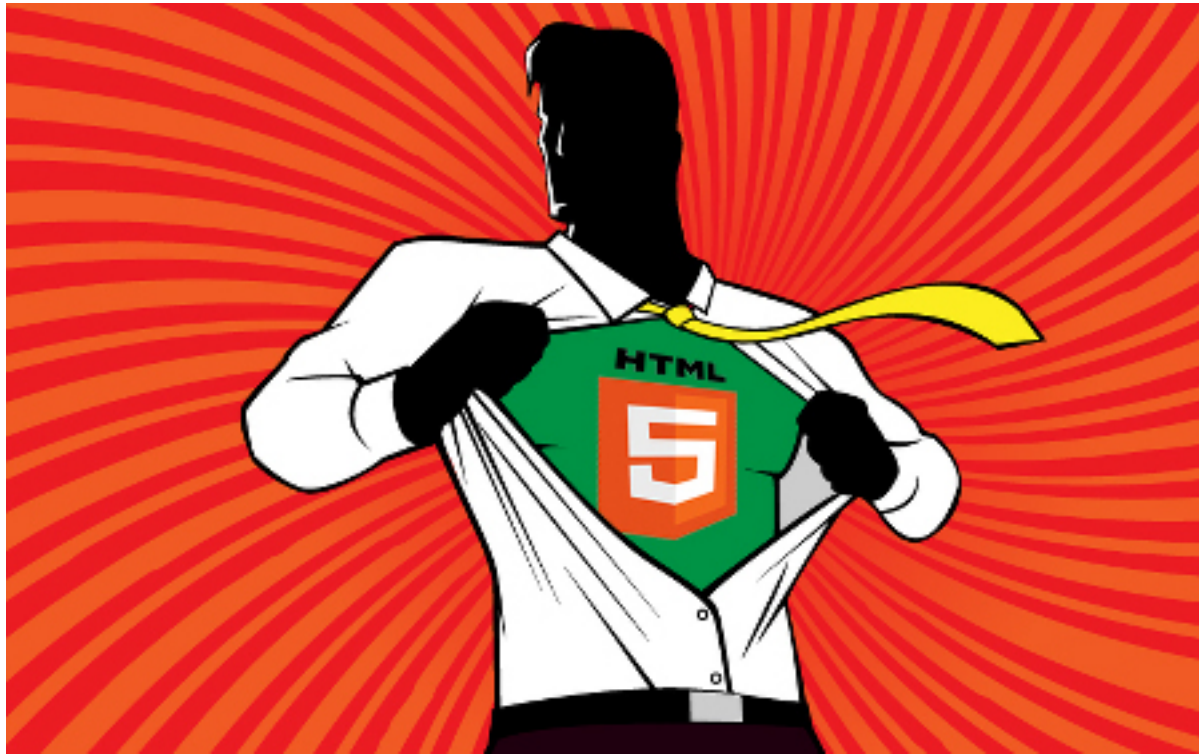


“Taller de Videojuegos con HTML 5”



Creación del Videojuego “Space Invaders”

Rogelio Ferreira Escutia



Estructura Básica



Archivos Básicos

- **marcianos.htm**
 - **Estructura de la página Web.**
- **marcianos.css**
 - **Diseño de la página.**
- **marcianos.js**
 - **Funciones JavaScript para manejo del Canvas.**



marcianos.htm

```
<!DOCTYPE html>
```

```
<html>
```

```
  <head>
```

```
    <title>Marcianos</title>
```

```
    <meta charset="utf-8" />
```

```
    <link rel="stylesheet" href="marcianos.css" />
```

```
    <script language="javascript" src="marcianos.js"></script>
```

```
  </head>
```

```
  <body>
```

```
    <h1>CANVAS Invaders</h1>
```

```
    <canvas id="miCanvas" width="600px" height="500px">
```

```
      Tu navegador no soporta CANVAS
```

```
    </canvas>
```

```
  </body>
```

```
</html>
```



marcianos.css

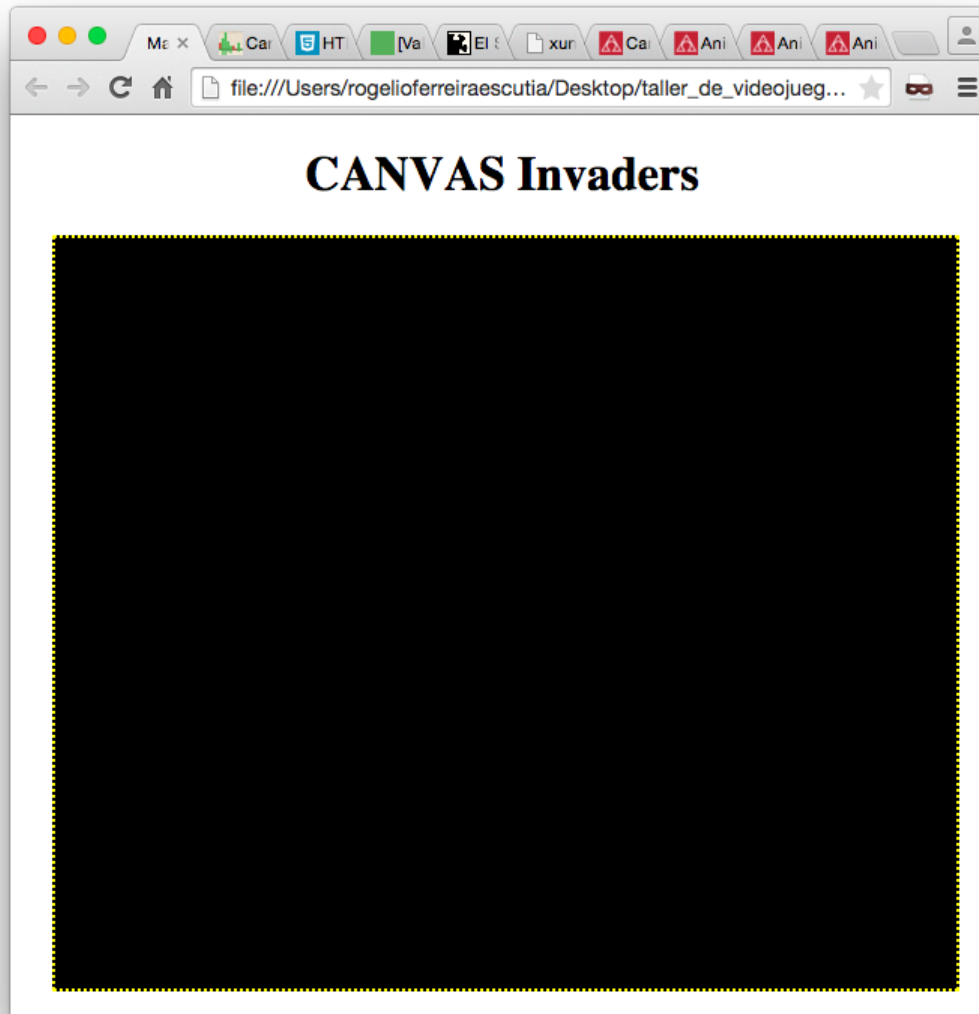
```
/*  marcianos.css  */  
  
body{  
    width:600px;  
    margin:0 auto;  
}  
h1{  
    text-align:center;  
}  
#miCanvas{  
    border:dotted 2px yellow;  
    background:black;  
}
```



marcianos.js

```
//  
// marcianos.js  
//  
  
window.onload = arrancarCanvas;  
  
function arrancarCanvas(){  
    canvas=document.getElementById('miCanvas');  
    if(canvas&&canvas.getContext){  
        ctx=canvas.getContext("2d");  
        if (ctx) {  
              
                //Aqui va a ir el código  
              
        } else { alert("Error al crear el contexto"); }  
    }  
}
```





Variables Globales



Variables Globales

- **Agregamos las variables globales que utilizaremos a nuestro archivo “marcianos.js”**



marcianos.js

```
var canvas, ctx;  
var x = 100;  
var y = 100;  
var KEY_ENTER = 13;  
var KEY_LEFT = 37;  
var KEY_UP = 38;  
var KEY_RIGHT = 39;  
var KEY_DOWN = 40;  
var BARRA = 32;  
var imagen, imagenEnemigo;  
  
var teclaPulsada = null;  
var tecla = [];  
var colorBala = "red";  
var balas_array = new Array();  
var enemigos_array = new Array();
```



Declarar Objetos



Declarar Objetos

- **Declaramos en nuestro archivo “marcianos.js”, los objetos que utilizaremos en nuestro juego.**



Declarar Objetos

```
function Bala(x,y,w){
    this.x = x;
    this.y = y;
    this.w = w;
    this.dibuja = function({});
}
function Jugador(x){
    this.x = x;
    this.y = 450;
    this.dibuja = function(x){
        this.x = x;
        ctx.drawImage(imagen, this.x, this.y, 30, 15);
    };
}
function Enemigo(x,y){
    this.x = x;
    this.y = y;
    this.w = 35;
    this.veces = 0;
    this.dx = 5;
    this.ciclos = 0;
    this.num = 14;
    this.figura = true;
    this.vive = true;
    this.dibuja = function({});
    //
}
```

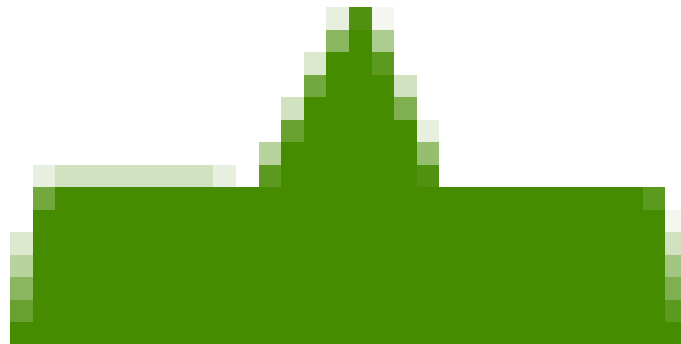


Gráfico del Cañón



Cañón

- **Nuestro cañón será el gráfico “torre.fw.png” el cual se coloca en el mismo directorio que nuestro código JavaScript:**



Dibujar Cañón



Dibujar Cañón

- En nuestra función “arrancarCanvas” cargamos nuestro “cañón” y lo colocamos en el centro en la parte inferior:

```
if (ctx) {  
  
    //Aqui va a ir el código  
    x = canvas.width/2;  
    imagen = new Image();  
    imagen.src = "torre.fw.png";  
    imagen.onload = function(){  
        jugador = new Jugador(0);  
        jugador.dibuja(canvas.width/2);  
        anima(); }  
  
} else { alert("Error al crear el contexto"); }
```



Función anima()

- En el código anterior se manda llamar a “anima” que es una función que invoca al

“requestAnimationFrame(anima);”
- Que se encargan de ejecutar las funciones de verifica y pinta:

```
function anima(){  
    requestAnimationFrame(anima);  
    verifica();  
    pinta();  
}
```



Función verifica()

- En el código anterior se manda llamar a “verifica” que es una función que detecta si se presionó la flecha izquierda o derecha, e incrementa el contador respectivo:

```
function verifica(){  
    if(tecla[KEY_RIGHT]) x+=10;  
    if(tecla[KEY_LEFT]) x-=10;  
}
```



Función verifica()

- En el código anterior se manda llamar a “verifica” que es una función que detecta si se presionó la flecha izquierda o derecha, e incrementa el contador respectivo:

```
function pinta(){  
    ctx.clearRect(0,0,canvas.width, canvas.height);  
    jugador.dibuja(x);  
}
```



Detectar tipo de objeto

- **Se crea el código para detectar el tipo de objeto “requestAnimationFrame” que utiliza el navegador que estamos utilizando:**

```
window.requestAnimationFrame=(function(){  
    return window.requestAnimationFrame ||  
        window.webkitRequestAnimationFrame ||  
        window.mozRequestAnimationFrame ||  
        function(callback){window.setTimeout(callback,17);}  
})();
```

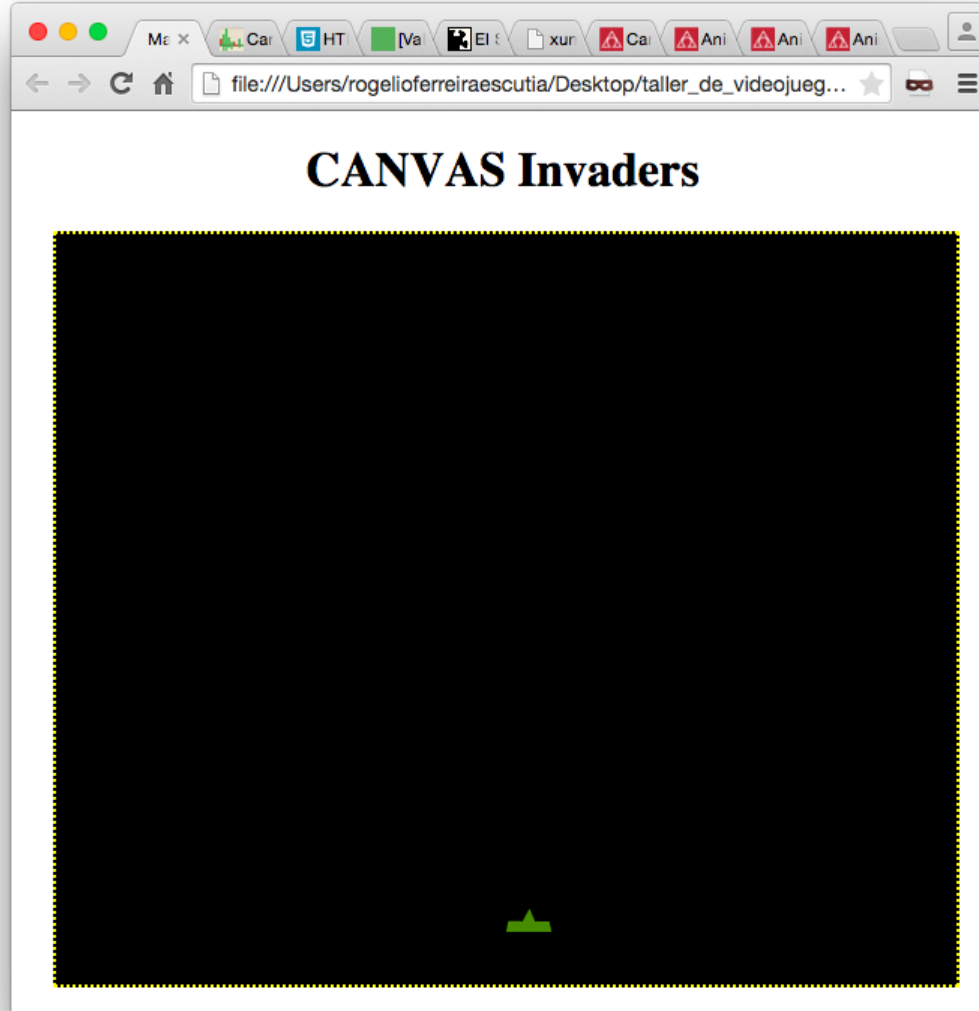


Detectar teclado

- Creamos una nueva sección (abajo de las funciones) para manejo de eventos (LISTENER).
- Este primer código sirve para detectar los eventos del teclado.

```
/******  
LISTENER  
*****/  
document.addEventListener("keydown",function(e){  
    teclaPulsada=e.keyCode;  
    tecla[e.keyCode]=true;  
});  
  
document.addEventListener("keyup",function(e){  
    tecla[e.keyCode]=false;  
});
```





Cañón y Balas



Verificar cañón (1)

- **Hasta aquí, si seguimos oprimiendo las teclas izquierda derecha, el cañón sale de la pantalla, por lo cual modificamos la función “verifica” para si llega al borde de la ventana, no se incremente o decremente mas su coordenada x.**
- **También se verificar si se oprime la barra espaciadora. Si es así se crea una nueva bala y se pone en falso la bandera de la barra espaciadora, para que no se generen demasiadas balas.**

Verificar cañón (2)

```
function verifica(){
    if(tecla[KEY_RIGHT]) x+=10;
    if(tecla[KEY_LEFT]) x-=10;
    //Verifica cañon
    if(x>canvas.width-10) x = canvas.width -10;
    if(x<0) x = 0;
    //Disparo
    if(tecla[BARRA]){
        balas_array.push(new Bala(jugador.x+12,jugador.y-3,5));
        tecla[BARRA]=false;
    }
}
```



Pintar balas

- **Se modifica la función “pinta”. Se le agrega un ciclo para pintar las balas que se agregaron al arreglo de balas.**

```
function pinta(){
    ctx.clearRect(0,0,canvas.width, canvas.height);
    jugador.dibuja(x);
    //
    for(var i=0; i<balas_array.length; i++){
        if(balas_array[i]!=null){
            balas_array[i].dibuja();
            if(balas_array[i].y<0) balas_array[i] = null;
        }
    }
}
```

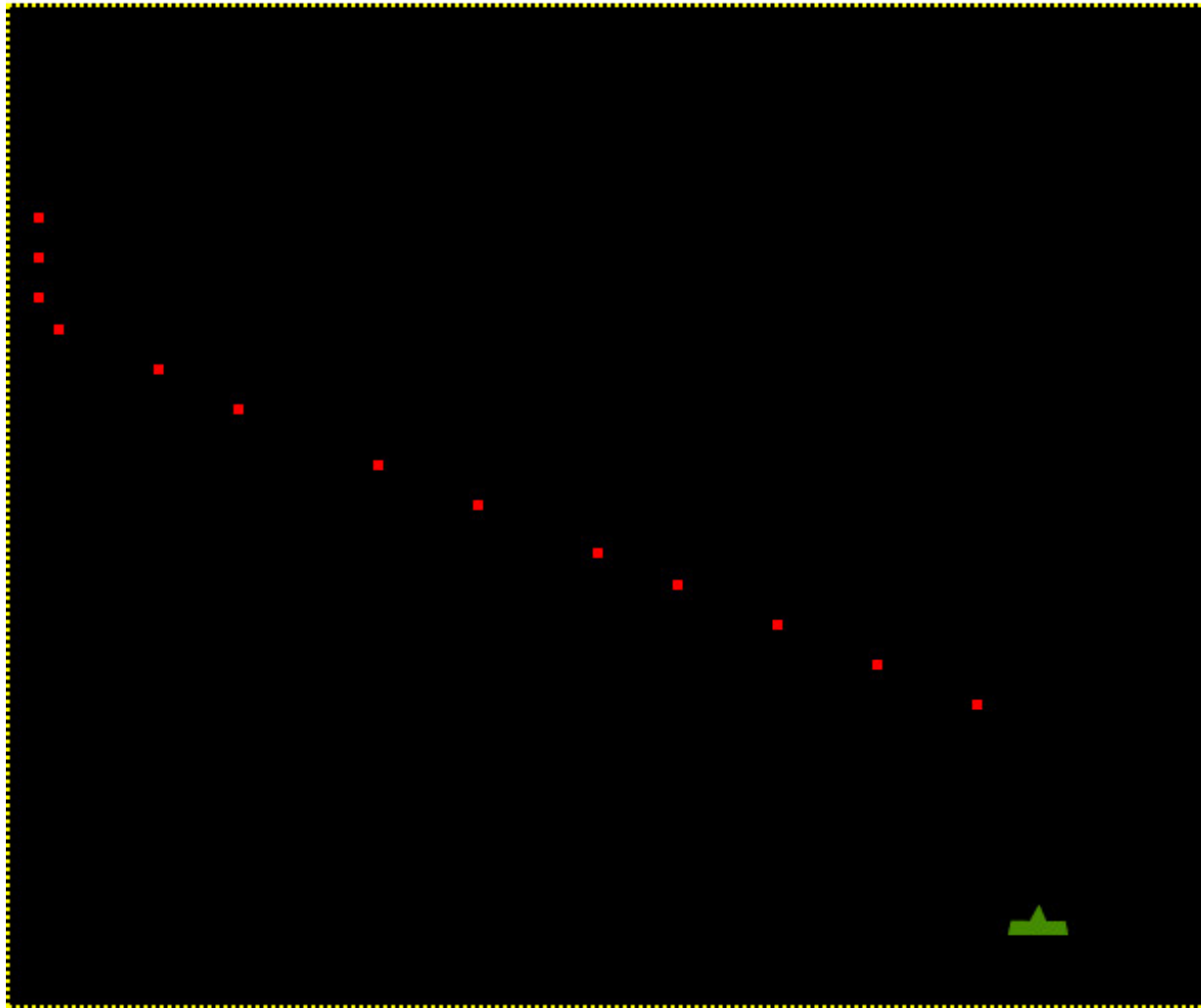


Pintar balas

- **Se modifica la función “bala” para que también pinte las balas cuando se invoque:**

```
function Bala(x,y,w){
    this.x = x;
    this.y = y;
    this.w = w;
    this.dibuja = function(){
        ctx.save();
        ctx.fillStyle = colorBala;
        ctx.fillRect(this.x, this.y, this.w, this.w);
        this.y = this.y - 4;
        ctx.restore();
    };
}
```

CANVAS Invaders

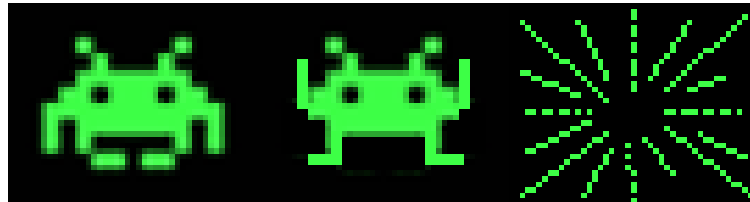


Naves Enemigas



Marcianos

- **Nuestro marciano (invasor) será el gráfico “invader.fw.png” el cual se coloca en el mismo directorio que nuestro código JavaScript:**



Pintar Marcianos

- **Se modifica la función “arrancarCanvas” para que al momento de arranque, dibuje también a los marcianos.**
- **El código de esta función queda de la siguiente manera:**




```

canvas = document.getElementById("miCanvas");
if(canvas && canvas.getContext){
    ctx = canvas.getContext("2d");
    if(ctx){
        x = canvas.width/2;
        imagen = new Image();
        imagen.src = "torre.fw.png";
        imagen.onload = function(){
            jugador = new Jugador(0);
            jugador.dibuja(canvas.width/2);
            anima();
        }
        imagenEnemigo = new Image();
        imagenEnemigo.src = "invader.fw.png";
        imagenEnemigo.onload = function(){
            for(var i=0; i<5; i++){
                for(var j=0; j<10; j++){
                    enemigos_array.push(
                        new Enemigo(100+40*j, 30+45*i));
                }
            }
        }
    } else {
        alert("Error al crear tu contexto");
    }
}

```



Marcianos

- Se modifica la función Enemigo, para que se incorpore el dibujo del marciano:

```
function Enemigo(x,y){
    this.x = x;
    this.y = y;
    this.w = 35;
    this.vecas = 0;
    this.dx = 5;
    this.ciclos = 0;
    this.num = 14;
    this.figura = true;
    this.vive = true;
    this.dibuja = function(){
        ctx.drawImage(imagenEnemigo,0,0,40,30,
            this.x, this.y, 35,30);
    };
}
```



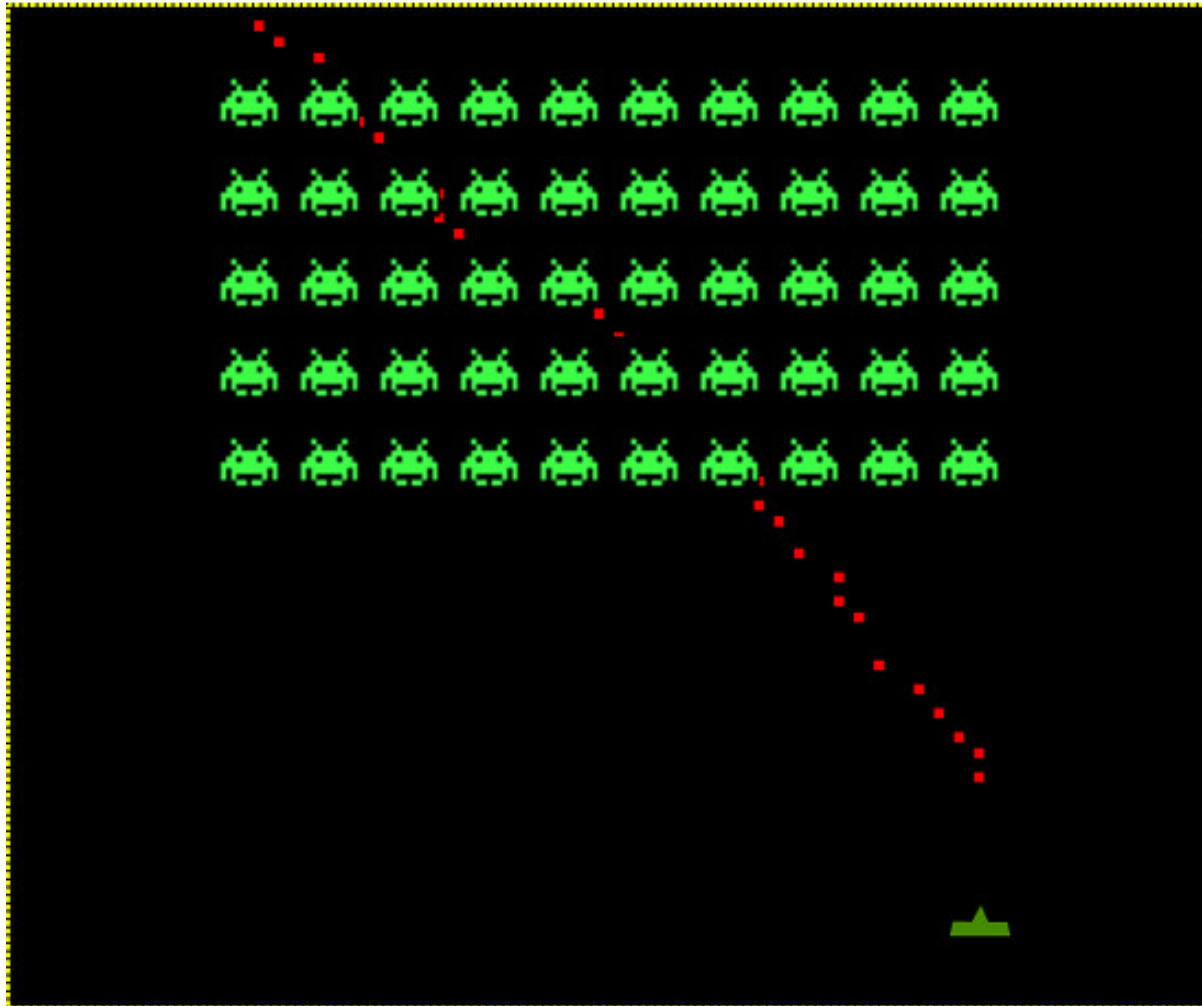
Marcianos

- Se modifica la función “pinta” para que pinte además del cañón y las balas, también los enemigos:

```
function pinta(){
    ctx.clearRect(0,0,canvas.width, canvas.height);
    jugador.dibuja(x);
    //Balas
    for(var i=0; i<balas_array.length; i++){
        if(balas_array[i]!=null){
            balas_array[i].dibuja();
            if(balas_array[i].y<0) balas_array[i] = null;
        }
    }
    //Enemigos
    for(var i=0; i<enemigos_array.length; i++){
        enemigos_array[i].dibuja();
    }
}
```



CANVAS Invaders



Moviendo las Naves Enemigas



Objeto Enemigo

- **Modificamos el objeto “Enemigo”, para que dibuje los marcianos cada medio segundo, y los mueve haciendo saltos en pixeles.**
- **Movemos los marcianos de izquierda a derecha hasta que llegue al borde derecho, luego baja y ahora avanzan hacia la izquierda.**



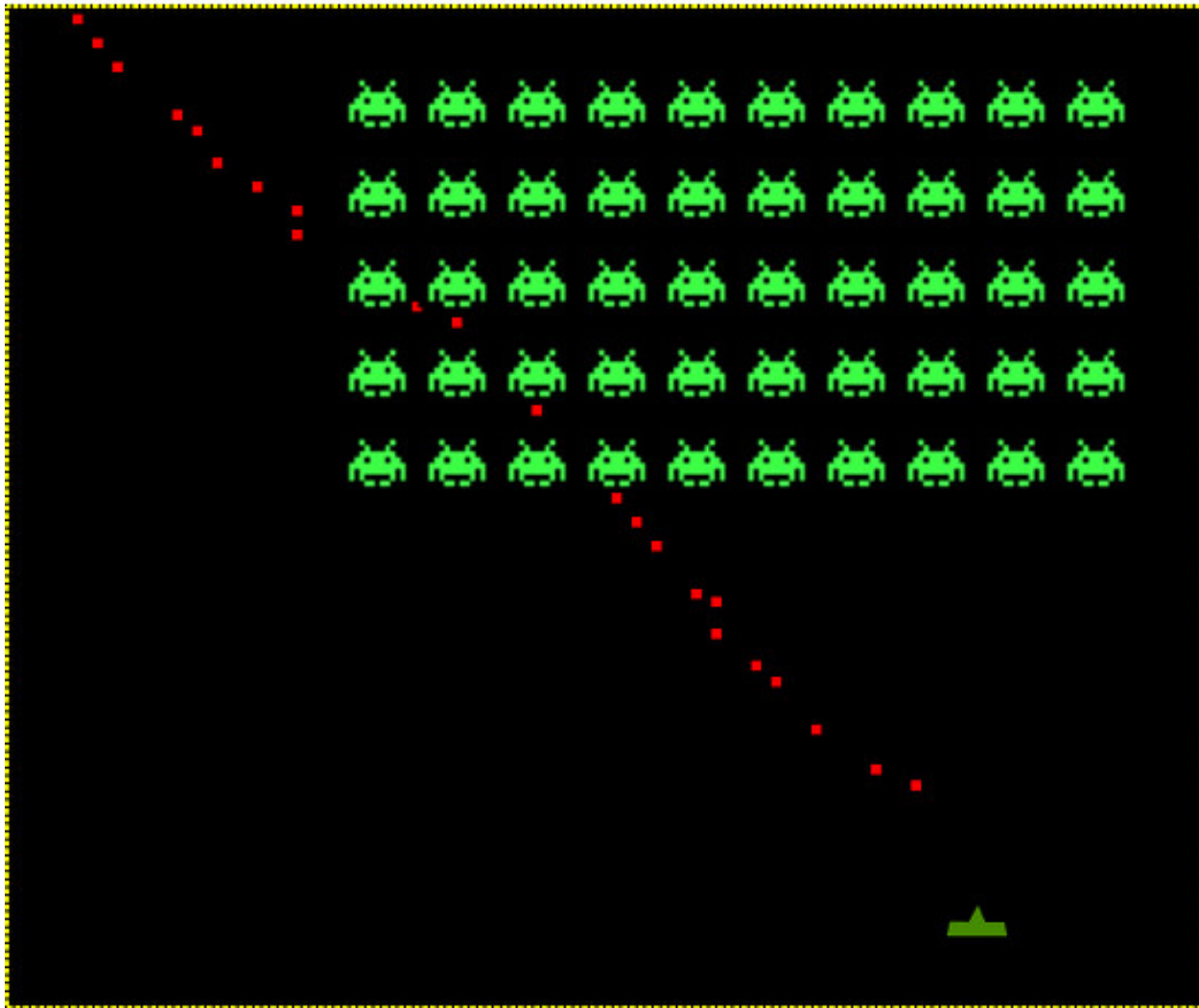
```

function Enemigo(x,y){
  this.x = x;
  this.y = y;
  this.w = 35;
  this.veces = 0;
  this.dx = 5;
  this.ciclos = 0;
  this.num = 14;
  this.figura = true;
  this.vive = true;
  this.dibuja = function(){
    //Retraso
    if(this.ciclos > 30){
      //saltitos
      if(this.veces>this.num){
        this.dx *= -1;
        this.veces = 0;
        this.num = 28;
        this.y += 20;
        this.dx = (this.dx>0)? this.dx++:this.dx--;
      } else {
        this.x += this.dx;
      }
      this.veces++;
      this.ciclos = 0;
    } else {
      this.ciclos++;
    }
    ctx.drawImage(imagenEnemigo,0,0,40,30,
      this.x, this.y, 35,30);
  };
}

```



CANVAS Invaders



Animando las Naves Enemigas



Objeto Enemigo

- **Se modifica el objeto enemigo para que pinte de manera alternada los objetos del Spritesheet, y queda el código de la siguiente manera:**



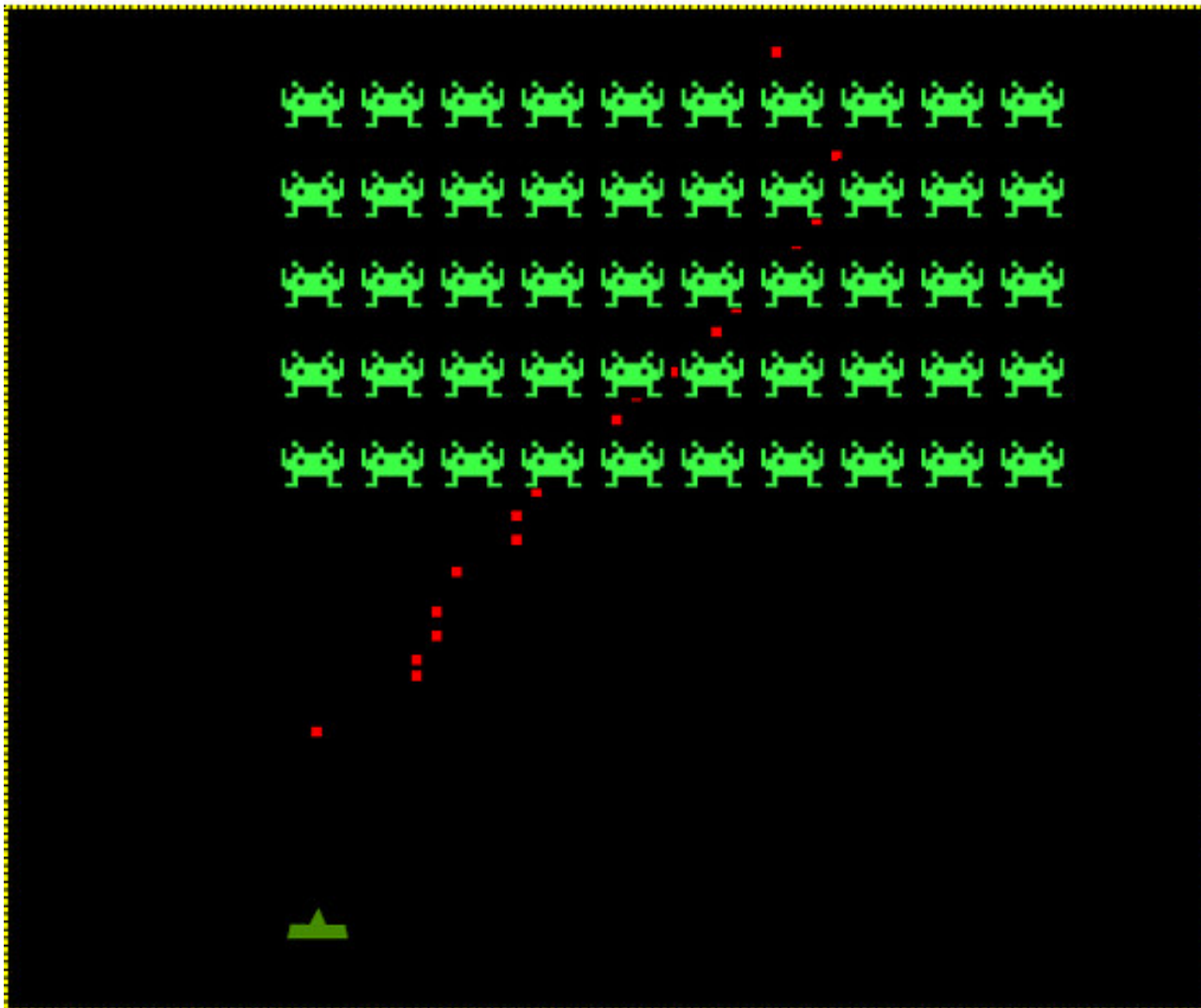
```

function Enemigo(x,y){
    this.x = x;
    this.y = y;
    this.w = 35;
    this.veces = 0;
    this.dx = 5;
    this.ciclos = 0;
    this.num = 14;
    this.figura = true;
    this.vive = true;
    this.dibuja = function(){
        //Retraso
        if(this.ciclos > 30){
            //saltitos
            if(this.veces>this.num){
                this.dx *= -1;
                this.veces = 0;
                this.num = 28;
                this.y += 20;
                this.dx = (this.dx>0)? this.dx++:this.dx--;
            } else {
                this.x += this.dx;
            }
            this.veces++;
            this.ciclos = 0;
            this.figura = !this.figura;
        } else {
            this.ciclos++;
        }
        if(this.figura){
            ctx.drawImage(imagenEnemigo,0,0,40,30,
                this.x, this.y, 35,30);
        } else {
            ctx.drawImage(imagenEnemigo,50,0,35,30,
                this.x, this.y, 35, 30);
        }
    };
}

```



CANVAS Invaders



Colisiones



Colisiones

- Ahora detectaremos cuando haya una colisión, es decir, que choque una bala con un marciano.
- Para lo anterior, primero modificamos la función “anima”, agregando la llamada a una nueva función llamada “colisiones”.

```
function anima(){  
    requestAnimationFrame(anima);  
    verifica();  
    pinta();  
    colisiones();  
}
```



Colisiones

- Creamos el código para la función “colisiones”

```
function colisiones(){
    for(var i=0; i<enemigos_array.length; i++){
        for(var j=0; j<balas_array.length; j++){
            enemigo = enemigos_array[i];
            bala = balas_array[j];
            if(enemigo != null && bala != null){
                if((bala.x > enemigo.x)&&
                    (bala.x < enemigo.x+enemigo.w)&&
                    (bala.y > enemigo.y)&&
                    (bala.y < enemigo.y+enemigo.w)){
                    enemigo.vive = false;
                    enemigos_array[i] = null;
                    balas_array[j] = null;
                }
            }
        }
    }
}
```



Colisiones

- **Modificamos el objeto “enemigo”**




```

function Enemigo(x,y){
  this.x = x;
  this.y = y;
  this.w = 35;
  this.veces = 0;
  this.dx = 5;
  this.ciclos = 0;
  this.num = 14;
  this.figura = true;
  this.vive = true;
  this.dibuja = function(){
    //Retraso
    if(this.ciclos > 30){
      //saltitos
      if(this.veces>this.num){
        this.dx *= -1;
        this.veces = 0;
        this.num = 28;
        this.y += 20;
        this.dx = (this.dx>0)? this.dx++:this.dx--;
      } else {
        this.x += this.dx;
      }
      this.veces++;
      this.ciclos = 0;
      this.figura = !this.figura;
    } else {
      this.ciclos++;
    }
    if(this.vive){
      if(this.figura){
        ctx.drawImage(imagenEnemigo,0,0,40,30, this.x, this.y, 35,30);
      } else {
        ctx.drawImage(imagenEnemigo,50,0,35,30, this.x, this.y, 35, 30);
      }
    } else {
      ctx.fillStyle = "black";
      ctx.fillRect(this.x, this.y, 35, 30);
    }
  }
};
}

```



Colisiones

- **Modificamos el nuestra función de “arrancarCanvas”:**



```

function arrancarCanvas(){
    canvas = document.getElementById("miCanvas");
    if(canvas && canvas.getContext){
        ctx = canvas.getContext("2d");
        if(ctx){
            x = canvas.width/2;
            imagen = new Image();
            imagenEnemigo = new Image();
            imagenEnemigo.src = "invader.fw.png";
            imagen.src = "torre.fw.png";
            imagen.onload = function(){
                jugador = new Jugador(0);
                jugador.dibuja(canvas.width/2);
                anima();
            }
            imagenEnemigo.onload = function(){
                for(var i=0; i<5; i++){
                    for(var j=0; j<10; j++){
                        enemigos_array.push(new Enemigo(100+40*j, 30+45*i));
                    }
                }
            }
        } else {
            alert("Error al crear tu contexto");
        }
    }
}

```



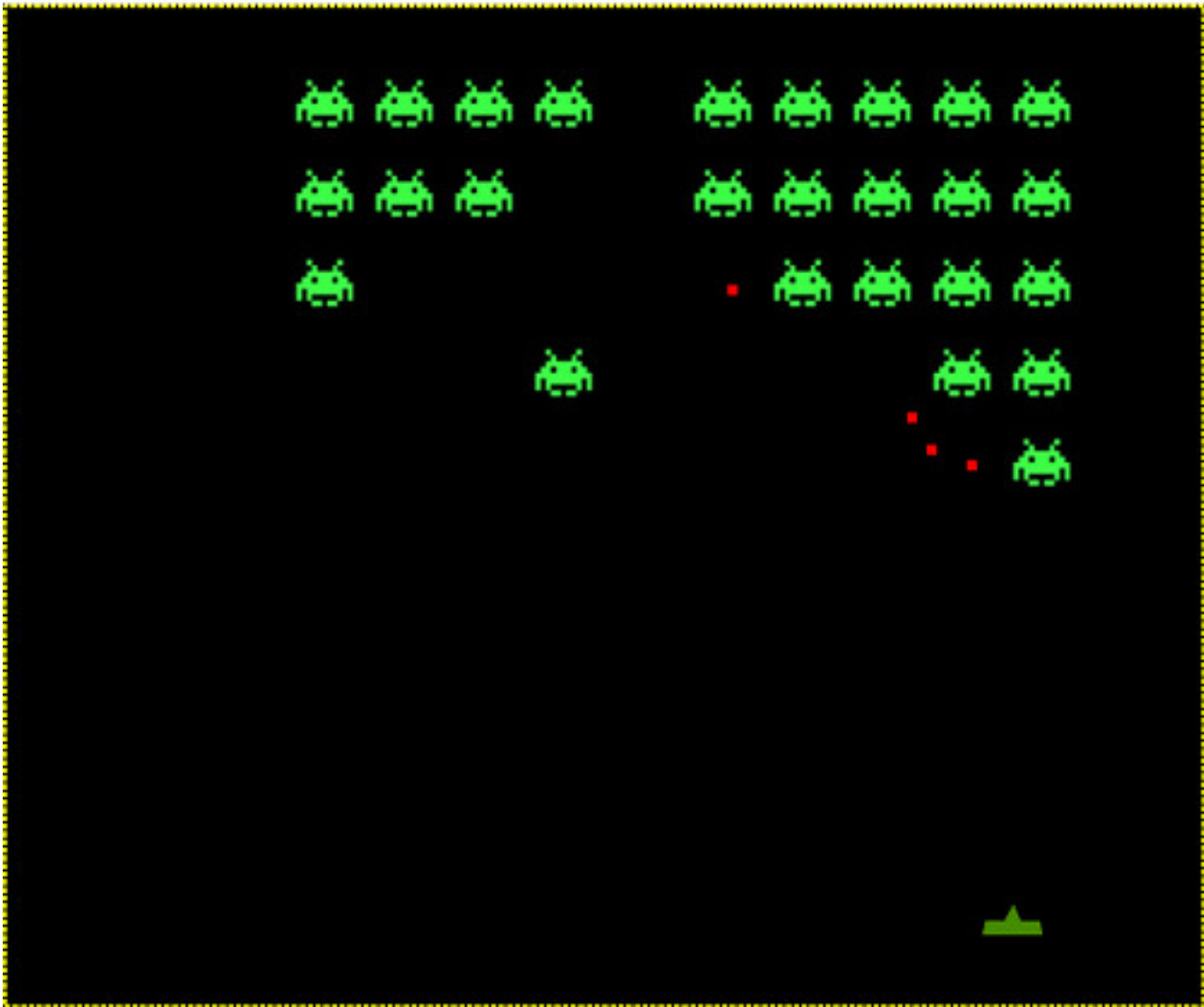
Colisiones

- **Modificamos el nuestra función de “pinta”:**

```
function pinta(){
    ctx.clearRect(0,0,canvas.width, canvas.height);
    jugador.dibuja(x);
    //Balas
    for(var i=0; i<balas_array.length; i++){
        if(balas_array[i]!=null){
            balas_array[i].dibuja();
            if(balas_array[i].y<0) balas_array[i] = null;
        }
    }
    //Enemigos
    for(var i=0; i<enemigos_array.length; i++){
        if(enemigos_array[i] != null){
            enemigos_array[i].dibuja();
        }
    }
}
```



CANVAS Invaders



Balas Enemigas



Balas Enemigas

- **Agregamos 2 nuevas variables globales a nuestro código para controlar las balas enemigas:**

```
var canvas, ctx;  
var x = 100;  
var y = 100;  
var KEY_ENTER = 13;  
var KEY_LEFT = 37;  
var KEY_UP = 38;  
var KEY_RIGHT = 39;  
var KEY_DOWN = 40;  
var BARRA = 32;  
var imagen, imagenEnemigo;  
  
var teclaPulsada = null;  
var tecla = [];  
var colorBala = "red";  
var balas_array = new Array();  
var enemigos_array = new Array();  
var balasEnemigas_array = new Array();  
var de;
```



Balas Enemigas

- **Agregamos a nuestra función principal “arrancarCanvas”, la siguiente línea para que dibuje balas enemigas cada segundo y medio:**

```
de = setTimeout(disparaEnemigo,1500);
```



Balas Enemigas

- Nuestra función principal “arrancarCanvas” queda así:

```
function arrancarCanvas(){
    canvas = document.getElementById("miCanvas");
    if(canvas && canvas.getContext){
        ctx = canvas.getContext("2d");
        if(ctx){
            x = canvas.width/2;
            imagen = new Image();
            imagenEnemigo = new Image();
            imagenEnemigo.src = "invader.fw.png";
            imagen.src = "torre.fw.png";
            imagen.onload = function(){
                jugador = new Jugador(0);
                jugador.dibuja(canvas.width/2);
                anima();
            }
            imagenEnemigo.onload = function(){
                for(var i=0; i<5; i++){
                    for(var j=0; j<10; j++){
                        enemigos_array.push(new Enemigo(100+40*j, 30+45*i));
                    }
                }
                de = setTimeout(disparaEnemigo,1500);
            }
        } else {
            alert("Error al crear tu contexto");
        }
    }
}
```



Balas Enemigas

- Creamos la nueva función “disparaEnemigo”, haciendo que disparen las 10 naves mas cercanas a nuestro cañón:

```
function disparaEnemigo(){
    var ultimos = new Array();
    for(var i=enemigos_array.length-1; i>0; i--){
        if(enemigos_array[i]!=null){
            ultimos.push(i);
        }
        if(ultimos.length==10) break;
    }
    d = ultimos[Math.floor(Math.random()*10)];
    balasEnemigas_array.push(
    new Bala(enemigos_array[d].x+enemigos_array[d].w/2,
    enemigos_array[d].y,5));
}
```



Balas Enemigas

- **Modificamos nuestra función “pinta”, para que también pinte a las balas enemigas:**



Balas Enemigas

```
function pinta(){
    ctx.clearRect(0,0,canvas.width, canvas.height);
    jugador.dibuja(x);
    //Balas
    for(var i=0; i<balas_array.length; i++){
        if(balas_array[i]!=null){
            balas_array[i].dibuja();
            if(balas_array[i].y<0) balas_array[i] = null;
        }
    }
    //Balas Enemigas
    for(var i=0; i<balasEnemigas_array.length; i++){
        if(balasEnemigas_array[i]!=null){
            balasEnemigas_array[i].dispara();
            if(balasEnemigas_array[i].y>canvas.height)
                balasEnemigas_array[i] = null;
        }
    }
    //Enemigos
    for(var i=0; i<enemigos_array.length; i++){
        if(enemigos_array[i] != null){
            enemigos_array[i].dibuja();
        }
    }
}
```



Balas Enemigas

- **Modificamos nuestro objeto “Bala”:**

```
function Bala(x,y,w){
    this.x = x;
    this.y = y;
    this.w = w;
    this.dibuja = function(){
        ctx.save();
        ctx.fillStyle = colorBala;
        ctx.fillRect(this.x, this.y, this.w, this.w);
        this.y = this.y - 4;
        ctx.restore();
    };
    this.dispara = function(){
        ctx.save();
        ctx.fillStyle = colorBala;
        ctx.fillRect(this.x, this.y, this.w, this.w);
        this.y = this.y + 6;
        ctx.restore();
    };
}
```



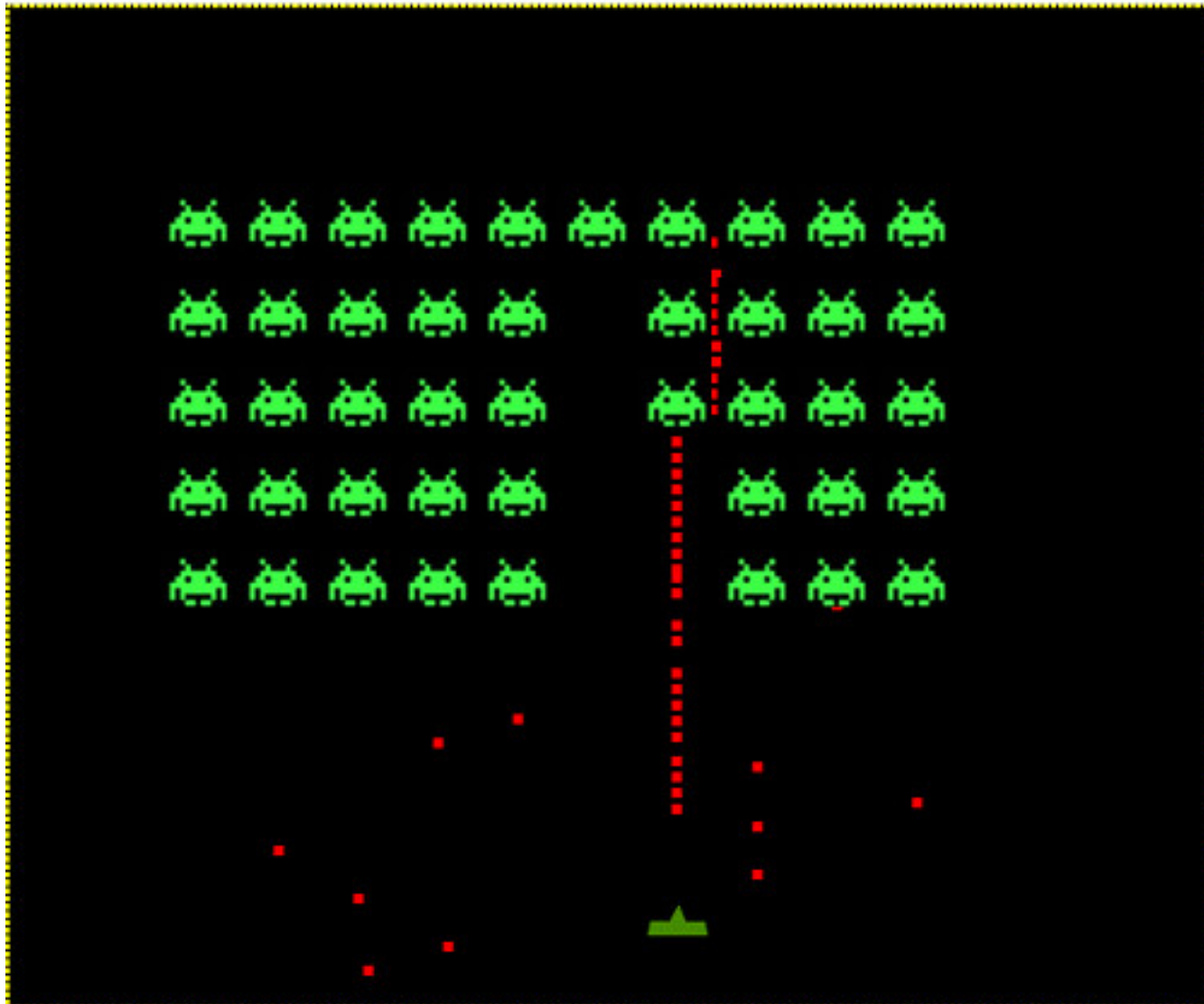
Balas Enemigas

- **Modificamos nuestra función “verifica”:**

```
function verifica(){
    if(tecla[KEY_RIGHT]) x+=10;
    if(tecla[KEY_LEFT]) x-=10;
    //Verifica cañon
    if(x>canvas.width-10) x = canvas.width -10;
    if(x<0) x = 0;
    //Disparo
    if(tecla[BARRA]){
        balas_array.push(new Bala(jugador.x+12,jugador.y-3,5));
        tecla[BARRA]=false;
        disparaEnemigo()
    }
}
```



CANVAS Invaders



***Detección de colisiones entre
balas enemigas y cañón
(fin del juego)***



Fin de Juego

- **Modificamos nuestro función “colisiones” para revisar si alguna de las balas chocó con nuestro cañón:**



```

function colisiones(){
    for(var i=0; i<enemigos_array.length; i++){
        for(var j=0; j<balas_array.length; j++){
            enemigo = enemigos_array[i];
            bala = balas_array[j];
            if(enemigo != null && bala != null){
                if((bala.x > enemigo.x)&&
                    (bala.x < enemigo.x+enemigo.w)&&
                    (bala.y > enemigo.y)&&
                    (bala.y < enemigo.y+enemigo.w)){
                    enemigo.vive = false;
                    enemigos_array[i] = null;
                    balas_array[j] = null;
                }
            }
        }
    }
    for(var j=0; j<balasEnemigas_array.length; j++){
        bala = balasEnemigas_array[j];
        if(bala != null){
            if((bala.x > jugador.x)&&
                (bala.x < jugador.x+jugador.w)&&
                (bala.y > jugador.y)&&
                (bala.y < jugador.y+jugador.h)){
                gameOver();
            }
        }
    }
}

```



Fin de Juego

- **SE observa que la función anterior manda llamar a la función "gameOver" cuando hay una colisión entre una bala enemiga y mi cañón, y el código es el siguiente:**

```
function gameOver(){  
    alert("Kapuuuum");  
}
```



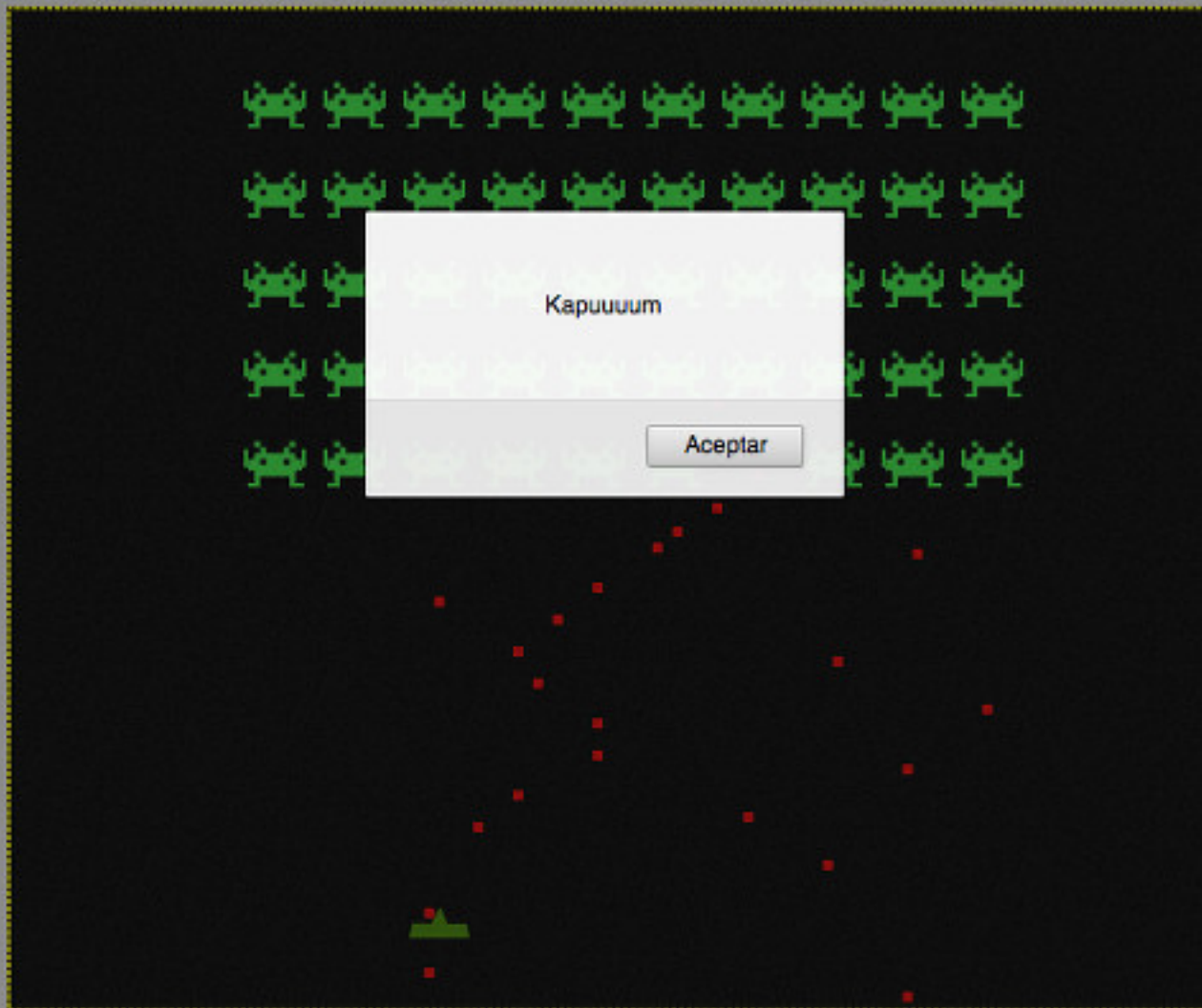
Fin de Juego

- **Agregamos a nuestro objeto “Jugador” su tamaño para que sea detectado cuando choque con una bala enemiga, y el código queda así:**

```
function Jugador(x){
    this.x = x;
    this.y = 450;
    this.w = 30;
    this.h = 15;
    this.dibuja = function(x){
        this.x = x;
        ctx.drawImage(imagen, this.x, this.y, this.w, this.h);
    };
}
```



CANVAS Invaders





Rogelio Ferreira Escutia

Instituto Tecnológico de Morelia
Departamento de Sistemas y Computación

Correo: *rogelio@itmorelia.edu.mx*
 rogeplus@gmail.com

Página Web: *<http://antares.itmorelia.edu.mx/~kaos/>*
 <http://www.xumarhu.net/>

Twitter: *<http://twitter.com/rogeplus>*
Facebook: *<http://www.facebook.com/groups/xumarhu.net/>*

