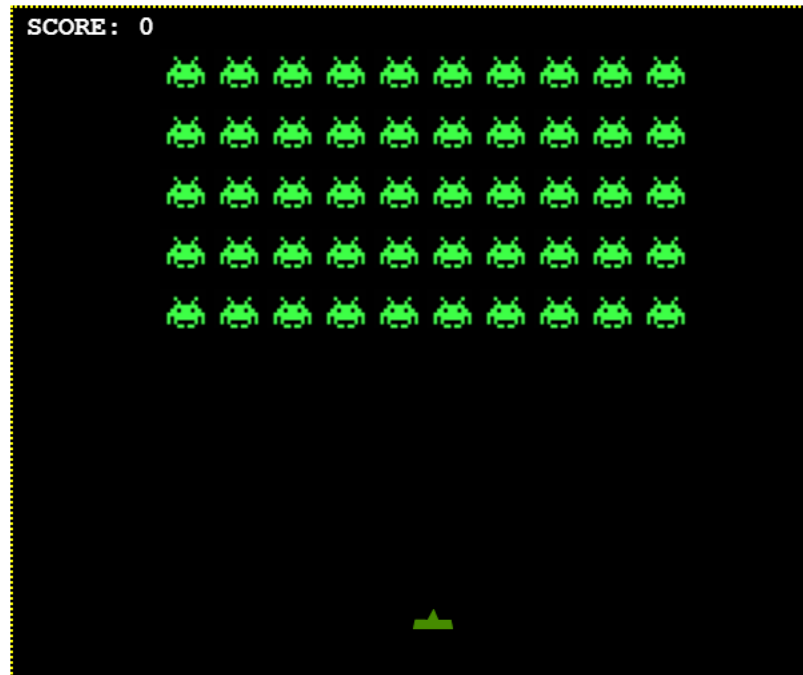


# Space Invaders



## Animación Base

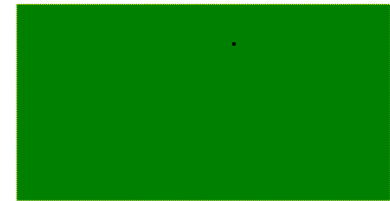
```

1  <!doctype html>
2  <html>
3  <head>
4    <meta charset="UTF-8">
5    <title>CANVAS</title>
6  </head>
7  <body>
8    <div>
9      <div>
10       <div>
11         <div>
12           <div>
13             <div>
14               <div>
15                 <div>
16                   <div>
17                     <div>
18                       <div>
19                         <div>
20                           <div>
21                             <div>
22                               <div>
23                                 <div>
24                                   <div>
25                                     <div>
26                                       <div>
27                                         <div>
28                                           <div>
29                                             <div>
30                                               <div>
31                                                 <div>
32                                                   <div>
33                                                     <div>
34                                                       <div>
35                                                         <div>
36                                                           <div>
37                                                             <div>
38                                                               <div>
39                                                                 <div>
40                                                                  <div>
41                                                                   <div>
42                                                                    <div>
43                                                                     <div>
44                                                                      <div>
45                                                                       <div>
46                                                                        <div>
47                                                                         <div>
48                                                                          <div>
49                                                                           <div>
50                                                                            <div>
51                                                                             <div>
52                                                                              <div>
53                                                                               <div>
54                                                                                <div>
55                                                                                 <div>
56                                                                                  <div>
57                                                                                   <div>
58                                                                                    <div>
59                                                                                     <div>
60                                                                                      <div>

```



CANVAS - Animación Base



El método: *requestAnimationFrame()* informa al navegador que quieres realizar una animación y solicita que el navegador programe el repintado de la ventana para el próximo ciclo de animación. El método acepta como argumento una función a la que llamar antes de efectuar el repintado.

Por lo general esta función ejecuta 60 fotogramas por segundo. Si dividimos 1000 milisegundos, es decir un segundo, entre 60, tendremos: 16.666666, es decir, aproximadamente 17, número que se encuentra frecuentemente en las animaciones. La mayoría de los monitores comunes refrescan una página a 60 hertzios, por lo que, si se hace una animación a más de 60 cuadros por segundo, está no se apreciara. En

la línea 37 se observa la ejecución de la función: *setTimeout* en intervalos de 17 milisegundos, es decir, 60 fotogramas por segundo.

En el anterior código definimos una función: *anima()*, que consta de tres partes:

1. el llamado de la función: *requestAnimationFrame()*, que a su vez tiene como parámetro la misma función *anima()*, es decir es una función recursiva.
2. La función de verificación de coordenadas, *verifica()*
3. Y la función *pinta()*, que ejecuta el dibujo de los objetos. Lo primero que hace es borrar todo el lienzo y vuelve lo vuelve a pintar, el resultado es una esfera negra que cruza el canvas.

En la línea 22 se controla la velocidad de la esfera.

## Aproximación 01

Las imágenes que se utilizarán en el desarrollo del videojuego son:



Los sonidos son:

- ☐ boing.mp3
- ☐ boing.ogg
- ☐ boom.mp3
- ☐ boom.ogg
- ☐ clic.mp3
- ☐ clic.ogg
- ☐ disparo.mp3
- ☐ disparo.ogg
- ☐ error.mp3
- ☐ error.ogg
- ☐ gameOver.mp3
- ☐ gameOver.ogg
- ☐ intro.mp3
- ☐ intro.ogg

Lo primero que debemos realizar es la definición de variables, en ellas guardaremos los valores de las teclas que vamos a utilizar en código ASCII.

```

1  <!doctype html>
2  <html>
3  <head>
4    <meta charset="UTF-8">
5    <title>Invasores</title>
6    <style>
7    body{
8      width:600px;
9      margin:0 auto;
10   }
11   h1{
12     text-align:center;
13   }
14   #miCanvas{
15     border:dotted 2px yellow;
16     background:black;
17   }
18   </style>
19   <script>
20   /******
21   VARIABLES
22   *****/
23   var canvas, ctx;
24   var x = 100;
25   var y = 100;
26   var KEY_ENTER = 13;
27   var KEY_LEFT = 37;
28   var KEY_UP = 38;
29   var KEY_RIGHT = 39;
30   var KEY_DOWN = 40;
31   var BARRA = 32;
32   var imagen, imagenEnemigo;
33
34   var teclaPulsada = null;
35   var tecla = [];
36   var colorBala = "red";
37   var balas_array = new Array();
38   var enemigos_array = new Array();

```

Luego vamos a definir los objetos que vamos a utilizar: las naves, el cañon y las balas.

```

39  /*****
40  OBJETOS
41  *****/
42  function Bala(x,y,w){
43      this.x = x;
44      this.y = y;
45      this.w = w;
46      this.dibuja = function(){};
47  }
48  function Jugador(x){
49      this.x = x;
50      this.y = 450;
51      this.dibuja = function(x){
52          this.x = x;
53          ctx.drawImage(imagen, this.x, this.y, 30, 15);
54      };
55  }
56  function Enemigo(x,y){
57      this.x = x;
58      this.y = y;
59      this.w = 35;
60      this.veces = 0;
61      this.dx = 5;
62      this.ciclos = 0;
63      this.num = 14;
64      this.figura = true;
65      this.vive = true;
66      this.dibuja = function(){};
67      //
68  }

```

Luego seguimos con las funciones de animación (vistas en el ejemplo anterior)

```

69  /*****
70  FUNCIONES
71  *****/
72  function anima(){
73      requestAnimationFrame(anima);
74      verifica();
75      pinta();
76  }
77  function verifica(){
78      if(tecla[KEY_RIGHT]) x+=10;
79      if(tecla[KEY_LEFT]) x-=10;
80  }
81  function pinta(){
82      ctx.clearRect(0,0,canvas.width, canvas.height);
83      jugador.dibuja(x);
84  }
85  window.requestAnimationFrame=(function(){
86      return window.requestAnimationFrame ||
87             window.webkitRequestAnimationFrame ||
88             window.mozRequestAnimationFrame ||
89             function(callback){window.setTimeout(callback,17);}
90  })();

```

Ahora seguimos con la función que permite cargar la imagen del cañon:

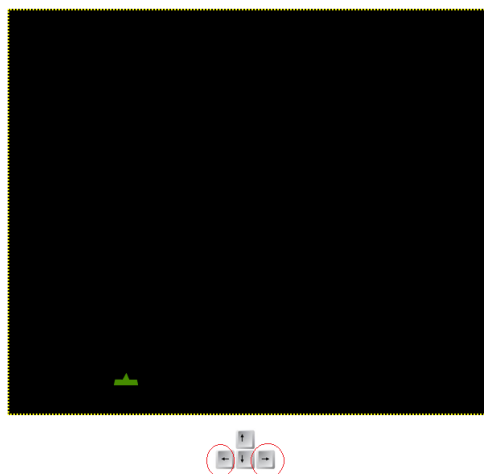
```

91  /*****
92  LISTENER
93  *****/
94  document.addEventListener("keydown",function(e){
95      teclaPulsada=e.keyCode;
96      tecla[e.keyCode]=true;
97  });
98  document.addEventListener("keyup",function(e){
99      tecla[e.keyCode]=false;
100  });
101  window.onload = function(){
102      canvas = document.getElementById("miCanvas");
103      if(canvas && canvas.getContext){
104          ctx = canvas.getContext("2d");
105          if(ctx){
106              x = canvas.width/2;
107              imagen = new Image();
108              imagen.src = "imagenes/torre.fw.png";
109              imagen.onload = function(){
110                  jugador = new Jugador(0);
111                  jugador.dibuja(canvas.width/2);
112                  anima();
113              }
114          } else {
115              alert("Error al crear tu contexto");
116          }
117      }
118  }
119  </script>
120  </head>
121
122  <body>
123      <h1>CANVAS Invaders</h1>
124      <canvas id="miCanvas" width="600px" height="500px">
125          Tu navegador no soporta CANVAS
126      </canvas>
127  </body>
128  </html>

```

De esta forma cuando rendericemos nuestra pagina web, tendremos un cañon que se puede mover en la pantalla al presionar las teclas.

CANVAS Invaders Aproximación 01



## Aproximación 02

En esta aproximación se van a agregar los disparos que se deben realizar desde el cañón, esto se realiza modificando el método verifica() definiendo los elementos que permitan realizar esta acción al presionar la tecla: barra espaciadora, esto se complementara con los métodos dibuja() y pinta() que permitirán dibujar las balas disparadas.

De igual forma se crearan las naves enemigas las cuales se crearán en la función onload() así como su movimiento dentro del canvas.

### 1. Definimos nuestras variables:

```

1  <!doctype html>
2  <html>
3  <head>
4    <meta charset="UTF-8">
5    <title>Invasores</title>
6    <style>
7    <body{
8      width:600px;
9      margin:0 auto;
10   }
11   <h1{
12     text-align:center;
13   }
14   <#miCanvas{
15     border:dotted 2px yellow;
16     background:black;
17   }
18   </style>
19   <script>
20   /*****
21    VARIABLES
22    *****/
23   var canvas, ctx;
24   var x = 100;
25   var y = 100;
26   var KEY_ENTER = 13;
27   var KEY_LEFT = 37;
28   var KEY_UP = 38;
29   var KEY_RIGHT = 39;
30   var KEY_DOWN = 40;
31   var BARRA = 32;
32   var imagen, imagenEnemigo;
33
34   var teclaPulsada = null;
35   var tecla = [];
36   var colorBala = "red";
37   var balas_array = new Array();
38   var enemigos_array = new Array();

```

## 2. Procedemos a definir los objetos:

```

39  /*****
40  OBJETOS
41  *****/
42  function Bala(x,y,w) {
43      this.x = x;
44      this.y = y;
45      this.w = w;
46      this.dibuja = function() {
47          ctx.save();
48          ctx.fillStyle = colorBala;
49          ctx.fillRect(this.x, this.y, this.w, this.w);
50          this.y = this.y - 4;
51          ctx.restore();
52      };
53  }
54  function Jugador(x) {
55      this.x = x;
56      this.y = 450;
57      this.dibuja = function(x) {
58          this.x = x;
59          ctx.drawImage(imagen, this.x, this.y, 30, 15);
60      };
61  }
62  function Enemigo(x,y) {
63      this.x = x;
64      this.y = y;
65      this.w = 35;
66      this.veces = 0;
67      this.dx = 5;
68      this.ciclos = 0;
69      this.num = 14;
70      this.figura = true;
71      this.vive = true;
72      this.dibuja = function() {
73          //Retraso
74          if(this.ciclos > 30) {
75              //saltos
76              if(this.veces > this.num) {
77                  this.dx *= -1;
78                  this.veces = 0;
79                  this.num = 28;
80                  this.y += 20;
81                  this.dx = (this.dx > 0) ? this.dx++ : this.dx--;
82              } else {
83                  this.x += this.dx;
84              }
85              this.veces++;
86              this.ciclos = 0;
87          } else {
88              this.ciclos++;
89          }
90          ctx.drawImage(imagenEnemigo, 0, 0, 40, 30,
91              this.x, this.y, 35, 30);
92      };
93  }

```



## 3. Funciones y Listener:

```

94  /*****
95  FUNCIONES
96  *****/
97  function anima() {
98      requestAnimationFrame(anima);
99      verifica();
100     pinta();
101 }
102 function verifica() {
103     if(tecla[KEY_RIGHT]) x+=10;
104     if(tecla[KEY_LEFT]) x-=10;
105     //Verifica cañon
106     if(x>canvas.width-10) x = canvas.width -10;
107     if(x<0) x = 0;
108     //Disparo
109     if(tecla[BARRA]){
110         balas_array.push(new Bala(jugador.x+12,jugador.y-3,5));
111         tecla[BARRA]=false;
112     }
113 }
114 function pinta() {
115     ctx.clearRect(0,0,canvas.width, canvas.height);
116     jugador.dibuja(x);
117     //Balas
118     for(var i=0; i<balas_array.length; i++){
119         if(balas_array[i]!=null){
120             balas_array[i].dibuja();
121             if(balas_array[i].y<0) balas_array[i] = null;
122         }
123     }
124     //Enemigos
125     for(var i=0; i<enemigos_array.length; i++){
126         enemigos_array[i].dibuja();
127     }
128 }
129 window.requestAnimationFrame=(function() {
130     return window.requestAnimationFrame ||
131         window.webkitRequestAnimationFrame ||
132         window.mozRequestAnimationFrame ||
133         function(callback) {window.setTimeout(callback,17);}
134 })();
135 /*****
136 LISTENER
137 *****/
138 document.addEventListener("keydown",function(e){
139     teclaPulsada=e.keyCode;
140     tecla[e.keyCode]=true;
141 });
142 document.addEventListener("keyup",function(e){
143     tecla[e.keyCode]=false;
144 });

```

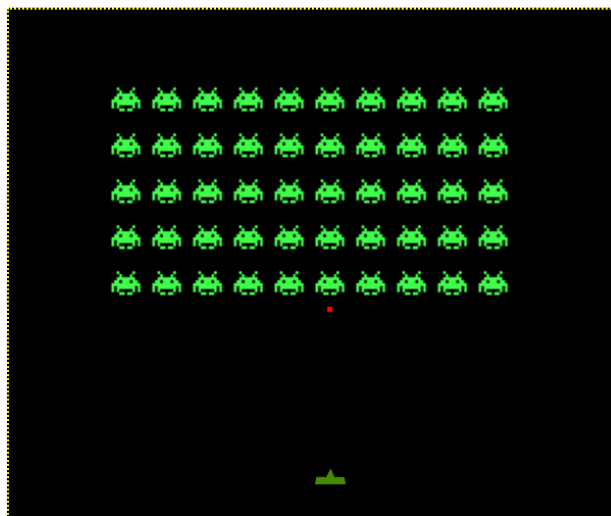
## 4. Y el Inicio:

```

145  /*****
146  INICIO
147  *****/
148  window.onload = function(){
149      canvas = document.getElementById("miCanvas");
150      if(canvas && canvas.getContext){
151          ctx = canvas.getContext("2d");
152          if(ctx){
153              x = canvas.width/2;
154              imagen = new Image();
155              imagenEnemigo = new Image();
156              imagenEnemigo.src = "imagenes/invader.fw.png";
157              imagen.src = "imagenes/torre.fw.png";
158              imagen.onload = function(){
159                  jugador = new Jugador(0);
160                  jugador.dibuja(canvas.width/2);
161                  anima();
162              }
163              imagenEnemigo.onload = function(){
164                  for(var i=0; i<5; i++){
165                      for(var j=0; j<10; j++){
166                          enemigos_array.push(new Enemigo(100+40*j, 30+45*i));
167                      }
168                  }
169              }
170              } else {
171                  alert("Error al crear tu contexto");
172              }
173          }
174      }
175  </script>
176  </head>
177
178  <body>
179  <h1>CANVAS Invaders</h1>
180  <canvas id="miCanvas" width="600px" height="500px">
181  Tu navegador no soporta CANVAS
182  </canvas>
183  </body>
184  </html>

```

## CANVAS Invaders



### Aproximación 03

En esta aproximación se van a agregar los movimientos de las naves de izquierda a derecha y el descenso de estas, si logran tocar el cañón el juego terminara; de igual forma los disparos de las naves que si logran impactar al cañón el juego terminara.

De igual forma se realizará la animación de las naves enemigas usando spritesheet para simular el movimiento. También vamos a realizar la interacción de los elementos a través de las colisiones, agregando una función colisiones().

De igual forma se agregará un puntaje o marcador, el mensaje Game Over al finalizar el juego y algunos sonidos.

#### 1. Definimos nuestras variables:

```

1  <!doctype html>
2  <html>
3  <head>
4    <meta charset="UTF-8">
5    <title>Invasores</title>
6  </head>
7  <body>
8    width:600px;
9    margin:0 auto;
10  </body>
11  <h1>
12    text-align:center;
13  </h1>
14  <div id="miCanvas">
15    border:dotted 2px yellow;
16    background:black;
17  </div>
18  </style>
19  <script>
20  /*****
21  VARIABLES
22  *****/
23  var canvas, ctx;
24  var x = 100;
25  var y = 100;
26  var KEY_ENTER = 13;
27  var KEY_LEFT = 37;
28  var KEY_UP = 38;
29  var KEY_RIGHT = 39;
30  var KEY_DOWN = 40;
31  var BARRA = 32;
32  var imagen, imagenEnemigo;
33
34  var teclaPulsada = null;
35  var tecla = [];
36  var colorBala = "red";
37  var balas_array = new Array();
38  var enemigos_array = new Array();
39  var balasEnemigas_array = new Array();
40  var de;
41  var puntos = 0;
42  var finJuego = false;

```

## 2. Procedemos a definir los objetos:

```

43  /*****
44  OBJETOS
45  *****/
46  function Bala(x,y,w){
47      this.x = x;
48      this.y = y;
49      this.w = w;
50      this.dibuja = function(){
51          ctx.save();
52          ctx.fillStyle = colorBala;
53          ctx.fillRect(this.x, this.y, this.w, this.w);
54          this.y = this.y - 4;
55          ctx.restore();
56      };
57      this.dispara = function(){
58          ctx.save();
59          ctx.fillStyle = colorBala;
60          ctx.fillRect(this.x, this.y, this.w, this.w);
61          this.y = this.y + 6;
62          ctx.restore();
63      };
64  }
65  function Jugador(x){
66      this.x = x;
67      this.y = 450;
68      this.w = 30;
69      this.h = 15;
70      this.dibuja = function(x){
71          this.x = x;
72          ctx.drawImage(imagen, this.x, this.y, this.w, this.h);
73      };
74  }
75  function Enemigo(x,y){
76      this.x = x;
77      this.y = y;
78      this.w = 35;
79      this.veces = 0;
80      this.dx = 5;
81      this.ciclos = 0;
82      this.num = 14;
83      this.figura = true;
84      this.vive = true;
85      this.dibuja = function(){
86          //Retraso
87          if(this.ciclos > 30){
88              //saltitos
89              if(this.veces > this.num){
90                  this.dx *= -1;
91                  this.veces = 0;
92                  this.num = 28;
93                  this.y += 20;
94                  this.dx = (this.dx > 0) ? this.dx++ : this.dx--;
95              } else {
96                  this.x += this.dx;
97              }
98              this.veces++;
99              this.ciclos = 0;
100              this.figura = !this.figura;
101          } else {
102              this.ciclos++;
103          }
104          if(this.vive){
105              if(this.figura){
106                  ctx.drawImage(imagenEnemigo, 0, 0, 40, 30, this.x, this.y, 35, 30);
107              } else {
108                  ctx.drawImage(imagenEnemigo, 50, 0, 35, 30, this.x, this.y, 35, 30);
109              }
110          } else {
111              ctx.fillStyle = "black";
112              ctx.fillRect(this.x, this.y, 35, 30);
113          }
114      };
115  }
116  }

```

## 3. Funciones y Listener:

```

117  /*****
118  FUNCIONES
119  *****/
120  function anima(){
121      if(finJuego==false){
122          requestAnimationFrame(anima);
123          verifica();
124          pinta();
125          colisiones();
126      }
127  }
128  function score(){
129      ctx.save();
130      ctx.fillStyle = "white";
131      ctx.clearRect(0,0,canvas.width,40);
132      ctx.font = "bold 20px Courier";
133      ctx.fillText("SCORE: "+puntos,10,20);
134      ctx.restore();
135  }
136  function mensaje(cadena){
137      var lon = (canvas.width-(53*cadena.length))/2;
138      ctx.fillStyle = "green";
139      ctx.clearRect(0,0,canvas.width, canvas.height);
140      ctx.font = "bold 50px Rosewood Std";
141      ctx.fillText(cadena,lon,220);
142  }
143  function colisiones(){
144      for(var i=0; i<enemigos_array.length; i++){
145          for(var j=0; j<balas_array.length; j++){
146              enemigo = enemigos_array[i];
147              bala = balas_array[j];
148              if(enemigo != null && bala != null){
149                  if((bala.x > enemigo.x)&&
150                     (bala.x < enemigo.x+enemigo.w)&&
151                     (bala.y > enemigo.y)&&
152                     (bala.y < enemigo.y+enemigo.h)){
153                      enemigo.vive = false;
154                      enemigos_array[i] = null;
155                      balas_array[j] = null;
156                      puntos += 10;
157                      boing.play();
158                  }
159              }
160          }
161      }
162      for(var j=0; j<balasEnemigas_array.length; j++){
163          bala = balasEnemigas_array[j];
164          if(bala != null){
165              if((bala.x > jugador.x)&&
166                 (bala.x < jugador.x+jugador.w)&&
167                 (bala.y > jugador.y)&&
168                 (bala.y < jugador.y+jugador.h)){
169                  gameOver();
170              }
171          }
172      }
173  }
174  function gameOver(){
175      ctx.clearRect(0,0,canvas.width, canvas.height);
176      balas_array = [];
177      enemigos_array = [];
178      balasEnemigas_array = [];
179      clearTimeout(de);
180      finJuego = true;
181      mensaje("GAME OVER");
182      fin.play();
183  }

```

```

184 function verifica(){
185     if(tecla[KEY_RIGHT]) x+=10;
186     if(tecla[KEY_LEFT]) x-=10;
187     //Verifica cañon
188     if(x>canvas.width-10) x = canvas.width -10;
189     if(x<0) x = 0;
190     //Disparo
191     if(tecla[BARRA]){
192         balas_array.push(
193             new Bala(jugador.x+12,jugador.y-3,5));
194         tecla[BARRA]=false;
195         disparaEnemigo();
196         disparo.play();
197     }
198 }
199 function pinta(){
200     ctx.clearRect(0,0,canvas.width, canvas.height);
201     score();
202     jugador.dibuja(x);
203     //Balas
204     for(var i=0; i<balas_array.length; i++){
205         if(balas_array[i]!=null){
206             balas_array[i].dibuja();
207             if(balas_array[i].y<0) balas_array[i] = null;
208         }
209     }
210     //Balas Enemigas
211     for(var i=0; i<balasEnemigas_array.length; i++){
212         if(balasEnemigas_array[i]!=null){
213             balasEnemigas_array[i].dispara();
214             if(balasEnemigas_array[i].y>canvas.height) balasEnemigas_array[i] = null;
215         }
216     }
217     //Enemigos
218     numEnemigos = 0;
219     for(var i=0; i<enemigos_array.length; i++){
220         if(enemigos_array[i] != null){
221             enemigos_array[i].dibuja();
222             numEnemigos++;
223             if(enemigos_array[i].y==jugador.y) gameOver();
224         }
225     }
226     if(numEnemigos==0) gameOver();
227 }
228 function disparaEnemigo(){
229     var ultimos = new Array();
230     for(var i=enemigos_array.length-1; i>0; i--){
231         if(enemigos_array[i]!=null){
232             ultimos.push(i);
233         }
234         if(ultimos.length==10) break;
235     }
236     d = ultimos[Math.floor(Math.random()*10)];
237     balasEnemigas_array.push(new Bala(enemigos_array[d].x+enemigos_array[d].w/2,
238     enemigos_array[d].y,5));
239 }
240 window.requestAnimationFrame=(function(){
241     return window.requestAnimationFrame ||
242         window.webkitRequestAnimationFrame ||
243         window.mozRequestAnimationFrame ||
244         function(callback){window.setTimeout(callback,17);};
245 })();
246 /*****
247 LISTENER
248 *****/
249 document.addEventListener("keydown",function(e){
250     teclaPulsada=e.keyCode;
251     tecla[e.keyCode]=true;
252 });
253 document.addEventListener("keyup",function(e){
254     tecla[e.keyCode]=false;
255 });

```

## 4. Y el Inicio:

```

256 /*****
257 INICIO
258 *****/
259 window.onload = function(){
260     canvas = document.getElementById("miCanvas");
261     if(canvas && canvas.getContext){
262         ctx = canvas.getContext("2d");
263         if(ctx){
264             var boing = document.getElementById("boing");
265             var disparo = document.getElementById("disparo");
266             var intro = document.getElementById("intro");
267             var fin = document.getElementById("fin");
268             x = canvas.width/2;
269             imagen = new Image();
270             imagenEnemigo = new Image();
271             imagenEnemigo.src = "imagenes/invader.fw.png";
272             imagen.src = "imagenes/torre.fw.png";
273             mensaje("INVASORES");
274             intro.play();
275             imagen.onload = function(){
276                 jugador = new Jugador(0);
277                 setTimeout(anima,3500);
278             }
279             imagenEnemigo.onload = function(){
280                 for(var i=0; i<5; i++){
281                     for(var j=0; j<10; j++){
282                         enemigos_array.push(new Enemigo(100+40*j, 30+45*i));
283                     }
284                 }
285                 de = setTimeout(disparaEnemigo,3500);
286             }
287             } else {
288                 alert("Error al crear tu contexto");
289             }
290         }
291     }
292 }
293 </script>
294 </head>
295 <body>
296 <h1>CANVAS Invaders</h1>
297 <canvas id="miCanvas" width="600px" height="500px">
298     Tu navegador no soporta CANVAS
299 </canvas>
300 <audio id="boing">
301     <source src="sonidos/boing.mp3">
302     <source src="sonidos/boing.ogg">
303 </audio>
304 <audio id="disparo">
305     <source src="sonidos/disparo.mp3">
306     <source src="sonidos/disparo.ogg">
307 </audio>
308 <audio id="intro">
309     <source src="sonidos/intro.mp3">
310     <source src="sonidos/intro.ogg">
311 </audio>
312 <audio id="fin">
313     <source src="sonidos/gameOver.mp3">
314     <source src="sonidos/gameOver.ogg">
315 </audio>
316 </body>
</html>

```

