



Animaciones en Canvas

- C.F.G.S. DAW
- 6 horas semanales
- Mes aprox. de impartición: Ene - Feb
- iPasen - cjaedia071@g.educaand.es

Carmelo José Jaén Díaz

Índice



Objetivo

Glosario

Interacción persona - ordenador

Objetivos

Características. Usable.

Características. Visual.

Características. Educativo y actualizado.

OBJETIVO



- Identificar los formatos de audio y vídeo que se van a utilizar.
- Realizar animaciones a partir de imágenes fijas.
- Aplicar la guía de estilo sobre los elementos multimedia de audio, vídeo y animaciones.
- Reconocer y analizar las tecnologías relacionadas con la inclusión de contenido multimedia e interactivo.
- Agregar elementos multimedia a los documentos web.
- Añadir interactividad a los elementos de un documento web.
- Verificar el funcionamiento de los elementos multimedia en diferentes navegadores.

GLOSARIO



Formato. De audio o de vídeo, tipo de fichero o archivo contenedor de alguno de dichos elementos. En función de este, el audio o vídeo será indicado para un determinado fin; incluso determina en qué navegador podrá ser reproducido.

Fotograma. Cada una de las imágenes que componen un vídeo, en concreto, el vídeo final se produce por la secuencia de fotogramas visualizada de acuerdo con una determinada frecuencia de imágenes por segundo.

Frecuencia. Número de vibraciones por segundo que dan origen al sonido analógico. El espectro de un sonido se caracteriza por su rango de frecuencias, medido en hercios (Hz).

Relación de aspecto. Relación que existe entre el ancho y el alto de un fotograma, y, por tanto, del vídeo, puesto que normalmente se mantiene el mismo tipo de formato a lo largo de toda la grabación. Las más conocidas son 4:3 y 16:9.

Sonido. Sensación producida en el órgano del oído por el movimiento vibratorio de los cuerpos, transmitido por un medio elástico como el aire.

GLOSARIO



Velocidad de transmisión (*bit rate*). Cantidad de espacio físico (en bits) que ocupa un segundo de duración de un audio. El audio tendrá más calidad cuanto mayor sea su *bit rate*, y el archivo que lo contiene tendrá mayor peso.

INTRODUCCIÓN



En la lección anterior hemos estudiado cómo pintar gráficos en un lienzo de `<canvas>`. Veamos ahora cómo realizar una pequeña animación y cómo interactuar con los elementos del lienzo mediante el teclado.



ANIMACIONES EN CANVAS



Para crear animaciones o videojuegos en Canvas seguiremos los siguientes pasos:

1. Pintamos un gráfico en Canvas.
2. Borramos el Canvas pasados unos milisegundos.
3. Volvemos a pintar un gráfico con un ligero cambio.
4. Borramos el Canvas pasados unos milisegundos.
5. Seguimos el mismo proceso de los pasos anteriores.

Esta técnica es como crear dibujos animados utilizando una hoja que hay que borrar en cada fotograma. Este borrado se realiza tan rápido que el ojo humano no lo detecta y lo que se aprecia es una animación. Este es el principio que se suele aplicar en los videojuegos en dos dimensiones.

ANIMACIONES EN CANVAS

Ejemplo



Tal y como estudiamos en la UT3.9 Instrucciones de tiempo, en JavaScript se utiliza el método `setInterval()` para ejecutar un código concreto cada cierto tiempo.

Sin embargo, para modificar el Canvas en intervalos de tiempo se utiliza el método `requestAnimationFrame(función)`.

Se propone el siguiente ejemplo para facilitar el estudio de las animaciones en Canvas:

1. Se implementa un lienzo a modo de escenario de juego.
2. Se implementa un cuadrado a modo de personaje.
3. Se implementan eventos de teclado para controlar al personaje
4. Mediante `requestAnimationFrame()`, se actualiza el lienzo dando la sensación de animación.

ANIMACIONES EN CANVAS

Ejemplo



Lo primero que tenemos que hacer para trabajar con Canvas es crear un lienzo y, a partir de ese lienzo, incluir elementos y modificarlos.

CÓDIGO HTML

1) Crearemos un lienzo (espacio rectangular vacío) en el que serán mostrados todos los elementos:

```
<canvas id="lienzo" width="600" height="400">
```

El navegador no soporta canvas

```
</canvas>
```

ANIMACIONES EN CANVAS

Ejemplo



A continuación, mediante JS, modificaremos el contenido de ese lienzo.

CÓDIGO JS

1) Definimos e inicializamos las variables que vamos a emplear en nuestro videojuego.

```
//Variables globales para almacenar el elemento canvas y contexto  
let canvas=null, lienzo=null;
```

```
//Posición inicial del personaje (un cuadrado)  
let x=50,y=50;
```

ANIMACIONES EN CANVAS

Ejemplo



//Variable para guardar la tecla presionada

```
let lastPress=null;
```

//En nuestro juego, usaremos las teclas izquierda, arriba, derecha y abajo, cuyos codigos de teclado para la propiedad KeyboardEvent: code son

```
const KEY_LEFT = "ArrowLeft";
```

```
const KEY_UP = "ArrowUp";
```

```
const KEY_RIGHT = "ArrowRight";
```

```
const KEY_DOWN = "ArrowDown";
```

ANIMACIONES EN CANVAS

Ejemplo



2) Implementamos la función `iniciar()` que se ejecutará una vez se haya cargado la página web gracias al evento `onload`.

```
function iniciar(){  
    canvas=document.getElementById('lienzo');  
    lienzo=canvas.getContext('2d'); //obtenemos el contexto de dibujo  
    run();  
}  
  
window.addEventListener("load", iniciar, false);
```

ANIMACIONES EN CANVAS

Ejemplo



3) La función `iniciar()` obtiene el elemento `<canvas>` y su contexto, tras esto, invocamos la función `run()`.

```
function iniciar(){
    canvas=document.getElementById( 'lienzo' );
    lienzo=canvas.getContext( '2d' ); //obtenemos el contexto de dibujo
    run();
}

window.addEventListener( "load", iniciar, false );
```

ANIMACIONES EN CANVAS

Ejemplo



4) La función “recursiva” `run()` repinta el lienzo una y otra vez. Adicionalmente implementa dos funciones para actualizar el contenido del mismo.

```
function run(){  
    //requestAnimationFrame(): informa al navegador de que quieres  
    realizar una animación y solicita que el navegador programe el  
    repintado de la ventana para el próximo ciclo de animación.  
    requestAnimationFrame(run); //animación optimizada  
    accionesJuego();  
    pintarLienzo(lienzo);  
}
```

ANIMACIONES EN CANVAS

Ejemplo



5) La función `accionesJuego()` modifica la posición del personaje en el lienzo. Esta posición viene definida por las variables `x` e `y` definidas anteriormente.

```
function accionesJuego(){  
    //Modificamos la dirección que tendrá nuestro player en función de  
    la tecla presionada  
    if (lastPress === KEY_RIGHT) {  
        x += 5; // Mover a la derecha  
    }
```

ANIMACIONES EN CANVAS

Ejemplo



```
else if (lastPress === KEY_LEFT){
    x -= 5; // Mover a la izquierda
}
else if (lastPress === KEY_UP) {
    y -= 5; // Mover hacia arriba
}
else if (lastPress === KEY_DOWN) {
    y += 5; // Mover hacia abajo
}

//verificamos si el personaje ha
salido del canvas, en cuyo caso,
haremos que aparezca por el otro
lado:
// Aparece por la izquierda
    if (x >= canvas.width) x = 0;
// Aparece por la derecha
    if (x < 0) x = canvas.width;
// Aparece por la parte superior
    if (y >= canvas.height) y = 0;
// Aparece por la parte inferior
    if (y < 0) y = canvas.height;
}
```


ANIMACIONES EN CANVAS

Ejemplo



6) La función `pintarLienzo()` actualiza el lienzo dibujando los componentes.

```
function pintarLienzo(lienzo){  
    lienzo.fillStyle="#F7F9FA"; //le ponemos un color al lienzo  
    lienzo.fillRect(0,0,canvas.width,canvas.height); //Dibujamos el  
lienzo  
    lienzo.fillStyle='#0f0';  
    lienzo.fillRect(x,y,10,10); //Dibujamos el jugador en x,y. Su  
tamaño es de 10x10  
}
```

ANIMACIONES EN CANVAS

Ejemplo



7) Por último mediante un evento de teclado **keydown**, almacenamos el código de la tecla pulsada. Se emplea el modelo de eventos del W3C para funciones anónimas estudiado en la UT5.3 Modelo de eventos del W3C.

```
document.addEventListener('keydown', function(event) {  
    //Creamos un manejador de evento para el teclado que se encargue de  
    almacenar la tecla presionada. El evento que nos interesa en este caso  
    es keydown  
    lastPress=event.code;  
}, false);
```

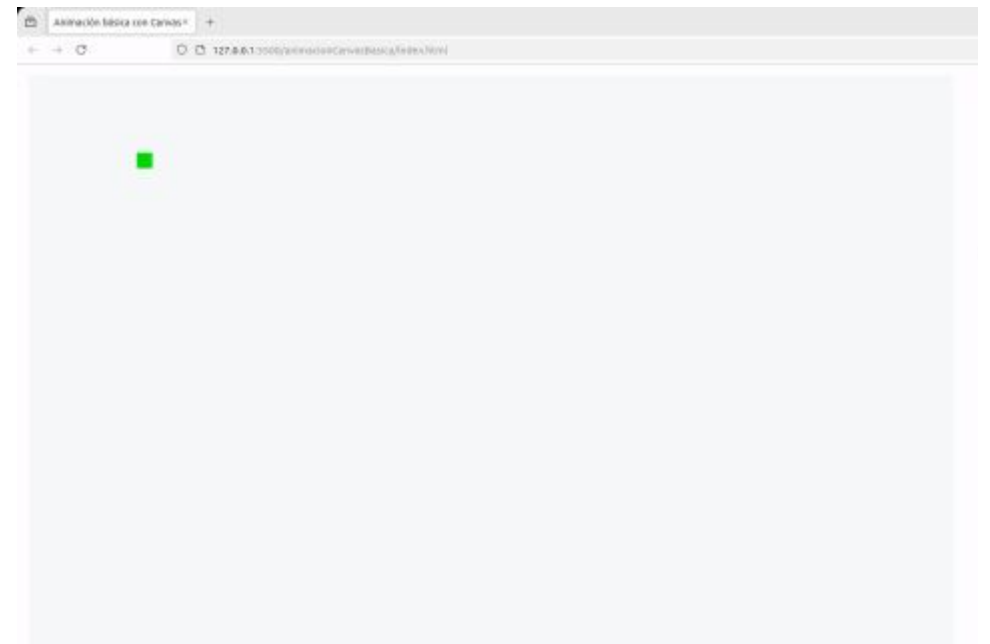
ANIMACIONES EN CANVAS

Ejemplo



La salida del código anterior puede verse en la imagen adjunta o en el siguiente link:

<https://drive.google.com/drive/folders/1qL2Y12TtlvAV42cdD9SsLzbeGPyX9ruS?usp=sharing>



ANIMACIONES EN CANVAS

Recursos



A continuación, se listan una serie de recursos que pueden inspirar tus proyectos utilizando Canvas:

- <https://tpec05.blogspot.com/2018/02/como-crear-una-cuadricula-de-dibujo-en.html>
- <https://www.benjoffe.com/code/games/torus/>
- <https://www.benjoffe.com/code/demos/canvascape/textures>
- <https://29a.ch/jswars/>
- <https://www.kevs3d.co.uk/dev/asteroids/>
- <https://www.paulbrunt.co.uk/bert/>
- <https://www.yvoschaap.com/chainrxn/>