul>
20. <li>
21. Color del canvas No.1:
22. <input id=*"cC1"* type=*"text"* placeholder=*"#b2b2b2"* />
23. </li>
24. <li>
25. Color del canvas No.2:
26. <input id=*"cC2"* type=*"text"* placeholder=*"#222222"* />
27. </li>
28. </ul>
29. <ul>
30. <li>
31. Color de la bola No.1:
32. <input id=*"cB1"* type=*"text"* placeholder=*"#6317B9"* />
33. </li>
34. <li>
35. Color de la bola No.2:
36. <input id=*"cB2"* type=*"text"* placeholder=*"#2196f3"* />
37. </li>
38. </ul>
39. <ul>
40. <li>
41. Tamaño de la bola No.1:
42. <input id=*"tB1"* type=*"text"* placeholder=*"10"* />
43. </li>
44. <li>
45. Tamaño de la bola No.2:
46. <input id=*"tB2"* type=*"text"* placeholder=*"8"* />
47. </li>
48. </ul>
49. <ul>
50. <li>
51. Número de filas:
52. <input id=*"numFilas"* type=*"text"* placeholder=*"Menor a 15"* />
53. </li>
54. <li>
55. Número de columnas:
56. <input id=*"numColumnas"* type=*"text"* placeholder=*"Menor a 9"* />
57. </li>
58. </ul>
59. <ul>
60. <li>
61. Color No.1 de ladrillos:
62. <input id=*"cL1"* type=*"text"* placeholder=*"#e61c1cdc"* />
63. </li>
64. <li>
65. Color No.2 de ladrillos:
66. <input id=*"cL2"* type=*"text"* placeholder=*"#e4bb06fa"* />
67. </li>
68. </ul>
69. <ul>
70. <li>
71. Jugar con 2 bolas:
72. <input id=*"check"* type=*"checkbox"* value=*"2"* />
73. </li>
74. <li>
75. Cantidad de vidas:
76. <input id=*"vidas"* type=*"text"* placeholder=*"3"* />
77. </li>
78. </ul>
79. <ul>
80. <li>
81. Cantidad de misiles:
82. <input id=*"misiles"* type=*"text"* placeholder=*"3"* />
83. </li>
84. <li>
85. Tamaño de la Paleta:
86. <input id=*"paleta"* type=*"text"* placeholder=*"100"* />
87. </li>
88. </ul>
89. <ul>
90. <li>
91. <input id=*"boton"* type=*"submit"* value=*"JUGAR"* onclick=*"jugar()"* />
92. </li>
93. </ul>
94. </form>
95. </div>
96. <div class=*"presentacion"*>
97. <h2>Juego 2D JavaScript</h2>
98. <img src=*"img/foto.PNG"* alt=*"juego 2D"* />
99. <h3>Usar Misil: tecla Espacio</h3>
100. <p>
101. Al comenzar el juego debe dar click, de lo contrario al darle espacio
102. se mostrara esta sección
103. </p>
104. </div>
105. </div>
106. <canvas id=*"miCanvas"* width=*"1200"* height=*"640"*> </canvas>
107. <div class=*"ganar"*>
108. <img src=*"img/ganar.gif"* alt=*""*>
109. <input class=*"volver"* type=*"submit"* value=*"VOLVER"* onclick=*"*document*.location.reload()"*/>
110. </div>
111. <div class=*"perder"*>
112. <img src=*"img/perder.gif"* alt=*""*>
113. <input class=*"volver"* type=*"submit"* value=*"VOLVER"* onclick=*"*document*.location.reload()"*/>
114. </div>
115. <script src=*"script.js"*></script>
116. </body>
117. </html>
118. @import url(*"https://fonts.googleapis.com/css2?family=Poppins&display=swap"*);
119. \* {
120. font-family: *"Poppins"*, sans-serif;
121. padding: 0;
122. margin: 0;
123. }
124. canvas {
125. transition: 0.5s;
126. display: block;
127. cursor: none;
128. position: fixed;
129. right: -100%;
130. transition: 1s;
131. }
132. canvas.moverCanvas {
133. position: relative;
134. right: 0;
135. margin: 50px auto 0 auto;
136. }
137. */\* Flex nos permite configurar el tipo de contenedor del elemento*
138. *justify-content: space-between; nos permite poner un elemento junto al otro*
139. *align-items: center; nos permite centrar el contenido*
140. *\*/*
141. .inicio {
142. width: 100%;
143. height: 100vh;
144. display: flex;
145. position: fixed;
146. justify-content: space-between;
147. align-items: center;
148. text-align: center;
149. left: 0;
150. transition: 1s;
151. }
152. .inicio .menu {
153. display: block;
154. width: 50%;
155. height: 100vh;
156. }
157. .inicio .menu form {
158. display: block;
159. margin: 0 auto;
160. }
161. .inicio form ul {
162. list-style: none;
163. width: 50%;
164. margin: auto;
165. }
166. .inicio form ul li {
167. padding: 7px;
168. justify-content: space-between;
169. display: flex;
170. }
171. .inicio form ul li input[type=*"text"*] {
172. border: none;
173. padding: 0 5px;
174. border-bottom: solid 2px #111111;
175. }
176. #boton, .volver{
177. background-color: #2196f3;
178. color: #ffffff;
179. display: block;
180. font-weight: 600;
181. border: none;
182. padding: 10px;
183. margin: auto;
184. width: 150px;
185. }
186. */\* Hover nos permite modificar un elemento cuando el curso se encuentra encima del elemento \*/*
187. #boton:hover,.volver:hover {
188. background-color: #f32f21d2;
189. cursor: pointer;
190. }
191. .inicio .presentacion {
192. display: block;
193. margin: auto;
194. width: 50%;
195. height: 100vh;
196. text-align: center;
197. color: #111111;
198. }
199. .inicio .presentacion h3 {
200. border-bottom: solid 2px #111111;
201. display: inline-block;
202. }
203. .inicio .presentacion img {
204. width: 80%;
205. padding: 100px 0;
206. }
207. */\* .mover es una clase que nos permitira mover el elemento esto lo logramos con javascript \*/*
208. .inicio.mover {
209. left: -100%;
210. }
211. .ganar, .perder {
212. width: 100%;
213. height: 100vh;
214. background-color: #ffffff;
215. display: block;
216. margin: auto;
217. position: fixed;
218. left: -100%;
219. transition: 1s;
220. }
221. .ganar.mover, .perder.mover {
222. left: 0;
223. }
224. .ganar img, .perder img{
225. display: block;
226. margin: 100px auto 0 auto;
227. box-shadow: 20px 20px 10px #333333;
228. }
229. .volver{
230. margin-top: 30px;
231. }
232. var canvas = document.getElementById(*"miCanvas"*);
233. var ctx = canvas.getContext(*"2d"*);
234. var ballRadius = 10;
235. var x = canvas.width / 2;
236. var y = canvas.height - 30;
237. var dx = 5;
238. var dy = -5;
239. var ballRadius2 = 8;
240. var x2 = x;
241. var y2 = y;
242. var dx2 = -1.5;
243. var dy2 = -1.5;
244. var segundaBola = false;
245. if (Math.random() \* 100 > 50) {
246. segundaBola = true;
247. }
248. *// Paleta*
249. var alturaPaleta = 15;
250. var anchuraPaleta = 140;
251. if (segundaBola) {
252. anchuraPaleta = 180;
253. }
254. var paletaPosX = (canvas.width - anchuraPaleta) / 2;
255. var flechaDerechaPulsada = false;
256. var flechaIzquierdaPulsada = false;
257. var numFilasLadrillos = 0;
258. var numColumnasLadrillos = 0;
259. do {
260. numFilasLadrillos = Math.round(Math.random() \* 100);
261. numColumnasLadrillos = Math.round(Math.random() \* 10);
262. } while (numFilasLadrillos < 1 || numFilasLadrillos > 14 || numColumnasLadrillos < 1 || numColumnasLadrillos > 8);
263. var anchoLadrillo = 70;
264. var alturaLadrillo = 20;
265. var rellenoLadrillos = 10;
266. var vacioSuperiorLadrillos = 50;
267. var vacioIzquierdoLadrillos = canvas.width;
268. var puntaje = 0;
269. *// Misil*
270. var imgMisil = new Image();
271. imgMisil.src = *"img/misil.png"*;
272. var posY\_Misil = canvas.height;
273. var posX\_Misil = 0;
274. var explosion = new Image();
275. explosion.src = *"img/explosion.png"*;
276. function misil(x) {
277. ctx.drawImage(imgMisil, x, posY\_Misil);
278. }
279. *// Caracteristicas de los elementos*
280. var dispararMisil = false;
281. var cantidadMisiles = 3;
282. var colorCanvas1 = *"#f2f2f2f2"*;
283. canvas.style.backgroundColor = colorCanvas1;
284. var colorCanvas2 = *"#222222"*;
285. var colorFigura = *"#ff0000"*;
286. var colorBola = *"#6317B9"*;
287. var colorBola2 = *"#2196f3"*;
288. var colorPaleta = *"#f5d109"*;
289. var colorLadrillo = *"#e61c1cdc"*;
290. var colorLadrillo2 = *"#e4bb06fa"*;
291. var colorTexto = *"#2196f3"*;
292. var vidas = 3;
293. var ladrillos = [];
294. for (var columna = 0; columna < numColumnasLadrillos; columna++) {
295. ladrillos[columna] = [];
296. for (var fila = 0; fila < numFilasLadrillos; fila++) {
297. ladrillos[columna][fila] = { *x*: 0, *y*: 0, *estado*: 10, *ciclo*: 10 };
298. }
299. }
300. document.addEventListener(*"keydown"*, teclaPresionada, false);
301. document.addEventListener(*"keyup"*, teclaLiberada, false);
302. document.addEventListener(*"mousemove"*, manejadorRaton, false);
303. function teclaPresionada(e) {
304. if (e.keyCode == 39) {
305. flechaDerechaPulsada = true;
306. } else if (e.keyCode == 37) {
307. flechaIzquierdaPulsada = true;
308. } else if (e.keyCode == 32) {
309. if (!dispararMisil && cantidadMisiles > 0) {
310. posX\_Misil = paletaPosX + anchuraPaleta / 2 - 20;
311. dispararMisil = true;
312. cantidadMisiles--;
313. }
314. }
315. }
316. function teclaLiberada(e) {
317. if (e.keyCode == 39) {
318. flechaDerechaPulsada = false;
319. } else if (e.keyCode == 37) {
320. flechaIzquierdaPulsada = false;
321. }
322. }
323. function manejadorRaton(e) {
324. var posXRatonDentroDeCanvas = e.clientX - canvas.offsetLeft;
325. if (posXRatonDentroDeCanvas > anchuraPaleta / 2 && posXRatonDentroDeCanvas < canvas.width - anchuraPaleta / 2) {
326. paletaPosX = posXRatonDentroDeCanvas - anchuraPaleta / 2;
327. }
328. }
329. function dibujarBola(cordernaX, cordenaY, tamaño, color) {
330. ctx.beginPath();
331. ctx.arc(cordernaX, cordenaY, tamaño, 0, Math.PI \* 2);
332. ctx.fillStyle = color;
333. ctx.fill();
334. ctx.closePath();
335. }
336. function dibujarPaleta() {
337. ctx.beginPath();
338. ctx.rect(paletaPosX, canvas.height - alturaPaleta, anchuraPaleta, alturaPaleta);
339. ctx.fillStyle = colorPaleta;
340. ctx.fill();
341. ctx.closePath();
342. }
343. var colorLadrillos = false;
344. var moverDerecha = false;
345. function dibujarLadrillos() {
346. for (var columnas = 0; columnas < numColumnasLadrillos; columnas++) {
347. for (var filas = 0; filas < numFilasLadrillos; filas++) {
348. if (ladrillos[columnas][filas].estado > 0) {
349. var b = ladrillos[columnas][filas];
350. var brickX = filas \* (anchoLadrillo + rellenoLadrillos) + vacioIzquierdoLadrillos;
351. var brickY = columnas \* (alturaLadrillo + rellenoLadrillos) + vacioSuperiorLadrillos;
352. if (b.estado < 10 && b.estado > 1) {
353. b.ciclo--;
354. if (b.ciclo == 0) {
355. if (b.estado > 1) {
356. b.estado--;
357. b.ciclo = 10;
358. }
359. }
360. }
361. ladrillos[columnas][filas].x = brickX;
362. ladrillos[columnas][filas].y = brickY;
363. ctx.beginPath();
364. ctx.rect(brickX, brickY, anchoLadrillo, alturaLadrillo);
365. if (b.estado == 10) {
366. if (colorLadrillos) {
367. if (filas % 2) {
368. ctx.fillStyle = colorLadrillo;
369. } else {
370. ctx.fillStyle = colorLadrillo2;
371. }
372. } else {
373. if (filas % 2) {
374. ctx.fillStyle = colorLadrillo2;
375. } else {
376. ctx.fillStyle = colorLadrillo;
377. }
378. }
379. } else if (b.estado == 9) {
380. ctx.fillStyle = *"#fad638"*;
381. } else if (b.estado == 8) {
382. ctx.fillStyle = *"#fbdb51"*;
383. } else if (b.estado == 7) {
384. ctx.fillStyle = *"#fbe16a"*;
385. } else if (b.estado == 6) {
386. ctx.fillStyle = *"#fce683"*;
387. } else if (b.estado == 5) {
388. ctx.fillStyle = *"#fceb9c"*;
389. } else if (b.estado == 4) {
390. ctx.fillStyle = *"#fdf0b4"*;
391. } else if (b.estado == 3) {
392. ctx.fillStyle = *"#fef5cd"*;
393. } else if (b.estado == 2) {
394. ctx.fillStyle = *"#fefae6"*;
395. } else if (b.estado == 1) {
396. ctx.fillStyle = *"transparent"*;
397. }
398. ctx.fill();
399. ctx.closePath();
400. }
401. }
402. }
403. if (!moverDerecha || vidas % 2 !== 0) {
404. if (vacioIzquierdoLadrillos >= 50) {
405. vacioIzquierdoLadrillos -= 10;
406. } else {
407. moverDerecha = true;
408. }
409. } else {
410. if (vacioIzquierdoLadrillos + filas \* (anchoLadrillo + rellenoLadrillos) - 50 <= canvas.width - 100) {
411. vacioIzquierdoLadrillos += 10;
412. }
413. }
414. }
415. *// Detectar la colision*
416. function detectarColision() {
417. for (var c = 0; c < numColumnasLadrillos; c++) {
418. for (var f = 0; f < numFilasLadrillos; f++) {
419. var b = ladrillos[c][f];
420. function sumarPuntaje() {
421. b.estado--;
422. b.ciclo--;
423. puntaje++;
424. if (puntaje == numFilasLadrillos \* numColumnasLadrillos) {
425. */\* Si se gana entonces agragamos una clase al canvas esto para moverlo, o desaparezca de la pagina, pero agregamos una clase llamada (mover) al elemento que tiene una otra clase definida llamada "ganar" esto para que cuando se active la mover permita mostrar este elemento en la pagina y clickJugar se vuelve false para que no se siga dibujando el canvas \*/*
426. document.getElementById(*"miCanvas"*).classList.toggle(*"moverCanvas"*);
427. document.querySelector(*".ganar"*).classList.toggle(*"mover"*);
428. clickJugar = false;
429. }
430. }
431. if (b.estado == 10) {
432. *// se modifica si la bola llega al ladrillo*
433. if (x > b.x && x < b.x + anchoLadrillo + ballRadius / 2 && y > b.y && y < b.y + alturaLadrillo + ballRadius / 2) {
434. dy = -dy;
435. if (Math.round(Math.random() \* 100) % 2) {
436. sumarPuntaje();
437. }
438. }
439. if (segundaBola) {
440. if (x2 > b.x && x2 < b.x + anchoLadrillo && y2 > b.y && y2 < b.y + alturaLadrillo) {
441. dy2 = -dy2;
442. if (Math.round(Math.random() \* 100) % 2) {
443. sumarPuntaje();
444. }
445. }
446. }
447. if (posX\_Misil + 20 > b.x && posX\_Misil + 20 < b.x + anchoLadrillo && posY\_Misil > b.y && posY\_Misil < b.y + alturaLadrillo) {
448. ctx.drawImage(explosion, b.x, b.y);
449. dispararMisil = false;
450. posY\_Misil = canvas.width / 2;
451. sumarPuntaje();
452. }
453. }
454. }
455. }
456. }
457. function dibujarPuntos\_Misiles() {
458. ctx.font = *"24px Poppins"*;
459. ctx.fillStyle = colorTexto;
460. ctx.fillText(*"Puntaje:* *"* + puntaje + *"    |    Misiles:* *"* + cantidadMisiles, 8, 20 );
461. }
462. function dibujarVidas() {
463. ctx.font = *"24px Poppins"*;
464. ctx.fillStyle = colorTexto;
465. ctx.fillText(*"Vidas:* *"* + vidas, canvas.width - 100, 20);
466. }
467. function quitarVidas() {
468. vidas--;
469. if (!vidas) {
470. */\* Si el numero de vidas llega a (0) cero, entonces debemos mostrar elemento que corresponde a perde, ocuultamos nuestro canvas con ayuda de CSS y mostramos esta seccion ademas detenemos el dibujo del canvas \*/*
471. document.getElementById(*"miCanvas"*).classList.toggle(*"moverCanvas"*);
472. document.querySelector(*".perder"*).classList.toggle(*"mover"*);
473. clickJugar = false;
474. } else {
475. x = canvas.width / 2;
476. y = canvas.height - 30;
477. x2 = canvas.width / 2;
478. y2 = canvas.height - 30;
479. dx = 5;
480. dy = -5;
481. dx2 = -1.5;
482. dy2 = -1.5;
483. paletaPosX = (canvas.width - anchuraPaleta) / 2;
484. }
485. }
486. */\* Como tenemos varios elementos tipo texto los cuales debemos conocer su valor y estos tiene como identificador un id entonces creamos una funcion para no escribir  mas de diez veces document.getElementById(documento).value;  \*/*
487. function valor(documento) {
488. var entregarValor = document.getElementById(documento);
489. return entregarValor.value;
490. }
491. */\* como debemos comprobar el color o el texto que el usuario introduzca o si no generara un error, entonces se debe verificar lo que el usuario a introducido.*
492. *Tenemos que verificar si el color es RGB lo cual este tipo de colores tiene como predeterminado comenzar con un numeral (#) lo cual es obligatorio que sea el primer caracter de lo introducido tambien este debe tener 4 o 7 caracteres ejemplo (#111) o (#111111) son el mismo color lo cual uno tiene 4 y el otro tiene 7 caracteres*
493. *si el color es verdadero, debebmos ver el id de este del texto que corresponde a un elemento HTML y verificamos dependiendo de esto alacenamos lo introducido por el usuario en uno de los tipos de elemntos en el canvas, o mejor dicho lo almacenamos en una variable, como color de canvas, color de bola etc.*
494. *\*/*
495. function comprobarColor(elemento, variable) {
496. if ((valor(elemento).length == 4 || valor(elemento).length == 7) && valor(elemento).charAt(0) == *"#"*) {
497. if (elemento === *"cT"*) {
498. colorTexto = valor(elemento);
499. } else if (elemento === *"cC1"*) {
500. colorCanvas1 = valor(elemento);
501. } else if (elemento === *"cC2"*) {
502. colorCanvas2 = valor(elemento);
503. } else if (elemento === *"cB1"*) {
504. colorBola = valor(elemento);
505. } else if (elemento === *"cB2"*) {
506. colorBola2 = valor(elemento);
507. } else if (elemento === *"cL1"*) {
508. colorLadrillo = valor(elemento);
509. } else if (elemento === *"cL2"*) {
510. colorLadrillo2 = valor(elemento);
511. }
512. } else if (valor(elemento).length !== 0) {
513. alert(*"Introduce bien los valores."*);
514. }
515. }
516. */\* Creamos esta variable ya que si el se gana el canvas se seguira dibujando lo cual no queremos que pase eso, ya que como elcanvas se seguira dibujando seguira perdiendo vidas por consiguiente se mostrara la ventana de ganar y perder consecutivamente \*/*
517. var clickJugar = false;
518. */\* En la siguiente funcion nos permitira comenzar el juego*
519. *primero al darle click a un boton este por lo general recargar nuevamente la pagina lo cual la primera instruccion es para evitar esto event.preventDefault();*
520. *lo siguinete es mostrar el canvas lo cual al elemnto cuya clase es inicio le agrega una segunda clase llamada mover, tambien pasa con el canvas, esto para permitir mover las secciones con CSS*
521. *Debemos comprobar los colores como los de los ladrillos etc, lo cual llamamos la funcion definida anteriormente*
522. *Ahora debemos comprobar el valor para las caracteristicas de la bola, la cantidad de ladrillos etc. Como debemos poner datos numericos debemos convertir el texto en numeros, esto lo logramos con parseFloat y parseInt, y asi dependiento del elemento donde se introdujo el texto se agregara a las caracteristicas o variables que definimo al pricipio del codigo numfilasLadrillos, ballradius etc*
523. *\*/*
524. function jugar() {
525. event.preventDefault();
526. document.querySelector(*".inicio"*).classList.toggle(*"mover"*);
527. document.getElementById(*"miCanvas"*).classList.toggle(*"moverCanvas"*);
528. comprobarColor(*"cT"*);
529. comprobarColor(*"cC1"*);
530. comprobarColor(*"cC2"*);
531. comprobarColor(*"cB1"*);
532. comprobarColor(*"cB2"*);
533. comprobarColor(*"cL1"*);
534. comprobarColor(*"cL2"*);
535. if (parseFloat(valor(*"tB1"*)) > 0 && parseFloat(valor(*"tB1"*)) <= 40) {
536. ballRadius = parseFloat(valor(*"tB1"*));
537. }
539. if (parseFloat(valor(*"tB2"*)) > 0 && parseFloat(valor(*"tB2"*)) <= 40) {
540. ballRadius2 = parseFloat(valor(*"tB2"*));
541. }
542. if (parseInt(valor(*"numFilas"*)) > 0 && parseInt(valor(*"numFilas"*)) <= 14) {
543. numFilasLadrillos = parseInt(valor(*"numFilas"*));
544. }
546. if (parseInt(valor(*"numColumnas"*)) > 0 && parseInt(valor(*"numColumnas"*)) <= 8) {
547. numColumnasLadrillos = parseInt(valor(*"numColumnas"*));
548. }
549. */\* Como tenemos un checkBox debemos saber si este se encuentra seleccionado para ver si dibujamos una segunda bola \*/*
550. if (document.getElementById(*"check"*).checked) {
551. segundaBola = true;
552. }
553. if (parseInt(valor(*"vidas"*)) > 0) {
554. vidas = parseInt(valor(*"vidas"*));
555. }
556. if (parseInt(valor(*"misiles"*)) > 0) {
557. cantidadMisiles = parseInt(valor(*"misiles"*));
558. }
560. if (parseFloat(valor(*"paleta"*)) > 0) {
561. anchuraPaleta = parseFloat(valor(*"paleta"*));
562. }
563. */\* Cambiamos el valor de la variable y llamamos a la funcion dibujar() para que comience el juego \*/*
564. clickJugar = true;
565. dibujar();
566. }
567. function dibujar() {
568. ctx.clearRect(0, 0, canvas.width, canvas.height);
569. dibujarPuntos\_Misiles();
570. dibujarVidas();
571. dibujarLadrillos();
572. dibujarBola(x, y, ballRadius, colorBola);
573. if (segundaBola) {
574. dibujarBola(x2, y2, ballRadius2, colorBola2);
575. }
576. dibujarPaleta();
577. detectarColision();
578. *// Rebote de la bola*
579. if (x + dx > canvas.width - ballRadius || x + dx < ballRadius) {
580. dx = -dx;
581. colorLadrillos = false; *// Cambia el color de los ladrillos si la bola principal rebota en x*
582. }
583. if (y + dy < ballRadius) {
584. dy = -dy;
585. canvas.style.backgroundColor = colorCanvas2;
586. colorLadrillos = true;
587. }
588. if (x2 + dx2 > canvas.width - ballRadius2 || x2 + dx2 < ballRadius2) {
589. dx2 = -dx2;
590. }
591. if (y2 + dy2 < ballRadius2) {
592. dy2 = -dy2;
593. }
594. *// Rebotar en Paleta o Game Over*
595. if (y + dy > canvas.height - ballRadius) {
596. if (x > paletaPosX && x < paletaPosX + anchuraPaleta + ballRadius / 2) {
597. dy = -dy;
598. canvas.style.backgroundColor = colorCanvas1;
599. } else {
600. quitarVidas();
601. }
602. }
603. if (segundaBola) {
604. if (y2 + dy2 > canvas.height - ballRadius2) {
605. if (x2 > paletaPosX && x2 < paletaPosX + anchuraPaleta) {
606. dy2 = -dy2;
607. } else {
608. quitarVidas();
609. }
610. }
611. }
612. *// Movimiento Paleta*
613. if (flechaDerechaPulsada && paletaPosX < canvas.width - anchuraPaleta) {
614. paletaPosX += 15;
615. } else if (flechaIzquierdaPulsada && paletaPosX > 0) {
616. paletaPosX -= 15;
617. }
618. *// Movimiento del misil*
619. if (dispararMisil) {
620. misil(posX\_Misil);
621. posY\_Misil -= 5;
622. }
623. if (posY\_Misil == 0) {
624. ctx.drawImage(explosion, posX\_Misil - 50, posY\_Misil);
625. dispararMisil = false;
626. posY\_Misil = canvas.height;
627. }
628. *// Se incrementan los valores de la pocion de las bolas*
629. x += dx;
630. y += dy;
631. x2 += dx2;
632. y2 += dy2;
633. */\* Si  clickJugar es true entonce el requestAnimationFrame comenzara a ejecutar la funcion dibujar \*/*
634. if(clickJugar){
635. requestAnimationFrame(dibujar);
636. }
637. }

El resultado es el siguiente:



**Gráfica 18. Interfaz inicial de la página.**

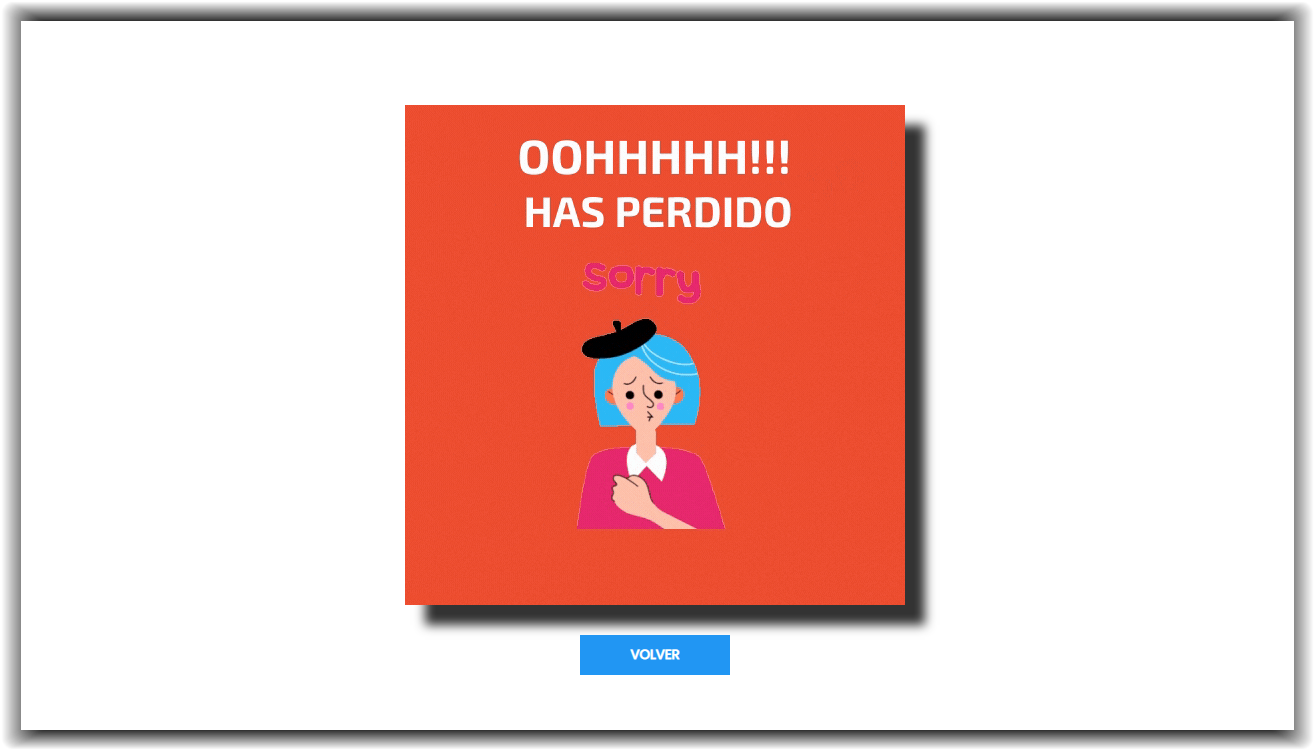
En la gráfica anterior podemos observar el resultado tras cargar la página el menu que creamos con los lementos HTML y los estilos en CSS.



**Gráfica 19. Interfaz cuando le damos al botón JUGAR.**

****

**Gráfica 20. Interfaz cuando eliminamos todos los ladrillos.**

****

**Gráfica 21. Interfaz cuando agotamos todas las vidas.**

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

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

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

# CONCLUSIONES

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

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

# BIBLIOGRAFÍA

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