

Tema	MOVIMIENTOS DEL SPRITE	
Descripción de la clase	El alumno aprenderá a mover el sprite usando las propiedades de velocidad del sprite. También creará sprites de borde en el juego y hará que rebote en el sprite principal desde los bordes del lienzo. Además, el alumno aprenderá a crear abstracciones en su código declarando funciones.	
Clase	PRO-C2	
Duración de la clase	50 minutos	
Objetivo	<ul style="list-style-type: none"> • Escribir funciones personalizadas para animar y mover la pelota en diferentes direcciones. • Escribir una función para crear bordes para el lienzo, de modo que la pelota rebote en los bordes. 	
Recursos requeridos	<ul style="list-style-type: none"> • Recursos para maestras: <ul style="list-style-type: none"> ○ Iniciar sesión en Code.org. ○ Laptop con conectividad a internet. ○ Auriculares con micrófono. ○ Cuaderno y bolígrafo. ○ Teléfono inteligente. • Recursos para alumnos: <ul style="list-style-type: none"> ○ Iniciar sesión en Code.org. ○ Laptop con conectividad a internet. ○ Auriculares con micrófono. ○ Cuaderno y bolígrafo. 	
Estructura de la clase	Diapositivas de Rompiendo el Hielo Actividad dirigida por la maestra 1 Actividad dirigida por el alumno 1 Actividad dirigida por la maestra 2 Actividad dirigida por el alumno 2 Conclusión	10 minutos 10 minutos 5 minutos 10 minutos 10 minutos 5 minutos
ROMPIENDO EL HIELO - 10 minutos		




La maestra inicia la presentación de diapositivas desde la diapositiva 1 a la 9
Consulta las notas en tu pantalla y sigue las instrucciones de cada diapositiva.

Acción de la maestra	Acción del alumno
<p>¿Cómo has estado? ¿Estás emocionado de aprender algo nuevo?</p> <p><i>Ejecuta la presentación desde la diapositiva 1 a la diapositiva 3.</i></p> <p>Los siguientes son los entregables de la sesión de Rompiendo el hielo.</p> <ul style="list-style-type: none"> • Conecta al alumno con la clase anterior. • Sesión del quiz de inicio. 	<p>REA: Respuesta variada.</p> <p>Haz clic en la pestaña de presentación de diapositivas y comienza a mostrarlas.</p>

PyR: haz clic en el quiz de la clase

Pregunta	Respuesta
<p>¿Cuál de los siguientes es el código correcto para establecer la animación de cohete a var rocket1?</p>  <p>A. <code>rocket1.setImage("rocket1");</code></p>	<p>C</p>

<p>B. <code>rocket1 = setAnimation("rocket1");</code> C. <code>rocket1.setAnimation("rocket1");</code> D. <code>rocket1 = setImage("rocket1");</code></p>	
<p>¿Cuál de los siguientes es el código correcto para reducir el tamaño de la animación del cohete?</p>  <p>A. <code>rocket1.scale = 25;</code> B. <code>rocket1.size = .25;</code> C. <code>rocket1.size = 0.25;</code> D. <code>rocket1.scale = 0.25;</code></p>	D
<p align="center">Continúa la sesión de Rompiendo el hielo</p>	
<p align="center">Detalles de la actividad</p>	<p align="center">Solución/Lineamientos</p>
<p><i>Ejecuta la presentación desde la diapositiva 4 a la diapositiva 9 para establecer el planteamiento del problema.</i></p> <p>Los siguientes son los entregables de la sesión Rompiendo el hielo:</p> <ul style="list-style-type: none"> • Objetos y propiedades del Sprite. • Uso de funciones. 	<p>Narra las diapositivas mediante el uso de gestos con las manos y métodos de modulación de voz para atraer más interés en los alumnos.</p>



La maestra finaliza la presentación de diapositivas

ACTIVIDAD DIRIGIDA POR LA MAESTRA 1 - 10 minutos

La maestra comienza a compartir su pantalla

DESAFÍO

- Aprende a escribir funciones.
- Aprende a escribir llamadas a funciones, usando el método de llamada por valor.

Acción de la maestra	Acción del alumno
<p>Recuerda que en la última clase aprendimos cómo declarar una variable y asignar valores a estas variables. También usamos ese concepto para asignar valores a diferentes propiedades del sprite. ¿Puedes nombrar algunas de las propiedades del sprite?</p> <p>¡Estupendo! Te acuerdas de la mayoría de ellas.</p> <p>En la última clase usamos algunas funciones predefinidas como:</p> <p>createSprite(): para crear un sprite rectangular.</p> <p>setAnimation(): para configurar la animación en el sprite.</p> <p>drawSprites(): para dibujar los sprites en el lienzo.</p> <p>Hoy aprenderemos cómo hacer nuestra propia función personalizada para rotar un sprite 10 grados.</p> <p><i>La maestra abre el enlace de la actividad y explica el código repetitivo que tiene dos sprites: un sprite de fantasma y un sprite de helicóptero. Actividad de la maestra 1.</i></p>	<p>REA: Sí.</p> <p>REA: setAnimation, shapeColor, x, y, scale.</p> <p>REA: Variado.</p>
Código:	

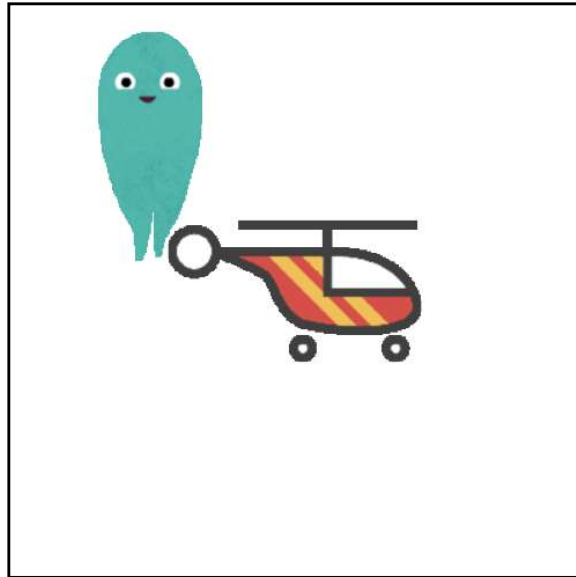
Espacio de trabajo:

```

1 var sprite = createSprite(100, 100, 20, 20);
2 sprite.setAnimation("ghost_standing_1");
3
4 var sprite2 = createSprite(200, 200, 20, 20);
5 sprite2.setAnimation("helicopter_1");
6
7
8 function draw() {
9   drawSprites();
10 }

```

Output:



El primer paso para usar una función es '**Definir la función**'.

- 1) Ve a **Functions** (*funciones*) y selecciona el bloque 'Define una función' en el área del **Espacio de trabajo**.

Espacio de trabajo:

```

1  var sprite = createSprite(100, 100, 20, 20);
2  sprite.setAnimation("ghost_standing_1");
3
4  var sprite2 = createSprite(200, 200, 20, 20);
5  sprite2.setAnimation("helicopter_1");
6
7
8  function draw() {
9      drawSprites();
10 }
11
12 function miFunción() {
13
14 }

```

Cambia el nombre de la función: el nombre de la función debe ser algo que describa su funcionalidad.

Ya que estamos escribiendo una función para rotar el sprite, debemos nombrar nuestra función similar a la funcionalidad, es decir, **rotateSprite()** (*rotar sprite*).

A continuación, debemos pensar si la función requiere alguna entrada. Por ejemplo, la función **setAnimation()** (*configurar la animación*) necesita el nombre de la animación que desees configurar para tu sprite; es decir, fantasma, voleibol o helicóptero.

Para la función **rotateSprite()** (*rotar sprite*) tendremos que decirle a la computadora qué sprite se va a rotar, por lo que debería ir como entrada a la función.

Siempre pasamos la entrada como parámetro en la definición de la función.

La maestra escribe el código dentro de la función para cambiar la rotación del sprite en 10.

Hemos definido con éxito la función **rotateSprite()**, pero para usarla tendremos que llamarla en nuestro programa.

Podemos llamar a la función por su nombre y pasar la entrada entre paréntesis.

Sintaxis: **function_name(parameter);**

La maestra realiza los cambios en el código para llamar a la función de rotación dos veces, una por cada objeto presente en el programa.

La maestra ejecuta el programa y muestra el output al alumno.

El alumno mira la consola del navegador y verifica el output del código escrito por la maestra.

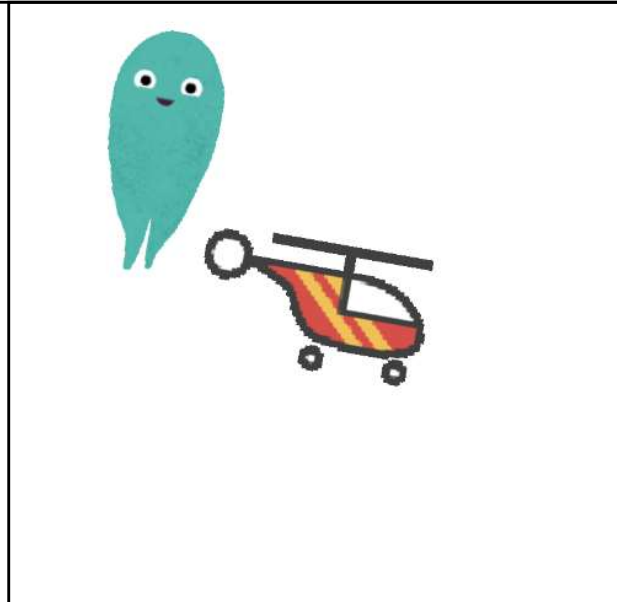
Código:

```

Espacio de trabajo:
1  var sprite = createSprite(100, 100, 20, 20);
2  sprite.setAnimation("ghost_standing_1");
3
4  var sprite2 = createSprite(200, 200, 20, 20);
5  sprite2.setAnimation("helicopter_1");
6
7  rotateSprite(sprite);
8  rotateSprite(sprite2);
9
10 function draw() {
11   drawSprites();
12 }
13
14 function rotateSprite(sprite) {
15   sprite.rotation = sprite.rotation + 10;
16   //sprite.rotation +=10;
17 }

```

Output:



¿Te diste cuenta?: Acabamos de instruir a la computadora una vez sobre cómo rotar el sprite, y podemos usarlo varias veces para diferentes sprites sin escribir las mismas líneas de código una y otra vez. ¿No es increíble?

REA: Sí.

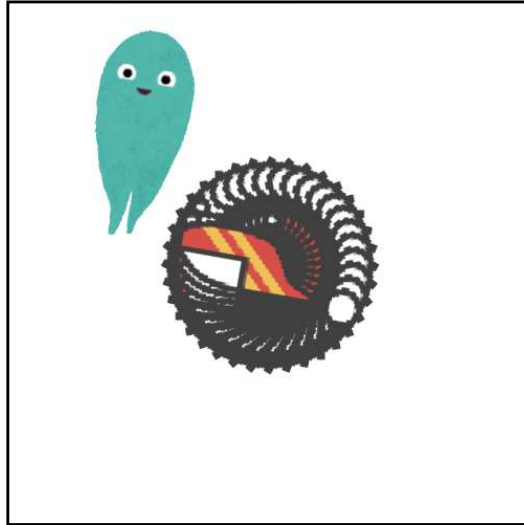
Las funciones se pueden llamar repetidamente para hacer el mismo conjunto de instrucciones en diferentes argumentos. Nos ayuda a evitar la repetición de código y, por lo tanto, acorta nuestro código.

Intentemos llamar a nuestra función dentro de la función **draw()** que se ejecuta repetidamente.

Código:

```
4 var sprite2 = createSprite(200, 200, 20, 20);
5 sprite2.setAnimation("helicopter_1");
6
7 rotateSprite(sprite);
8 rotateSprite(sprite2);
9
10 function draw() {
11     rotateSprite(sprite2);
12     drawSprites();
13 }
```

Output:



¿Viste que ahora el sprite del helicóptero sigue girando repetidamente?

Porque ahora estamos llamando a la función **rotateSprite()** dentro de **draw()**.

También llamamos a estas funciones personalizadas funciones “**definidas por el usuario**”, porque hemos creado y personalizado estas funciones desde cero.

Tengo un desafío para ti ahora. ¿Puedes escribir una función para escalar el objeto a la mitad del tamaño original?

REA: Sí.

La maestra deja de compartir su pantalla

ACTIVIDAD DIRIGIDA POR EL ALUMNO 1 - 5 minutos

Ahora es tu turno. Comparte tu pantalla conmigo.

- Pídele al alumno que presione la tecla **ESC** para volver al panel.
- Guía al alumno para que comience a compartir la pantalla.
- La maestra cambia a pantalla completa.

DESAFÍO

- El alumno escribe la función con parámetros para cambiar el tamaño de los sprites.
- El alumno aprende a llamar a la función en el programa.



La maestra inicia la presentación de diapositivas en la diapositiva 10.
Consulta las notas en tu pantalla y sigue las instrucciones de cada diapositiva.

Acción de la maestra	Acción del alumno
<p><i>Guía al alumno a abrir el enlace de la actividad del alumno 1 e inicia sesión en code.org si aún no lo has hecho.</i></p> <ol style="list-style-type: none"> 1) Crea una nueva función y asígnale un nombre apropiado. En este caso, resizeSprite (cambiar el tamaño del sprite). 2) Agrega los parámetros de entrada a la nueva función, es decir, sprite. 3) Llama a la nueva función en el código por su nombre de función y sus entradas. resizeSprite(sprite). 4) Agrega el código para llamar a esta función varias veces en el programa con diferentes entradas. <p><i>Guía al niño y comprueba si hay errores tipográficos. Anima al niño a que lo haga solo. Haz preguntas para el siguiente paso, si el niño se atasca en algún lugar.</i></p> <p><i>La apreciación y el refuerzo positivo ayudarán al niño a ganar confianza para programar.</i></p>	<p><i>El alumno hace clic en el Enlace de la actividad del alumno 1 y comienza a escribir el código de su primera función personalizada para cambiar el tamaño del objeto.</i></p>
<p>Código:</p>	

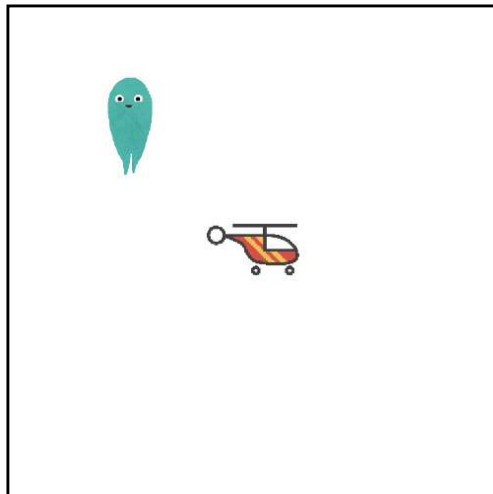
Espacio de trabajo:

```

1  var sprite = createSprite(100, 100, 20, 20);
2  sprite.setAnimation("ghost_standing_1");
3
4  var sprite2 = createSprite(200, 200, 20, 20);
5  sprite2.setAnimation("helicopter_1");
6
7  resizeSprite(sprite);
8  resizeSprite(sprite2);
9
10 function draw() {
11   drawSprites();
12 }
13
14 function resizeSprite(sprite){
15   sprite.scale = 0.5;
16 }
17

```

Output:



Hiciste un trabajo fabuloso. Tengo mucha confianza en ti. Será un gran viaje juntos.

La maestra guía al alumno para dejar de compartir la pantalla

ACTIVIDAD DIRIGIDA POR LA MAESTRA 2 - 10 minutos

DESAFÍO

- Aprende a agregar la velocidad al objeto de fantasma.
- Aprende a crear bordes e introduce el evento de pulsar el mouse.



La maestra inicia la presentación de diapositivas desde la diapositiva 11 a la 13.

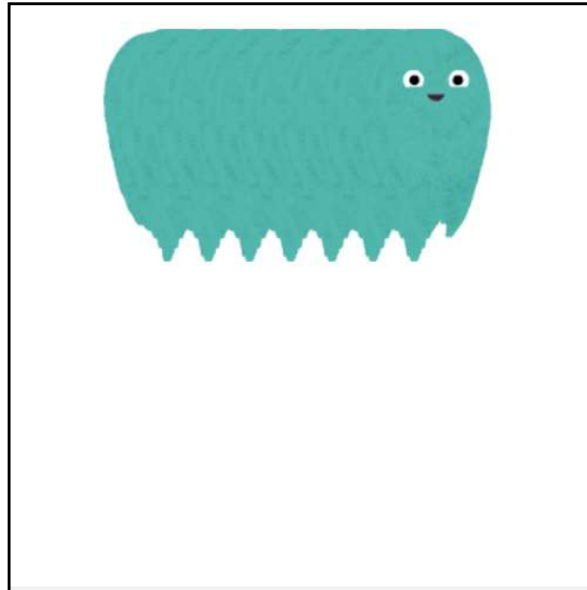
Consulta las notas en tu pantalla y sigue las instrucciones de cada diapositiva.

Acción de la maestra	Acción del alumno
<p>En la última clase, aprendimos cómo crear un sprite, configurar la animación del sprite y jugar con algunas propiedades del sprite. Hoy aprenderemos cómo darle movimiento a un sprite en el juego.</p> <p><i>La maestra hace clic en el enlace de la actividad de la maestra 2 y abre el código de la clase anterior.</i></p>	<p><i>El alumno observa.</i></p>
<p>Actualmente, el sprite es estático.</p> <p>¿Lo movemos dándole rapidez?</p> <p>En ciencia y programación, la rapidez también se conoce como velocidad.</p> <p>¿Puedes encontrar una propiedad de la lista que pueda ayudarnos a darle algo de velocidad/rapidez al sprite?</p> <p><i>La maestra ayuda al alumno a encontrar velocityX y velocityY. La maestra asigna un valor positivo a velocityX primero y ejecuta el código.</i></p>	<p>REA: sí.</p> <p>REA: sí</p> <p>REA: sí</p> <p><i>El alumno observa el código.</i></p>
<p>Código:</p>	

Espacio de trabajo:

```
1 var sprite = createSprite(100, 100, 20, 20);
2 sprite.setAnimation("ghost_standing_1");
3
4 sprite.velocityX = 2;
5
6 function draw() {
7   drawSprites();
8 }
```

Output:



¿Qué crees que pasó?

En realidad, el sprite se está moviendo, pero dibujándose continuamente en diferentes posiciones. ¿Recuerdas que te dije que la función **draw()** sigue ejecutándose hasta que termina el juego?

¿Qué debemos hacer para ver solo el último sprite dibujado en el lienzo/pantalla?

Sí, debemos limpiar la pantalla antes de dibujar el sprite en una nueva posición.

REA: El sprite está aumentando de tamaño.

REA: Sí.

REA: Limpia la pantalla antes de dibujar el nuevo objeto.

Probemos eso. Vamos a la pestaña **'Drawing'** (*dibujando*), en la **Caja de herramientas** y verificamos la función **background()** (*fondo*).

La maestra escribe el código para borrar la pantalla después de cada pasada haciendo que el fondo sea blanco en cada pasada.

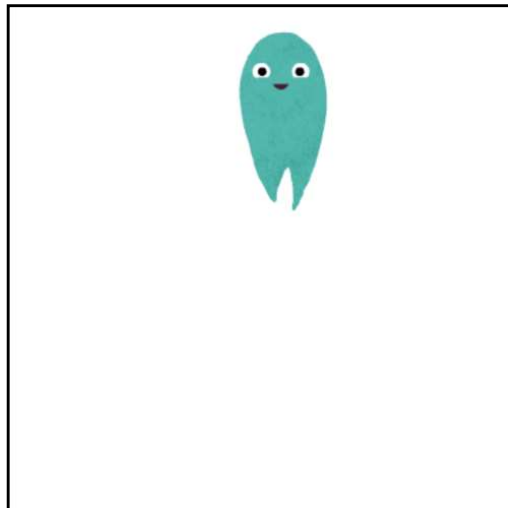
Código:

```

Espacio de trabajo:
1  var sprite = createSprite(100, 100, 20, 20);
2  sprite.setAnimation("ghost_standing_1");
3
4  sprite.velocityX = 2;
5
6  function draw() {
7    background("white");
8    drawSprites();
9  }

```

Output:



¡Estupendo! Ahora el sprite se mueve horizontalmente hacia la derecha.

Intentemos asignar **velocityY** en lugar de **velocityX**. ¿Qué piensas tú que sucederá?

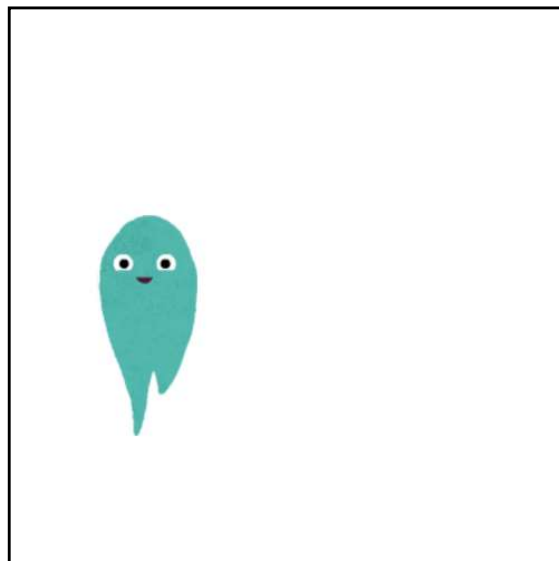
La maestra edita y ejecuta el código.

REA: El objeto se moverá verticalmente en lugar de horizontalmente.

Código:

Espacio de trabajo:	
1	<code>var sprite = createSprite(100, 100, 20, 20);</code>
2	<code>sprite.setAnimation("ghost_standing_1");</code>
3	
4	<code>sprite.velocityY = 2;</code>
5	
6	<code>function draw() {</code>
7	<code> background("white");</code>
8	<code> drawSprites();</code>
9	<code>}</code>

Output:



El sprite ahora se mueve verticalmente hacia abajo. ¿Qué crees que pasará si le damos tanto **velocityX** como **velocityY**? ¡Sí! El objeto se moverá tanto en dirección horizontal como vertical. Parecerá que se mueve en diagonal.

REA:

El objeto comienza a moverse en diagonal.

La maestra escribe y ejecuta el código.

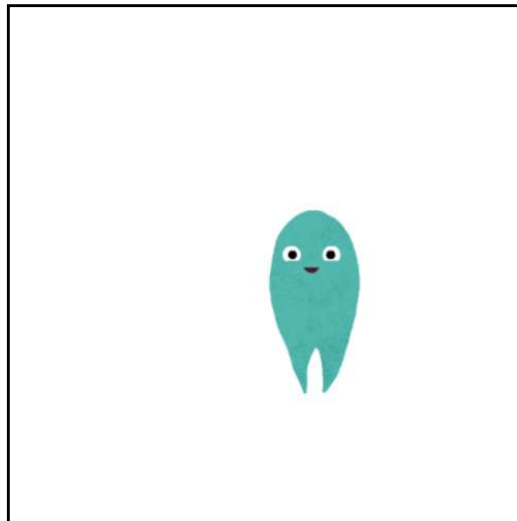
Código:

```

Espacio de trabajo:
1  var sprite = createSprite(100, 100, 20, 20);
2  sprite.setAnimation("ghost_standing_1");
3
4  sprite.velocityX = 2;
5  sprite.velocityY = 2;
6
7  function draw() {
8    background("white");
9    drawSprites();
10 }

```

Output:



Tenemos nuestro sprite moviéndose en la esquina inferior derecha. ¿Cómo movemos nuestro sprite en la dirección opuesta horizontalmente?

*La maestra cambia **velocityX** del sprite de 2 a -2 y ejecuta el código.*

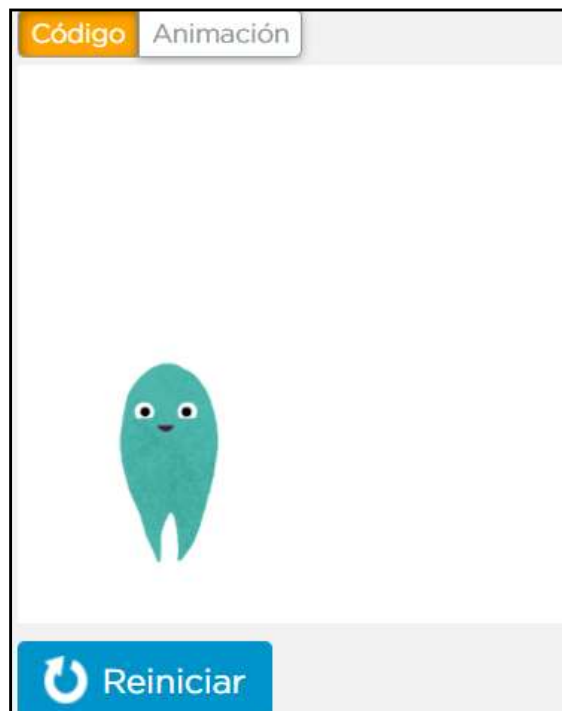
REA: Dando un valor negativo a **velocityX**.

Código:

Espacio de trabajo:

```
1 var sprite = createSprite(200, 200, 20, 20);
2 sprite.setAnimation("ghost_standing_1");
3
4 sprite.velocityX = -2;
5 sprite.velocityY = 2;
6
7 function draw() {
8   background("white");
9   drawSprites();
10 }
```

Output:



De manera similar, podemos mover nuestro sprite verticalmente hacia arriba en lugar de hacia abajo asignando un valor negativo a **velocityY**.

La maestra puede pedirle al niño que elija los valores, tanto para la velocidad como para predecir el movimiento y probarlo.

También podemos aumentar/disminuir la velocidad en una de las direcciones, lo que hace que el objeto se mueva más rápido en una dirección.

Por ejemplo, si la velocidad en la dirección x es mayor, el sprite

se moverá horizontalmente más rápido.

De manera similar, si aumentamos el valor de **velocityY** y disminuimos **velocityX**, el sprite se moverá verticalmente más rápido.

¿Lo intentamos?

REA: Sí

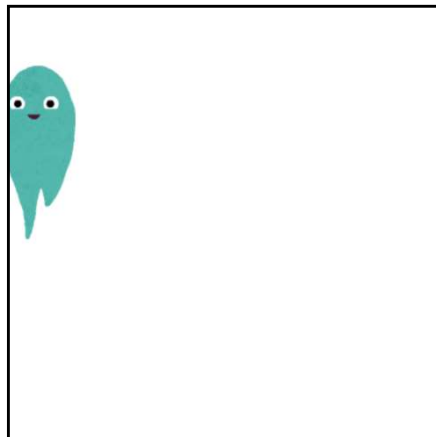
CÓDIGO:

```

Espacio de trabajo:
1  var sprite = createSprite(100, 100, 20, 20);
2  sprite.setAnimation("ghost_standing_1");
3
4  sprite.velocityX = -2;
5  sprite.velocityY = 1;
6
7  function draw() {
8    background("white");
9    drawSprites();
10 }

```

OUTPUT:



¡Estupendo! Parece que ahora podemos mover el sprite a donde queramos.

Ahora que nuestro sprite ha comenzado a moverse. ¿Notas algún problema en el juego?

Para solucionar esto, creamos bordes en la pantalla.

¿Encuentras algo en la pestaña de sprites para crear bordes en el lienzo?

REA: Sí, el sprite sale de la pantalla.

REA: **createEdgeSprites()** (crear sprites de borde).

`createEdgeSprites()`

La maestra agrega "**createEdgeSprites()**" al código y le pregunta al alumno: ¿por qué debemos agregar el código fuera de **draw()**?

La maestra ejecuta el código y muestra el output.

REA: Porque necesitamos crear los límites una vez. Todo lo que escribimos dentro de la función **draw()** se repite una y otra vez.

El alumno observa la pantalla y aprende el código.

CÓDIGO:

Espacio de trabajo:

```

1 var sprite = createSprite(100, 100, 20, 20);
2 sprite.setAnimation("ghost_standing_1");
3
4 sprite.velocityX = -2;
5 sprite.velocityY = 1;
6
7 createEdgeSprites();
8
9 function draw() {
10   background("white");
11   drawSprites();
12 }
```

¡Oh! El fantasma todavía no rebota en los límites/bordes.
¿Por qué?

Hicimos sprites de borde, pero no le pedimos a la computadora que hiciera que el sprite rebotara en los bordes de los límites. Recuerda que una computadora solo hace lo que le decimos. ¿Puedes buscar un **bounceOff(target)** (rebotar en) en la pestaña **Sprite**?

La maestra agrega la instrucción **bounceOff** en el código y cambia el objetivo a bordes.

¿Puedes decirme por qué agregamos la instrucción **bounceOff()** en la función **draw()**?

REA: Variada

REA: Sí.

REA: Porque queremos que el objeto **rebote en** las paredes/bordes a lo largo del juego.

La maestra debe ayudar al niño con la explicación, si el niño no puede contestar.

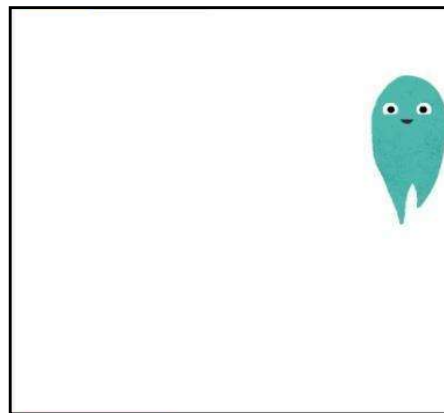
CÓDIGO:

```

Espacio de trabajo:
1  var sprite = createSprite(100, 100, 20, 20);
2  sprite.setAnimation("ghost_standing_1");
3
4  sprite.velocityX = -2;
5  sprite.velocityY = 1;
6
7  createEdgeSprites();
8
9  function draw() {
10   background("white");
11   sprite.bounceOff(edges);
12   drawSprites();
13 }
14
15

```

OUTPUT:



La maestra deja de compartir la pantalla

ACTIVIDAD DIRIGIDA POR EL ALUMNO 2 - 10 minutos

Ahora es tu turno. Comparte tu pantalla conmigo.

- Pídele al alumno que presione la tecla ESC para volver al panel.
- Guía al alumno para que comience a compartir la pantalla.
- La maestra cambia a pantalla completa.

DESAFÍO

- Haz que la pelota rebote en los bordes.
- Agrega un evento de pulsación del mouse. La pelota debe moverse o ganar velocidad solo cuando se hace clic en el botón del mouse.



La maestra puede mostrar la presentación de diapositivas desde la diapositiva 14 a la 15. Consulta las notas en tu pantalla y sigue las instrucciones de cada diapositiva.

Acción de la maestra	Acción del alumno
<p><i>Pídele al alumno que haga clic en la Actividad del alumno 2 y luego haga clic en Remix en el archivo.</i></p> <p>En la última clase, habías creado el sprite de pelota para el juego. Agreguemos movimiento a nuestro sprite de pelota.</p> <p><i>Guía al alumno para que asigne velocidad a la pelota.</i></p>	<p><i>El alumno escribe el código sobre la etapa actual del juego, para asignar la velocidad al sprite.</i></p> <p><i>Ejecuta el código para observar el output:</i></p>

CÓDIGO:

Espacio de trabajo:

```

1  var ball;
2
3  ball = createSprite(100,100,20,20);
4  ball.setAnimation("volleyball12_1");
5  ball.scale = 0.1;
6
7  ball.velocityX = 4;
8  ball.velocityY = 2;
9
10 function draw() {
11   background("white");
12   drawSprites();
13 }

```

OUTPUT:



Guía al alumno para que haga que la pelota rebote en los bordes, como se muestra en el siguiente fragmento de código:

Observa al alumno, verifica que no haya errores tipográficos. Oriéntalo para que escriba correctamente el código.

El alumno escribe el código para hacer que la pelota rebote desde los bordes.

CÓDIGO:


```


Espacio de trabajo:
1  var ball;
2
3  ball = createSprite(100,100,20,20);
4  ball.setAnimation("volleyball2_1");
5  ball.scale = 0.1;
6
7  ball.velocityX = 4;
8  ball.velocityY = 2;
9
10 createEdgeSprites();
11
12
13 function draw() {
14   background("white");
15   drawSprites();
16
17   ball.bounceOff(edges);
18 }
  
```

OUTPUT:

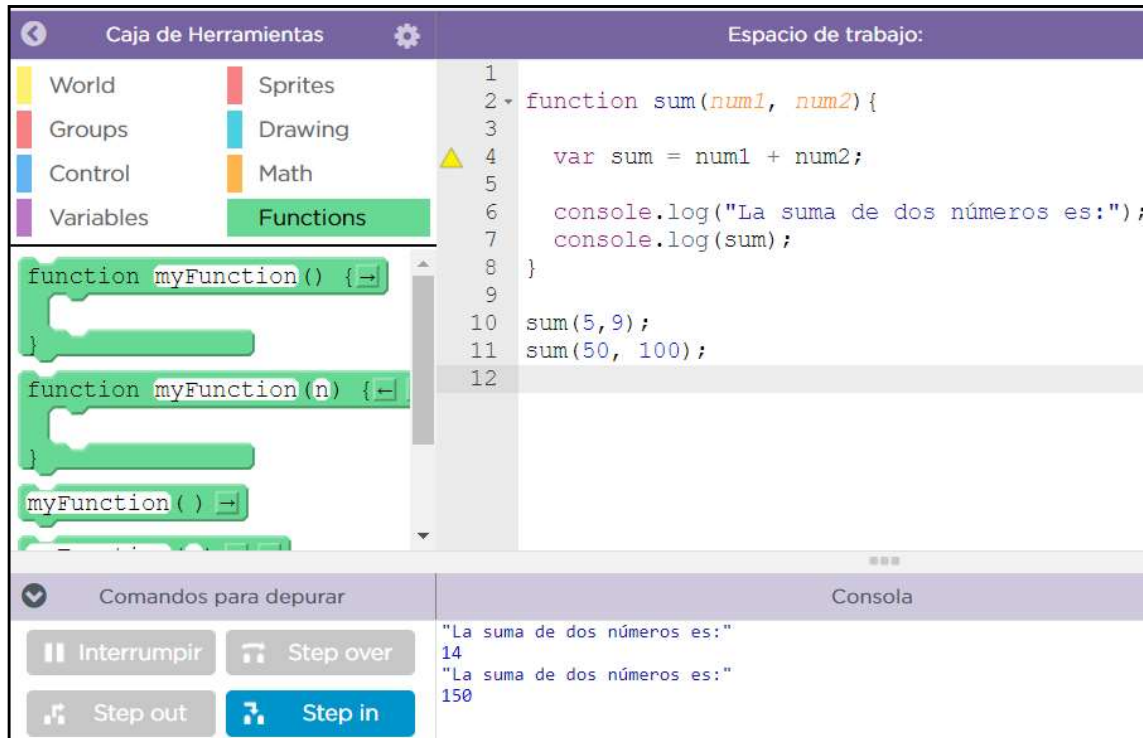
<div></div>		<div></div>
<p>¡Estupendo! Ahora nuestra pelota rebota en los bordes. ¿Ves que la pelota comienza a moverse tan pronto como ejecutamos el programa?</p> <p>¡Increíble! Has completado con éxito el desafío de hoy. Estoy muy impresionada.</p> <p>En la siguiente clase crearemos un sprite de paleta para nuestro juego y lo moveremos usando nuestro mouse.</p>	<p>REA: Sí.</p>	
<p>Guía al alumno para que deje de compartir la pantalla</p>		
<p>CONCLUSIÓN - 5 minutos</p>		
<p><u>RETROALIMENTACIÓN</u></p> <ul style="list-style-type: none">• Aprecia y felicita al alumno por intentar aprender un concepto difícil.• Pregúntale cómo se siente después de la sesión.• Revisa y comprueba su comprensión.		
<div><div></div><div><p>La maestra puede mostrar la presentación de diapositivas desde la diapositiva 16 a la 25.</p><p>Consulta las notas en tu pantalla y sigue las instrucciones de cada diapositiva.</p></div></div>		
<p>Por ahora, ¿podemos resumir rápidamente lo que hemos aprendido hoy?</p>	<p>REA:</p> <ul style="list-style-type: none">- Aprendimos a escribir funciones en nuestro código.- Aprendimos a dar movimiento a los sprites.	

	- Creamos bordes y agregamos la función <code>bounceOff()</code> para el sprite.
Sesión de PyR: haz clic en el quiz de la clase	
Pregunta	Respuesta
<p>¿Por qué escribimos funciones?</p> <p>A. Para que el código sea más legible. B. Para hacer que el código sea reutilizable. C. Para evitar escribir código repetido. D. Todo lo anterior.</p>	D
<p>Elige la sintaxis de código correcta para hacer que la pelota rebote en las paredes.</p> <p>A. <code>var ball = createSprite(200, 200,10,10);</code> <code>ball.velocityX = 2;</code> <code>ball.velocityY = 3;</code></p> <pre>function draw() { background("white"); ball.bounce(edges); drawSprites(); }</pre> <p>B. <code>var ball = createSprite (200, 200,10,10);</code> <code>ball.velocityX = 2;</code> <code>ball.velocityY = 3;</code> <code>createEdgeSprites();</code></p> <pre>function draw() { background("white"); drawSprites(); }</pre> <p>C. <code>var ball = createSprite(200, 200,10,10);</code> <code>ball.velocityX = 2;</code> <code>ball.velocityY = 3;</code> <code>createEdgeSprites();</code></p> <pre>function draw() { background("white"); ball.bounceOff(edges); drawSprites(); }</pre>	C

<pre> } D. var ball = createSprite(200, 200,10,10); ball.velocityX = 2; ball.velocityY = 3; createEdgeSprites(); function draw() { background("white"); ball.bounceOff(); drawSprites(); }. </pre>	
<p>¿Qué función se usa para evitar que el sprite salga de la pantalla al crear un límite alrededor del lienzo?</p> <p>A. drawSprites() B. createSprites() C. bounceOff() D. createEdgeSprites()</p>	<p>D</p>
<p>Felicidades por tu excelente trabajo.</p> <p>Muy bien, parece que hoy hemos aprendido mucho en la clase.</p> <p>En la siguiente clase aprenderemos a controlar el movimiento de la pelota.</p> <p>¿No es interesante?</p>	<p>Asegúrate de dar al menos 2 felicitaciones durante la clase por:</p> <div data-bbox="1149 1024 1416 1423">  </div>
<p>Descripción del proyecto</p> <p>PEGA Y CORRE</p> <p>Objetivo del proyecto: Hoy, aprendiste a definir funciones personalizadas. Además, aprendiste a crear sprites de borde y rebotar la pelota desde los bordes. Ahora crearás un pequeño juego de laberinto, para hacer rebotar las pelotas en movimiento desde los bordes y los sprites de recuadro.</p>	<p><i>El alumno interactúa con la maestra durante el proyecto.</i></p>

<p>En este proyecto, tendrás que practicar lo aprendido en clase y aplicarlo para hacer que la pelota se mueva y rebote dentro del laberinto, y que no salga del lienzo.</p> <p>Historia:</p> <p>A Dodo le encanta jugar con la pelota y hacer que rebote en las paredes y en diferentes objetos. Pero siempre existe el temor de romper cosas dentro de la casa. La madre de Dodo lo ayuda a crear una arena llena de objetos en el patio trasero, pero desde que Dodo ha comenzado su viaje en programación, ahora está ansioso por intentar crear una arena virtual donde pueda hacer rebotar su pelota y jugar sin temor a romper cosas.</p> <p>¿Puedes ayudar a Dodo a construir una arena así?</p> <p>Estoy muy emocionada de ver la solución de tu proyecto y sé que lo harás muy bien.</p> <p>¡Adiós!</p>	
<div> <div>La maestra finaliza la presentación de diapositivas</div>  </div>	
<div> <div>ACTIVIDAD ADICIONAL</div> <div>El alumno comienza a compartir su pantalla</div> </div>	
<p>Hoy aprendimos a crear una función personalizada. ¿Puedes crear una función para sumar dos números?</p> <p><i>Guía al alumno para que haga clic en Crear-> Laboratorio de Juegos para comenzar un nuevo proyecto desde cero.</i></p> <ul style="list-style-type: none"> - ¿Cuál debería ser el nombre de la función? - ¿Cuántas entradas deben pasarse a la función? <p><i>La maestra guía al niño en la escritura de la función sum() (suma) con dos parámetros de entrada num1 (número 1), y num2 (número 2). Imprime el resultado en la consola.</i></p> <p><i>La maestra ayuda al niño a llamar a la función con diferentes</i></p>	<p>REA: suma</p> <p>REA: 2</p>

entradas en el mismo código y ejecutar el código.



Estuviste fantástico hoy. Entendiste rápidamente todos los conceptos.

Actividad	Nombre de la actividad	Enlaces
Enlace de la actividad de la maestra 1	Función definida por el usuario: Rotar Sprite	https://studio.code.org/projects/gamelab/TG2LO7MXH4n6HGgh8K7C9akBI3nb-zYHvedPPAHgsFc
Enlace de la actividad de la maestra 1 (código de referencia)	Función definida por el usuario: Rotar Sprite	https://studio.code.org/projects/gamelab/l3vPzWdy3ep1LS5brCjoCWcGnxEKAYapvreFZNQVPY
Enlace de la actividad del alumno 1	Cambiar el tamaño del Sprite	https://studio.code.org/projects/gamelab/plafrqqGJ4Zkju7HTnxkfCZyFYTW-qf4fEdUvED_tIE
Código de referencia de la maestra (cambiar el tamaño)	Cambiar el tamaño del Sprite	https://studio.code.org/projects/gamelab/l3vPzWdy3ep1LS5brCjoP6DCVAfvkRQLaiDi8ux1gl

Enlace de la actividad de la maestra 2	Movimiento del fantasma	https://studio.code.org/projects/gamelab/l3vPzWd_y3ep1LS5brCjoMAWatVVVT8d5PAhg9d9-9s
Enlace de la actividad de la maestra 2 (código de referencia)	Movimiento del fantasma	https://studio.code.org/projects/gamelab/l3vPzWd_y3ep1LS5brCjoN50X5urJyRWGfr5sbttTVw
Enlace de la actividad del alumno 2	Movimiento de la pelota	https://studio.code.org/projects/gamelab/yjNaXvk2L1ugxoJW-jKoVJ6cFb6stNUU25AUNY8YEIk
Código de referencia de la maestra (movimiento de la pelota)	Movimiento de la pelota	https://studio.code.org/projects/gamelab/kAFPN8ukxX6zxKageR_R8x2Dn4AMZy1xatil1s5yIsU
Solución del proyecto	PEGA Y CORRE	https://studio.code.org/projects/gamelab/ip4IOCswuZu1dYfh2Oj1kWY2sZzPLPULh9WUPNScefQ
Enlace de ayuda visual de referencia de la maestra	Enlace de ayuda visual con notas	https://s3-whjr-curriculum-uploads.whjr.online/eab886ff-9127-42be-b79c-05a6b931a8a9.html
Enlace de ayuda visual de referencia de la maestra	Enlace de ayuda visual sin notas	https://s3-whjr-curriculum-uploads.whjr.online/fc8884fb-2c8d-4aa7-b3f8-3eff109fa9e4.html
Referencia del quiz de la clase de la maestra	Quiz de la clase	https://s3-whjr-curriculum-uploads.whjr.online/666c381f-4e02-42c9-bb90-2d8105b9ed2d.pdf
Actividad adicional (enlace de referencia)	Función sum	https://studio.code.org/projects/gamelab/kAFPN8ukxX6zxKageR_R89z4pv6b9mdcX3O50bWRZtE