

Unidad 5. Contenidos multimedia en la web: animaciones. Canvas

Canvas y JavaScript.

El API canvas de HTML5 está muy relacionada con el mundo de los juegos para la web. Cuenta con una amplia aceptación, estando disponible en la mayoría de navegadores con una implementación rápida y eficiente. El aprendizaje de esta API también es una gran introducción a la programación gráfica de bajo nivel lo que te permitirá aplicar estos conocimientos a una gran variedad de tecnologías relacionadas con el diseño de juegos.

Actividad 1. Dibujando en el lienzo.

En las siguientes actividades veremos como dibujar líneas, formas, imágenes y texto en un lienzo Canvas.

Apartado A. Lienzo.

Para dibujar es necesario disponer de un espacio en el que hacerlo, un lienzo.

Crea una aplicación web con un lienzo Canvas de 256 x 256 píxeles con un borde de 1 píxel, discontinuo y de color negro.

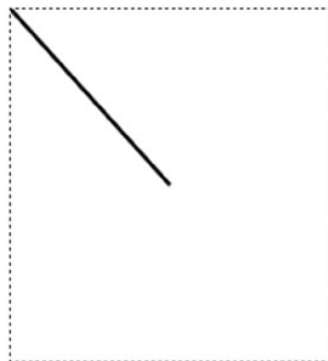
Apartado B. Contexto.

A continuación hay que especificar el "contexto" del lienzo. El contexto puede entenderse como una superficie que se coloca justo encima del lienzo y que renderiza los gráficos que contiene. Se puede especificar un contexto de 2 o 3 dimensiones.

Añade un contexto 2D al lienzo Canvas.

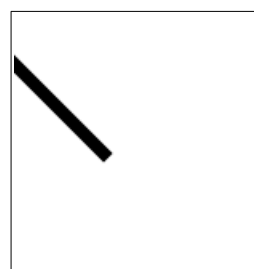
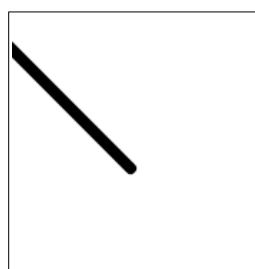
Apartado C. Líneas (`strokeStyle`, `lineWidth`, `beginPath`, `moveTo`, `lineTo`, `stroke`)

Dibuja una línea negra de 3 píxeles de grosor que vaya desde la esquina superior izquierda (0, 0) hasta el centro del lienzo (128, 128).



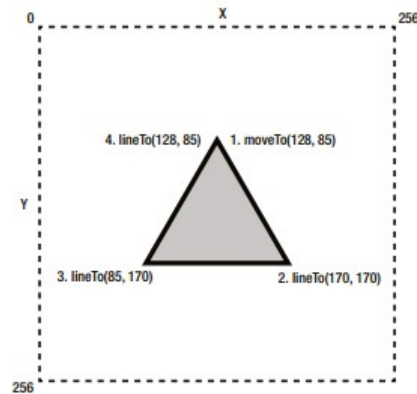
Apartado C.1. Terminaciones de línea (`lineCap`)

Hay tres posibles terminaciones de línea: `square`, `butt` y `round`. Comprueba su funcionamiento. Aumenta el ancho de línea para observar claramente sus diferencias.



Apartado C.2. Unión de líneas (fillStyle, fill)

Crea un triángulo como el de la imagen incluyendo el relleno.



Apartado C.3. Creación de formas complejas.

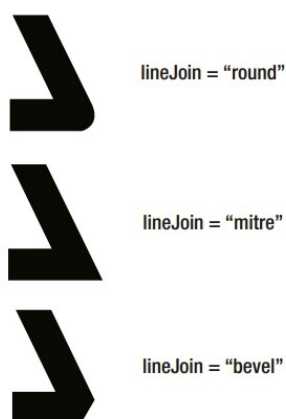
Cuando una forma está constituida por un gran número de puntos, una práctica común es crear un array con sus coordenadas y utilizar una función que los vaya conectando.

Crea una función `dibujarRuta` que dado un array de coordenadas, conecte los puntos para dibujar una forma. Por ejemplo, con el siguiente array la función debe dibujar el triángulo del apartado anterior:

```
let triangulo1 = [
  [128, 85],
  [170, 170],
  [85, 170]
];
```

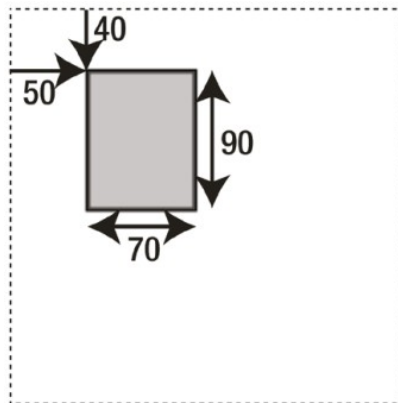
Apartado C.4. Formas de unión.

Hay tres posibles formas de unión de líneas: `round`, `mitre` y `bevel`. Comprueba su funcionamiento.



Apartado D. Cuadrados y rectángulos (rect, strokeRect, fillRect)

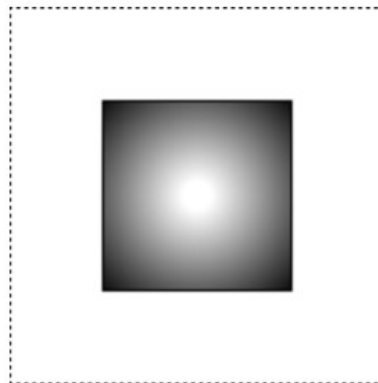
Crea un cuadrado como el de la imagen.



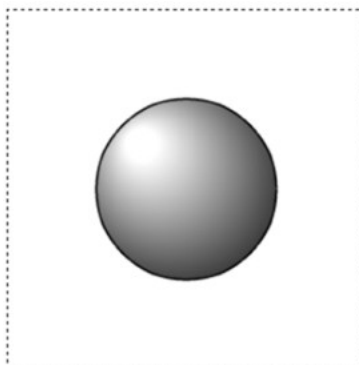
A continuación crea un rectángulo de 100 x 50 píxeles partiendo de la coordenada 110, 170.

Apartado E. Gradiente (createLinearGradient)

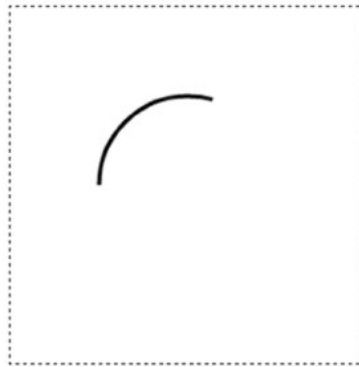
Es posible crear dos tipos de gradientes: lineal o radial. Crea un cuadrado como el de la imagen con un gradiente radial.

**Apartado F. Círculos y arcos (arc)**

Crea un círculo como el de la imagen.



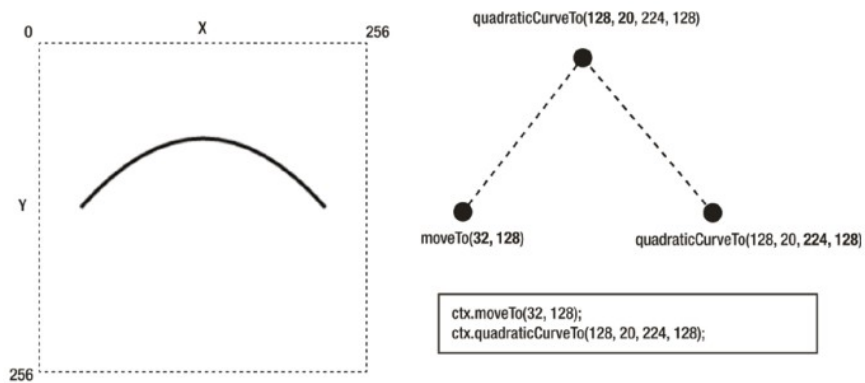
A continuación crea un arco como el siguiente:



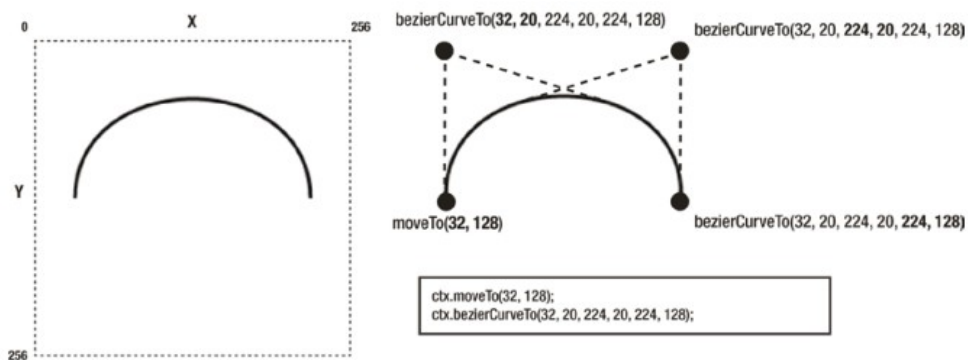
Apartado G. Líneas curvas (quadraticCurveTo, bezierCurveTo)

Se pueden dibujar dos tipos de líneas curvas: cuadráticas o bezier.

Crea una curva cuadrática como la siguiente:

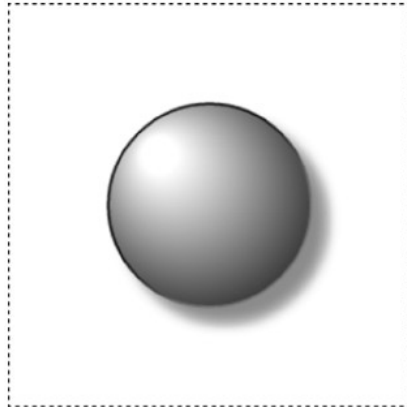


Crea una curva bezier como la siguiente:



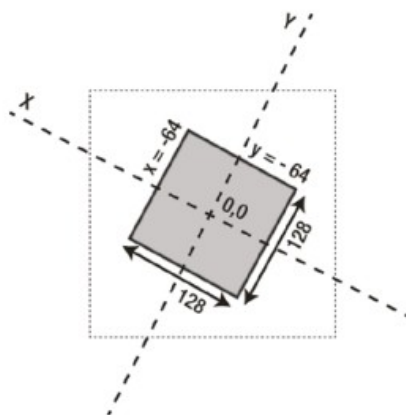
Apartado H. Sombras (`shadowColor`, `shadowOffsetX`, `shadowOffsetY`, `shadowBlur`)

Se pueden añadir sombras a cualquier línea o forma. Crea una imagen como la siguiente:

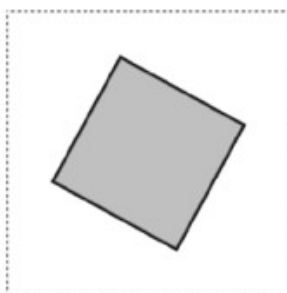
**Apartado I. Rotar (`save`, `translate`, `rotate`, `restore`)**

El API de canvas no incorpora un método para rotar un elemento de forma individual. Para rotar un elemento se hace rotar el lienzo.

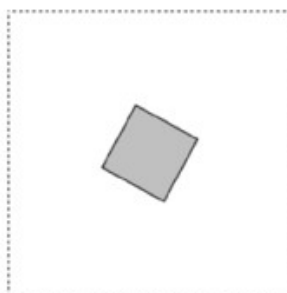
Rota el cuadrado creado anteriormente para obtener una imagen como la siguiente:

**Apartado J. Escalar (`scale`)**

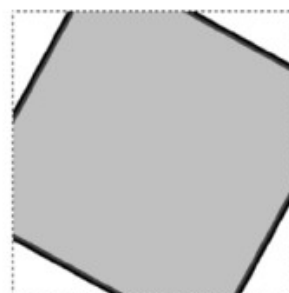
A partir del cuadrado creado anteriormente, crea un cuadrado de la mitad y otro del doble de su tamaño.



`ctx.scale(1, 1)`



`ctx.scale(0.5, 0.5)`

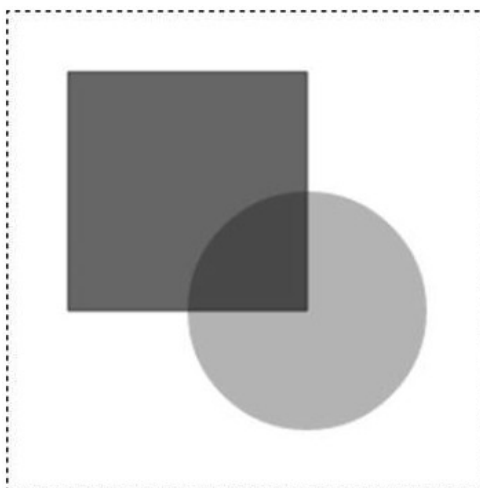


`ctx.scale(2, 2)`

Apartado K. Transparencia.

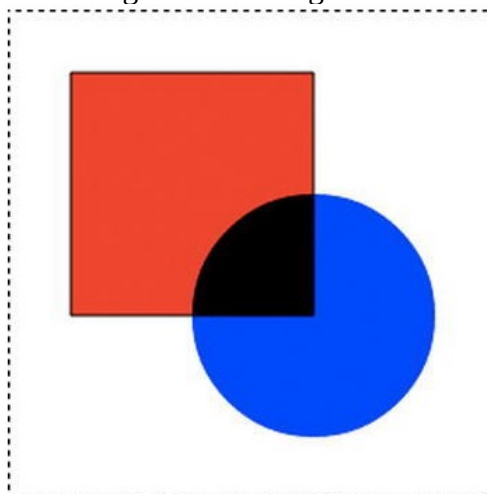
Hay dos formas de hacer transparente un elemento: usar colores RGBA o HSLA fijando el valor alpha a un número menor de 1 o utilizando la propiedad `globalAlpha` del contexto del canvas. Su uso es similar a `rotate` (ya que también afecta al canvas completo) por lo que será necesario encerrar la propiedad entre los métodos `save` y `restore`.

Crea las siguientes formas en el lienzo:

**Apartado L. Modos de mezcla (Blend Modes)**

La propiedad `globalCompositeOperation` del contexto del canvas determina como se deberían combinar los colores de dos formas o imágenes que se superponen.

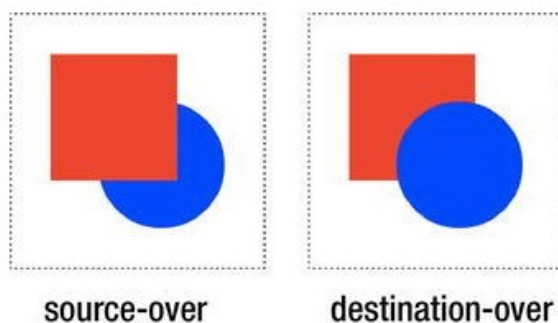
Hay 16 modos de mezcla. Crea una imagen como la siguiente utilizando el modo `multiple`:



Apartado M. Efectos de composición.

La propiedad `globalCompositeOperation` del contexto del canvas también proporciona control sobre cómo se combinan las formas que se superponen.

Hay 12 operaciones que se pueden aplicar (conocidas como operaciones Porter-Duff). Prueba las operaciones `source-over` y `destination-over` para obtener unas imágenes como las siguientes:

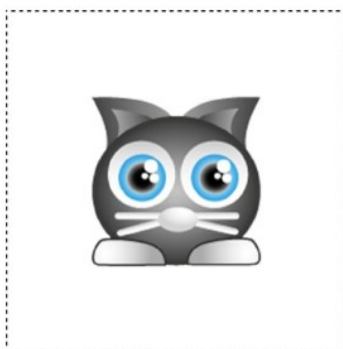
**Apartado N. Formas con imagen (createPattern)**

Es posible rellenar una forma con una imagen utilizando un patrón.

Crea un cuadrado con relleno utilizando la imagen `gato64.png`.

**Apartado Ñ. Imágenes (drawImage)**

Canvas permite añadir imágenes al lienzo sin utilizar ninguna forma. Añade al lienzo la imagen `gato64.png`.



Apartado O. Máscara de imagen (clip)

Una máscara es como un marco de ventana. Una máscara hará visible únicamente la parte que se encuentre dentro de su marco, la parte que quede fuera no se mostrará.

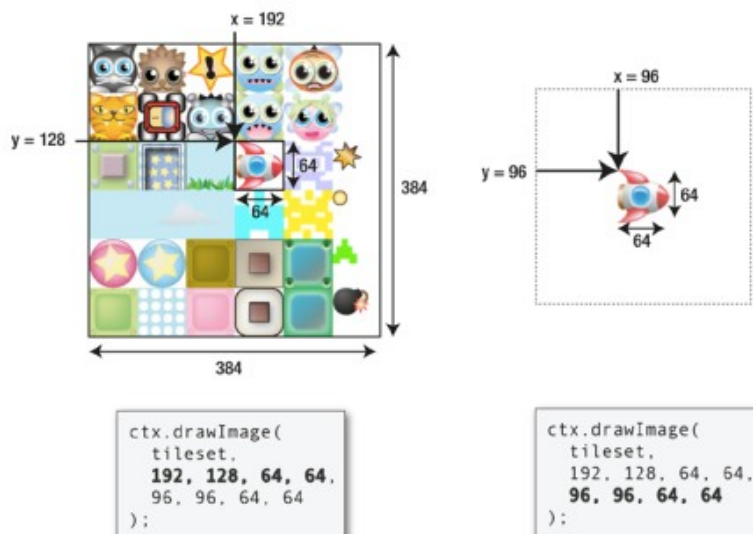
Crea la siguiente figura en el lienzo utilizando un círculo como máscara de la imagen.



Apartado P. Mostrar parte de una imagen (blitting)

Una práctica usual en el desarrollo de videojuegos es almacenar todos los personajes y objetos en un único fichero de imagen (conocido como tileset o sprite sheet) y recuperar cada elemento posicionando el canvas en la parte que se necesite. Así se consigue renderizar los gráficos de un juego de una forma rápida y realmente eficiente. Esta técnica se denomina *blitting*.

Muestra el cohete en el lienzo a partir del tileset proporcionado.



Apartado Q. Texto (measureText, textBaseLine, fillText)

Crea un texto como el siguiente en negrita, letras rojas y centrado en el lienzo.

