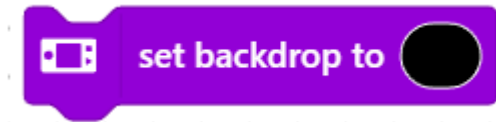


Trabajando con sprites en mblock

Para poder realizar este capítulo he tenido que poner como lenguaje el inglés, ya que si está como español no salen todos los bloques.

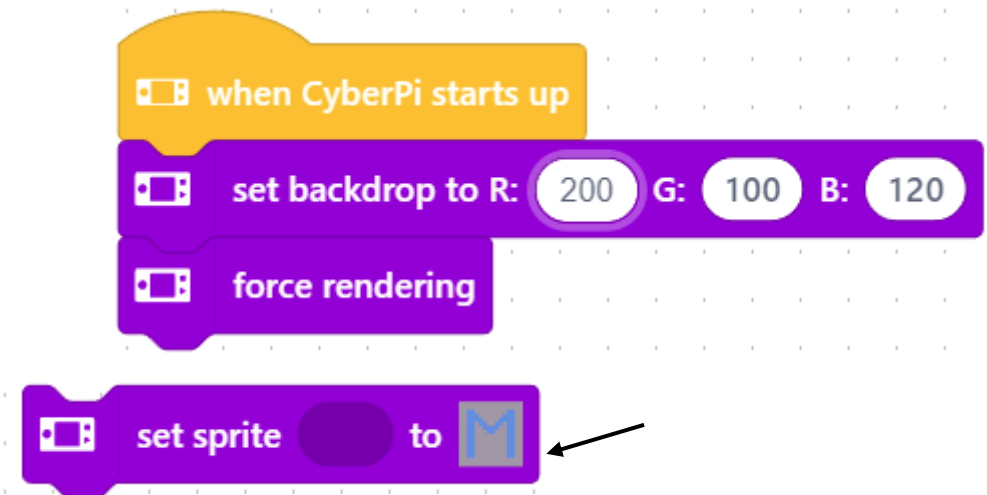
Para hacer los ejercicios tienes que cargarlos en el CyberPi ya que en vivo no funcionan.



Establece el color de fondo de la pantalla:

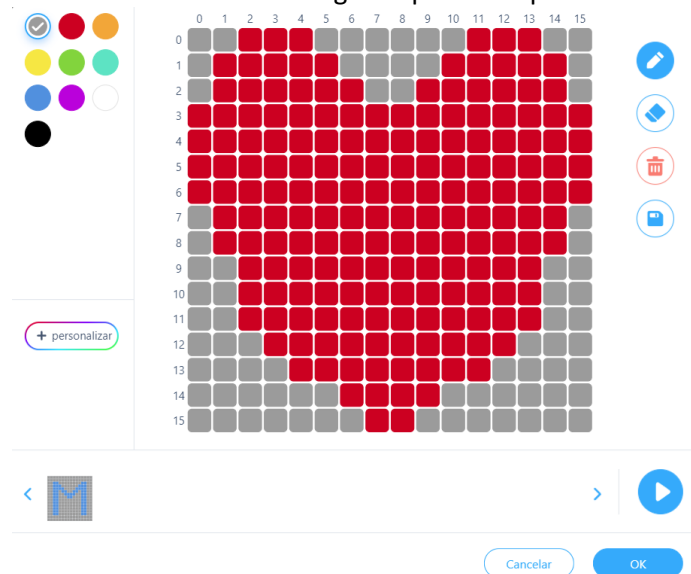


Establece el color de fondo con los valores RGB que son valores desde 0 a 255.



Definimos una variable y la establecemos como una imagen.

Al seleccionar el cuadrado donde está la imagen la podemos personalizar.



Esta por defecto se posiciona en medio de la pantalla.

```
when CyberPi starts up
  set sprite s_imagen to [heart icon]
  force rendering
```



```
set sprite [ ] to music ▼
```

Podemos imprimir una serie de iconos predefinidos.

```
when CyberPi starts up
  set sprite s_imagen to music ▼
  force rendering
```



```
set sprite [ ] to hello world
```

Crea un objeto y lo establece como texto.

```
when CyberPi starts up
  set sprite s_imagen to ¡Hola Mundo!
  force rendering

set sprite [ ] to QR code www.makeblock.com
```

Crea un objeto y lo establece en un código QR.

when CyberPi starts up

set sprite **s_imagen** to QR code www.peremanelv.com/robotica

force rendering



sprite **s_imagen** flips left-right ▼

Voltea el objeto de izquierda a derecha o de arriba abajo.

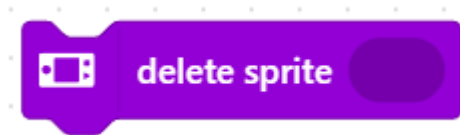
when CyberPi starts up

set sprite **s_imagen** to Pere Manel

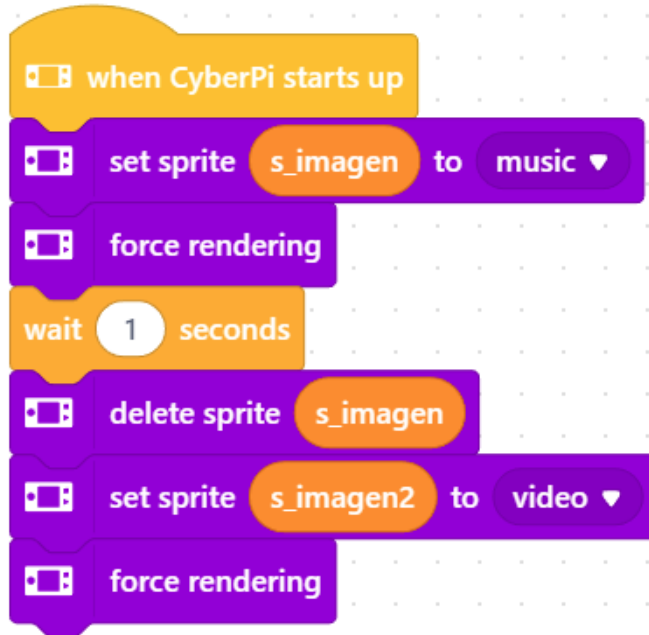
sprite **s_imagen** flips left-right ▼

force rendering

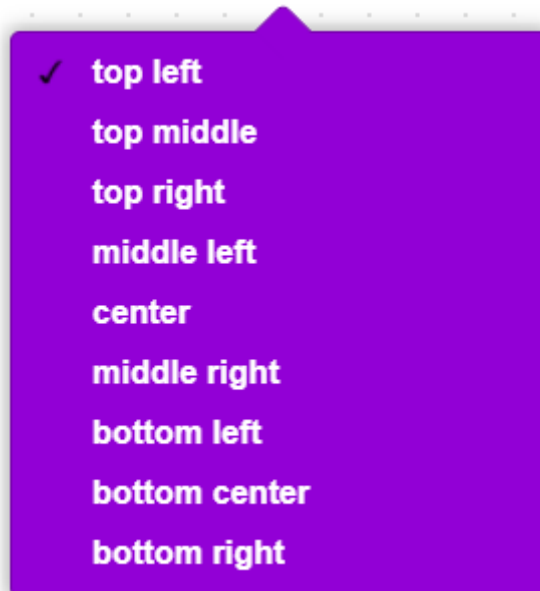


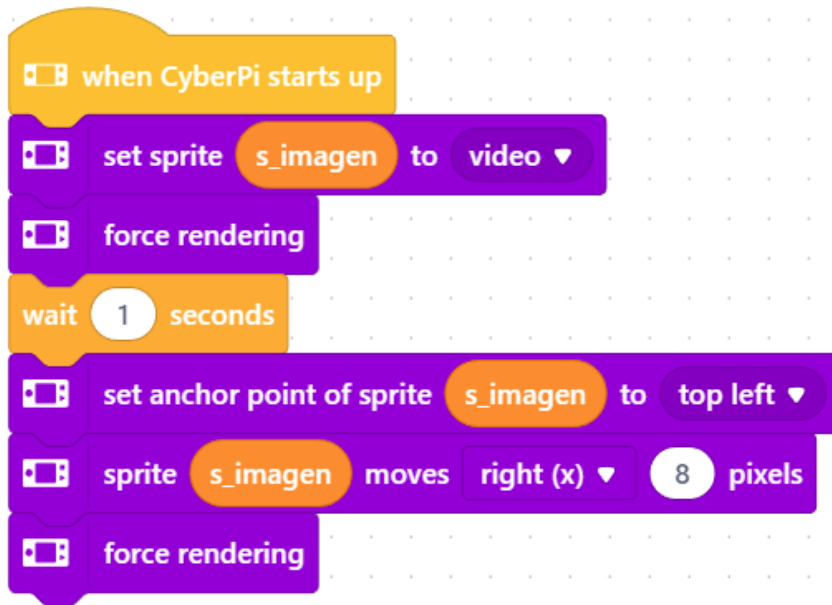


Borra un Sprite.

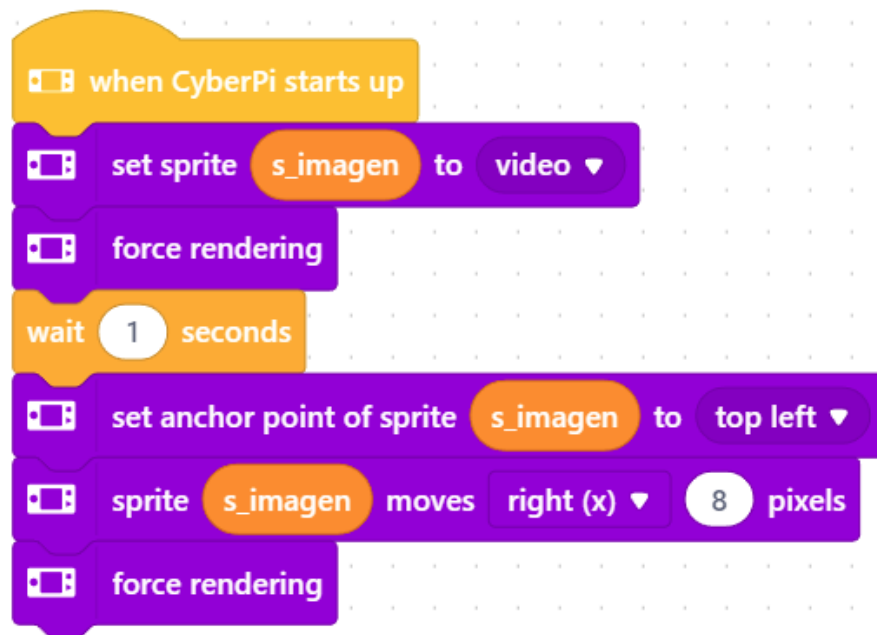


Establece el punto de anclaje del Sprite.





Después de cargar el programa en CyberPi y presionar el botón A, el ícono de **video** se muestra en el medio de la pantalla y, 1 segundo después, el ícono se mueve 8 píxeles hacia la derecha desde el punto superior izquierdo.



Después de cargar el programa en CyberPi y presionar el botón A, el ícono de **video** se muestra en el medio de la pantalla y, 1 segundo después, el ícono se mueve 8 píxeles hacia la derecha desde el punto superior izquierdo.

Vamos a realizar un pequeño programa para mover con las dirección del Joystick el Sprite.

```
when CyberPi starts up
  set sprite s_imagen to video
  force rendering
```

```
when joystick pulled←
  set anchor point of sprite s_imagen to top left
  sprite s_imagen moves right (x) -8 pixels
  force rendering
```

```
when joystick pulled→
  set anchor point of sprite s_imagen to top left
  sprite s_imagen moves right (x) 8 pixels
  force rendering
```

```
when joystick pulled↑
  set anchor point of sprite s_imagen to top left
  sprite s_imagen moves down (y) -8 pixels
  force rendering
```

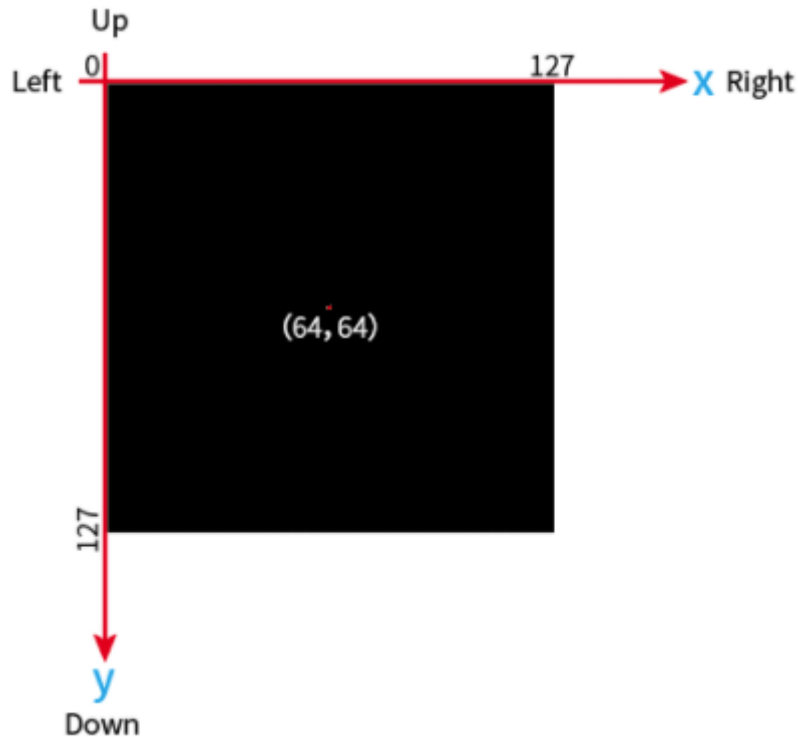
```
when joystick pulled↓
  set anchor point of sprite s_imagen to top left
  sprite s_imagen moves down (y) 8 pixels
  force rendering
```

```

sprite [ ] goes to x 64 y 64

```

Muevo el objeto especificado a un punto.



```

when CyberPi starts up
  set sprite s_imagen to video
  force rendering
  wait 1 seconds
  set anchor point of sprite s_imagen to top left
  sprite s_imagen goes to x 0 y 0
  force rendering

```

Que ponga el ancla en la parte superior izquierda.

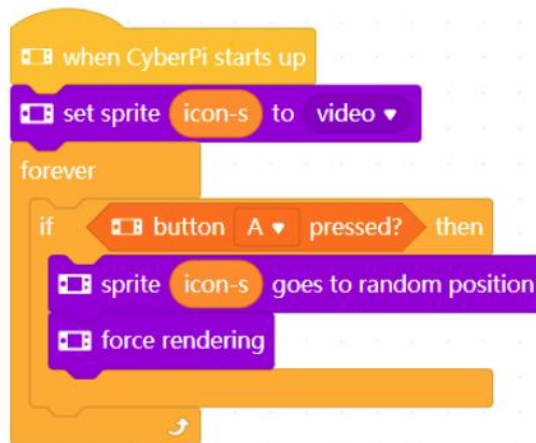
Después de cargar el programa en CyberPi y presionar el botón A, el ícono de **video** se muestra en el medio de la pantalla, y 1 segundo después, el ícono se mueve a la esquina superior izquierda de la pantalla, con el punto superior izquierdo ubicado en el punto (0,0).

```

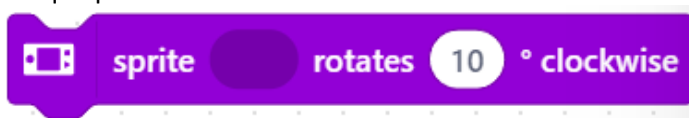
sprite [ ] goes to random position

```

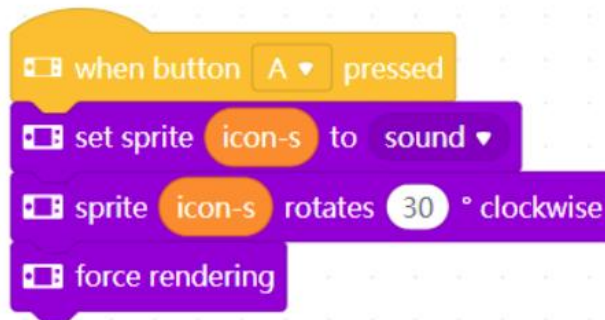
Muevo el objeto especificado de forma aleatoria.



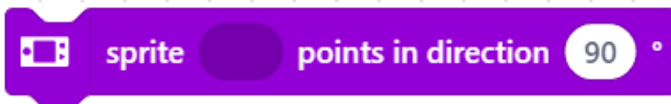
Después de cargar el programa en CyberPi, el ícono de **video** se mueve aleatoriamente cada vez que presiona el botón A.



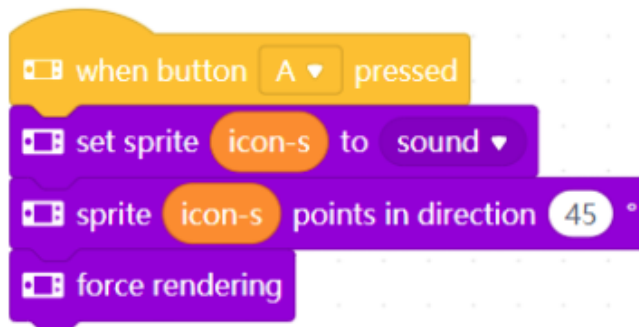
Gira el Sprite el número de grados especificado.



Después de cargar el programa en CyberPi, el ícono de **sonido** gira 30 grados en el sentido de las agujas del reloj cada vez que presiona el botón A.



Establece la dirección en grados que enfrentará el Sprite.



Después de cargar el programa en CyberPi, el ícono de **sonido** que mira en la dirección de 45 grados se muestra en el centro de la pantalla cuando presiona el botón A.



Cambias el tamaño del Sprite.


```

when CyberPi starts up
  set sprite s_imagen to video
  set sprite s_imagen size to 200 %
  force rendering
  set sprite [ ] size to 200 %

```

Cambias el color del Sprite

```

when CyberPi starts up
  set sprite s_imagen to video
  set sprite s_imagen color to [blue]
  force rendering
  set sprite [ ] color to R: 255 G: 255 B: 255

```

Cambiar el color del Sprite con los valores RGB

```

reset sprite [ ] to default color

```

Restablece el color por defecto.

```

show [ ] sprite [ ]

```

Muestra u oculta un Sprite.

```

sprite [ ] goes to front layer

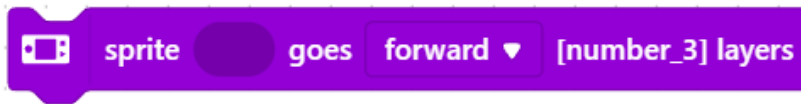
```

Trae el Sprite especificado al frente o atrás.

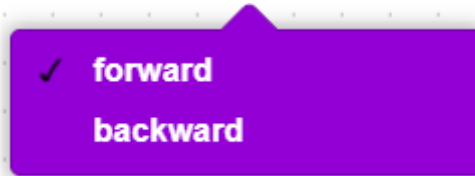
```

[ ]
  ✓ front
  back

```



Trae el Sprite especificado hacia adelante o lo envía hacia atrás en una capa.



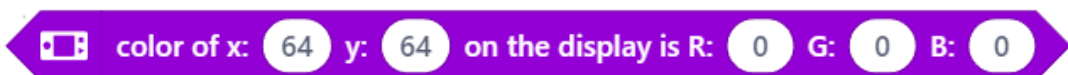
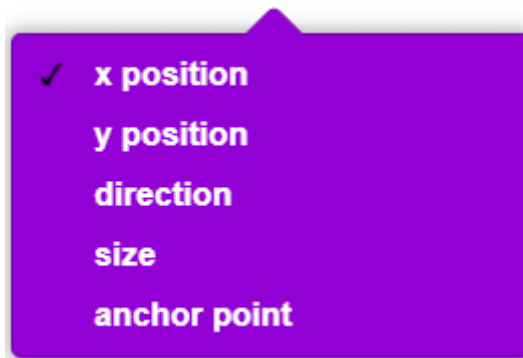
Determina si dos Sprites especificados entran en contacto entre sí.



Determina si el objeto especificado toca el borde de la pantalla.



Este bloque te permite obtener las coordenadas x, la coordenada y, la dirección, en tamaño o el punto de anclaje.



Determina si el punto especificado (x, y) se muestra en el color especificado.



Renderiza el objeto especificado para que surtan efecto.

Ejemplo práctico:

Queremos hacer un proyecto que será una pelota que se tiene que mover según la inclinación que le hagamos, si toca a los bordes se tiene que reproducir un sonido.

En la siguiente página está la solución.

```
cuando CyberPi se inicia
  set sprite s_pelota to [ball]
  sprite s_pelota goes to x 64 y 64
  force rendering
  forever
    if [ángulo de inclinación hacia adelante] > 10 then
      sprite s_pelota moves down (y) -1 pixels
      force rendering
      if sprite s_pelota on edge? then
        sprite s_pelota moves down (y) 5 pixels
        force rendering
    if [ángulo de inclinación hacia atrás] > 10 then
      sprite s_pelota moves down (y) 1 pixels
      force rendering
      if sprite s_pelota on edge? then
        sprite s_pelota moves down (y) -5 pixels
        force rendering
    if [ángulo de inclinación hacia la derecha] > 10 then
      sprite s_pelota moves right (x) 1 pixels
      force rendering
      if sprite s_pelota on edge? then
        sprite s_pelota moves right (x) -5 pixels
        force rendering
    if [ángulo de inclinación hacia la izquierda] > 10 then
      sprite s_pelota moves left 1 pixels
      force rendering
      if sprite s_pelota on edge? then
        sprite s_pelota moves left -5 pixels
        force rendering
```

