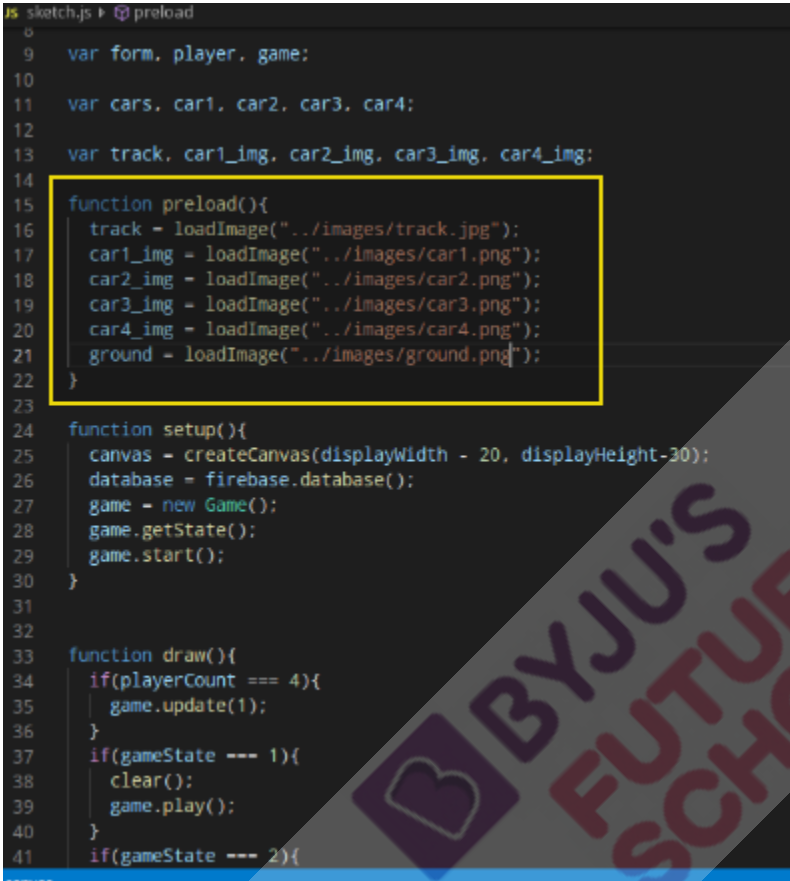



<b>Tema</b>	<b>Carreras de Autos Reales</b>	
<b>Descripción de la Clase</b>	Los alumnos reemplazarán los sprites con autos reales de su elección. Los alumnos agregarán una pista de carreras como fondo. Los alumnos también escribirán una condición para finalizar el juego.	
<b>Clase</b>	<b>C39</b>	
<b>Duración de la clase</b>	<b>45 minutos</b>	
<b>Objetivo</b>	<ul style="list-style-type: none"> <li>• Agregar una pista en segundo plano.</li> <li>• Reemplazar los sprites del auto, con imágenes de autos reales.</li> <li>• Escribir una condición para terminar el juego.</li> </ul>	
<b>Recursos Requeridos</b>	<ul style="list-style-type: none"> <li>• Recursos para Profesoras <ul style="list-style-type: none"> <li>○ Laptop con conectividad a internet</li> <li>○ Auriculares con micrófono</li> <li>○ Cuaderno y bolígrafo</li> </ul> </li> <li>• Recursos para Alumnos <ul style="list-style-type: none"> <li>○ Laptop con conectividad a internet</li> <li>○ Auriculares con micrófono</li> <li>○ Cuaderno y bolígrafo</li> </ul> </li> </ul>	
<b>Estructura de la Clase</b>	<b>Rompiendo el Hielo</b> <b>Actividad dirigida por la Profesora</b> <b>Actividad dirigida por el Alumno</b> <b>Conclusión</b>	<b>5 minutos</b> <b>10 minutos</b> <b>20 minutos</b> <b>5 minutos</b>
<p style="text-align: center;"><b><u>CONTEXTO</u></b></p> <ul style="list-style-type: none"> <li>• Creación de un juego añadiendo gráficos personalizados: elección de autos y pistas.</li> </ul>		
<b>Pasos de Clase</b>	<b>Acción de la Profesora</b>	<b>Acción del Alumno</b>
<b>Paso 1: Rompiendo el Hielo</b>	¡Hola! Bienvenido de nuevo. ¿Recuerdas dónde nos quedamos en	<b>REA:</b> Habíamos creado los sprites rectangulares de

(5 minutos)	la última clase?	autos, que se movían con el cambio en los valores de la base de datos de la distancia recorrida para cada auto. También, ajustamos el tamaño del lienzo y la posición de todos los elementos para cubrir el tamaño de la pantalla.
	Entonces, ya tenemos un juego de carreras de autos, ¿no?  ¿Qué te gustaría hacer ahora? ¿Qué haría que el juego fuera más emocionante?	<b>REA:</b> Si. <b>REA:</b> Añadir pistas, autos reales, sonidos, etc.
	Vamos a hacer exactamente eso en la clase de hoy.  Vamos a agregar la pista. También podrás elegir qué autos quieres para la carrera. Finalmente, escribirás una condición que indica cuándo terminará el juego.	-
<b>La Profesora Inicia Compartir Pantalla</b>		
<p style="text-align: center;"><b><u>DESAFÍO</u></b></p> <ul style="list-style-type: none"> <li>• Agregar la pista en segundo plano.</li> <li>• Reemplazar los sprites rectangulares con autos.</li> </ul>		
<b>Paso 2:</b> <b>Actividad dirigida por la Profesora</b> <b>(10 minutos)</b>	Antes de comenzar a agregar las imágenes y los gráficos y hacer que nuestro juego se vea genial, repasemos nuestro código hasta ahora.	<i>El alumno revisa el código. El alumno explora cada clase: las propiedades y funciones de la clase, los objetos creados con la</i>

	Guíe al alumno para que revise el código.	<i>clase.</i>
	<p>¿Puedes expresar con tus propias palabras?</p> <p>- ¿Qué está sucediendo en el juego?</p> <p>- ¿Qué está viendo el usuario?</p> <p>- ¿Cuál es el código que se ejecuta detrás de él?</p> <p><i>Ayude al alumno donde tenga dificultades.</i></p>	<i>El alumno explica los diferentes eventos que ocurren en el juego, y cómo el código escrito en el juego lo está influenciando.</i>
	<p>¡Impresionante!</p> <p>Recuerda, siempre nos hemos centrado en escribir el código limpio y legible. No solo ayuda a otros a comprender tu código, sino que también nos ayuda cuando escribimos funciones adicionales en nuestro código. Facilita el agregar código nuevo, en la parte superior de nuestro código escrito anteriormente.</p> <p>Lo verás en la clase de hoy, donde agregaremos todos los gráficos para el juego, y escribiremos la funcionalidad para finalizar el juego.</p> <p>Empecemos.</p>	<i>El alumno escucha.</i>
<b>La Profesora Detiene Compartir Pantalla</b>		
	Ahora es tu turno. Comparte tu pantalla conmigo.	
<ul style="list-style-type: none"> <li>• <b>Pídale al Alumno que presione la tecla ESC para volver al panel</b></li> <li>• <b>Guíe al Alumno para que comience a Compartir Pantalla</b></li> <li>• <b>La Profesora entra en Pantalla Completa</b></li> </ul>		

<b>ACTIVIDAD</b> • Escribe la condición para terminar el juego.		
<b>Paso 3:</b> <b>Actividad</b> <b>dirigida por el</b> <b>Alumno</b> <b>(20 minutos)</b>	<p>Guíe al alumno para que clone el código de la clase anterior, o descargue las imágenes necesarias para el proyecto; y que las añada a su código escrito previamente, y habilite el live share, copie el enlace y lo comparta con la profesora.</p> <p>&lt;La profesora abre el enlace e inicia la colaboración en vivo.&gt;</p>	<p>El alumno clona el proyecto de la clase anterior o agrega imágenes en su proyecto anterior.</p> <p>El alumno habilita el live share y comparte el enlace con la profesora.</p> <p>El alumno se asegura de que las claves API sean de su base de datos.</p>
	<p>Carguemos todas las imágenes necesarias en el código.</p> <p>¿Dónde debemos cargar la imagen en nuestro código?</p> <p>¿Qué función se usa para cargar la imagen?</p>	<p><b>REA:</b> En la función preload() [dentro de sketch.js].</p> <p><b>REA:</b> loadImage ()</p>
	<p>¡Genial!, carguemos todas las imágenes.</p> <p>Tenemos 5 imágenes en total:</p> <ul style="list-style-type: none"> <li>- el suelo (ground)</li> <li>- la pista (track)</li> <li>- 4 imágenes para los autos</li> </ul> <p><b>Nota:</b> Utiliza aquí la ruta absoluta para los archivos de imagen.</p> <p><b>Ejemplo de la ruta absoluta:</b> C:\Documentos\JuegoCorredorAutos\imágenes\car1.png</p> <p><b>Ejemplo de una ruta relativa:</b></p>	<p>El alumno carga todas las imágenes.</p>

	images\car1.png	
 <pre> 15 sketch.js ▶ preload 16 17 var form, player, game; 18 19 var cars, car1, car2, car3, car4; 20 21 var track, car1_img, car2_img, car3_img, car4_img; 22 23 function preload(){ 24   track = loadImage("../images/track.jpg"); 25   car1_img = loadImage("../images/car1.png"); 26   car2_img = loadImage("../images/car2.png"); 27   car3_img = loadImage("../images/car3.png"); 28   car4_img = loadImage("../images/car4.png"); 29   ground = loadImage("../images/ground.png"); 30 } 31 32 function setup(){ 33   canvas = createCanvas(displayWidth - 20, displayHeight-30); 34   database = firebase.database(); 35   game = new Game(); 36   game.getState(); 37   game.start(); 38 } 39 40 function draw(){ 41   if(playerCount === 4){ 42     game.update(1); 43   } 44   if(gameState === 1){ 45     clear(); 46     game.play(); 47   } 48   if(gameState === 2){ </pre>		
	<p>Tenemos imágenes de autos reales cargadas en nuestro código.</p> <p>Podemos agregar animación a nuestros aburridos sprites rectangulares, y hacer que se vean como los autos geniales que tenemos.</p> <p>¿Recuerdas cómo agregar imágenes a nuestros sprites?</p>	<p><b>REA:</b> Usando addImage ()</p>

	<p>Genial, agreguemos las imágenes de sprites en nuestro código.</p>	<p><i>El alumno añade todas las imágenes a los diferentes autos. El alumno da trayectorias absolutas para todas las imágenes.</i></p>
 <pre> 18 } 19 20 async start(){ 21   if(gameState === 0){ 22     player = new Player(); 23     var playerCountRef = await database.ref('playerCount').once("value"); 24     if(playerCountRef.exists()){ 25       playerCount = playerCountRef.val(); 26       player.getCount(); 27     } 28     form = new Form() 29     form.display(); 30   } 31 32   car1 = createSprite(100,200); 33   car1.addImage("car1",car1_img); 34   car2 = createSprite(300,200); 35   car2.addImage("car2",car2_img); 36   car3 = createSprite(500,200); 37   car3.addImage("car3",car3_img); 38   car4 = createSprite(700,200); 39   car4.addImage("car4",car4_img); 40   cars = [car1, car2, car3, car4]; 41 } 42 43 play(){ 44   form.hide(); 45 46   Player.getPlayerInfo(); 47 48   if(allPlayers !== undefined){ 49     background("#c68767"); 50     image(track, 0,-displayHeight*4,displayWidth, displayHeight*5); </pre>		
	<p>Estupendo. Carguemos un suelo, y la pista donde se desarrollará el juego de carreras de autos.</p> <p>Podemos usar una imagen de fondo o un color de fondo como fondo.</p> <p>Escribamos el código para eso.</p>	<p><i>El alumno escribe el código para agregar el fondo a la imagen.</i></p>

```

32 car1 = createSprite(100,200);
33 car1.addImage("car1",car1_img);
34 car2 = createSprite(300,200);
35 car2.addImage("car2",car2_img);
36 car3 = createSprite(500,200);
37 car3.addImage("car3",car3_img);
38 car4 = createSprite(700,200);
39 car4.addImage("car4",car4_img);
40 cars = [car1, car2, car3, car4];
41 }
42
43 play(){
44   form.hide();
45
46   Player.getPlayerInfo();
47
48   if(allPlayers !== undefined){
49     background("#c68767");
50   }
51   image(track, 0,-displayHeight*4,displayWidth, displayHeight*5);
52
53   //var display_position = 100;
54
55   //index of the array
56   var index = 0;
57
58   //x and y position of the cars
59   var x = 175 ;
60   var y;
61
62   for(var plr in allPlayers){
63     //add 1 to the index for every loop
64     index = index + 1 ;

```

Ahora podemos agregar imágenes de pistas en el juego. Ya sabes cómo cargar una imagen en el juego.

¿Cuáles son los diferentes argumentos que toma image()?

**REA:**  
Usar image()

*El alumno consulta la documentación de image() para comprender los diferentes argumentos que toma.*

La imagen de la pista es muy larga. Queremos que comience incluso antes de 0, 0.  
Empecemos en 0: displayHeight\*4  
Esto iniciará la imagen de la pista 4 veces por encima de nuestro origen de pantalla. ¡Podríamos dar un

*El alumno usa la función image, para dibujar la imagen de la pista sobre la función background.*



número más alto si quisiéramos una pista más larga!

También queremos que nuestra imagen ocupe un ancho y un alto de `displayWidth`, `displayHeight*5`. Queremos que la imagen cubra la pista empezando por el origen.

Queremos que la pantalla esté cubierta con la pista.

```

32 car1 = createSprite(100,200);
33 car1.addImage("car1",car1_img);
34 car2 = createSprite(300,200);
35 car2.addImage("car2",car2_img);
36 car3 = createSprite(500,200);
37 car3.addImage("car3",car3_img);
38 car4 = createSprite(700,200);
39 car4.addImage("car4",car4_img);
40 cars = [car1, car2, car3, car4];
41 }
42
43 play(){
44   form.hide();
45
46   Player.getPlayerInfo();
47
48   if(allPlayers !== undefined){
49     background("#c68767");
50
51     image(track, 0,-displayHeight*4,displayWidth, displayHeight*5);
52
53     //var display_position = 100;
54
55     //index of the array
56     var index = 0;
57
58     //x and y position of the cars
59     var x = 175 ;
60     var y;
61
62     for(var plr in allPlayers){
63       //add 1 to the index for every loop
64       index = index + 1 ;

```



	<p>Ejecutemos el código y veamos qué pasa.</p>	<p><i>El alumno abre la consola Firebase, y se asegura de que gameState y playCount estén en 0.</i></p> <p><i>El alumno ejecuta el código en 4 navegadores diferentes para ver el resultado.</i></p>
	<p>¿Qué ves?</p> <p>¿Hay algo incorrecto?</p>	<p><b>REA:</b></p> <p>Los autos no están correctamente alineados.</p>
	<p>Alineemos los autos. Solo necesitamos cambiar los valores de x.</p> <p><i>Permita que el alumno experimente con diferentes valores de x.</i></p>	<p><i>El alumno modifica el valor de x para que los autos estén alineados con la pista.</i></p>

```

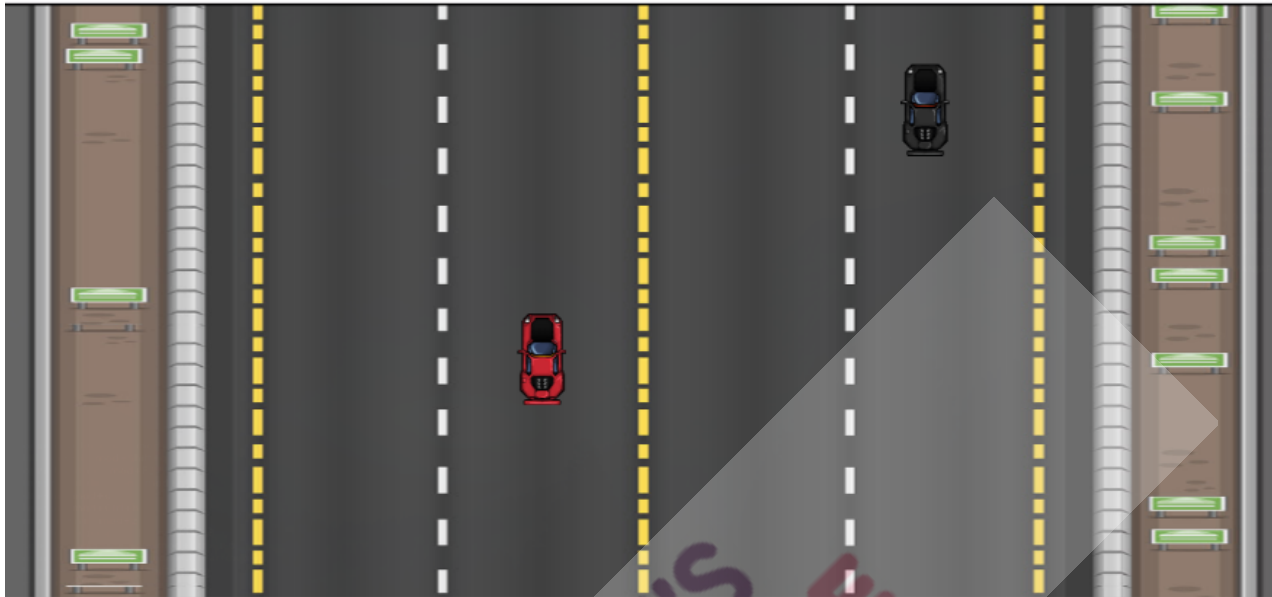
47
48   if(allPlayers !== undefined){
49     background("#c68767");
50
51     image(track, 0,-displayHeight*4,displayWidth, displayHeight*5);
52
53     //var display_position = 100;
54
55     //index of the array
56     var index = 0;
57
58     //x and y position of the cars
59     var x = 175 ;
60     var y;
61
62     for(var plr in allPlayers){
63       //add 1 to the index for every loop
64       index = index + 1 ;
65
66       //position the cars a little away from each other in x direction
67       x = x + 200;
68       //use data form the database to display the cars in y direction
69       y = displayHeight - allPlayers[plr].distance;
70       cars[index-1].x = x;
71       cars[index-1].y = y;
72
73       if (index === player.index){
74         cars[index - 1].shapeColor = "red";
75         camera.position.x = displayWidth/2;
76         camera.position.y = cars[index-1].y;
77       }
78
79       //textSize(15);

```

Ejecutemos el código nuevamente y veamos qué sucede.

*El alumno abre la consola Firebase, y se asegura de que gameState y playCount sea 0. El alumno ejecuta el código en 4 navegadores diferentes para ver el resultado.*

*El alumno usa las teclas de flecha para mover los autos en diferentes navegadores.*



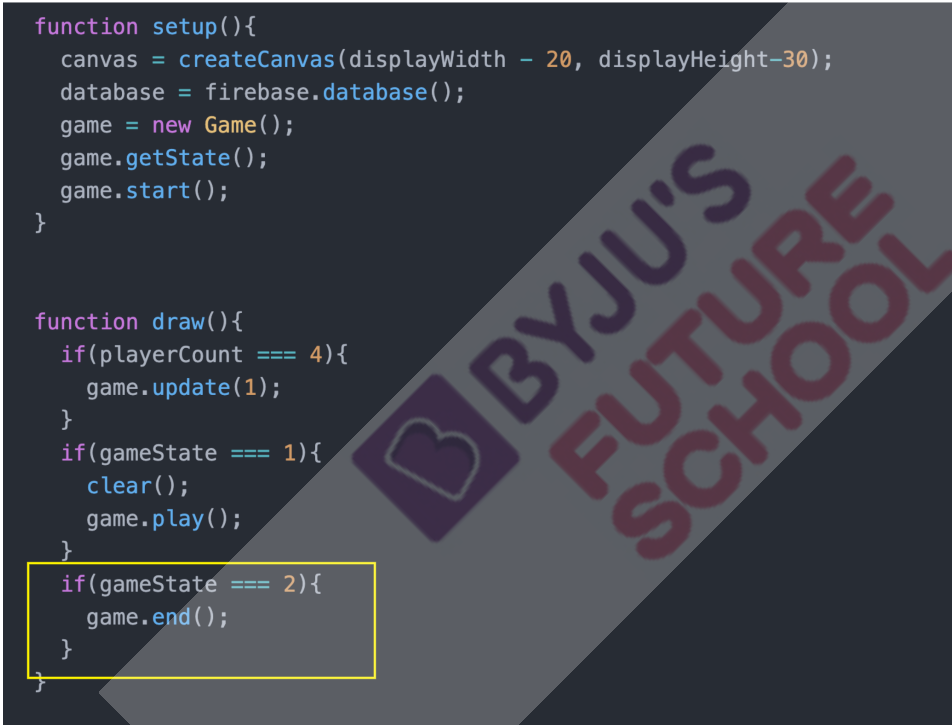
Bien, escribamos una condición para finalizar el juego, cuando el jugador llegue a la señal de fin en la carretera.  
Cambiaremos el estado del juego si la distancia recorrida por un jugador es de 3860.

*El alumno escribe el código para finalizar el juego cuando el jugador llega al estado final.*

```

74 cars[index - 1].shapeColor = "red";
75 camera.position.x = displayWidth/2;
76 camera.position.y = cars[index-1].y;
77 }
78
79 //textSize(15);
80 //text(allPlayers[plr].name + ": " + allPlayers[plr].distance, 120,display_position)
81 }
82
83 }
84
85 if(keyIsDown(UP_ARROW) && player.index !== null){
86   player.distance +=10
87   player.update();
88 }
89
90 if(player.distance > 3860){
91   gameState = 2;
92 }
93
94 drawSprites();
95 }

```

	<p>Escribamos una función end() dentro de Game Class, y llamemos a esto usando un objeto de juego en nuestro archivo sketch.js.</p> <p>Actualizaremos gameState en nuestra base de datos aquí. Además, imprimiremos en la consola que el juego ha finalizado.</p>	<p><i>El alumno escribe la función end() y actualiza gameState en la base de datos.</i></p>
 <pre> function setup(){   canvas = createCanvas(displayWidth - 20, displayHeight-30);   database = firebase.database();   game = new Game();   game.getState();   game.start(); }  function draw(){   if(playerCount === 4){     game.update(1);   }   if(gameState === 1){     clear();     game.play();   }   if(gameState === 2){     game.end();   } } </pre>		

```

3      cars[index - 1].shapeColor = "red";
4      camera.position.x = displayWidth/2;
5      camera.position.y = cars[index-1].y;
6  }
7
8      //textSize(15);
9      //text(allPlayers[plr].name + ": " + allPlayers[plr].distance, 120,display_posit
10     }
11
12     }
13
14     if(keyIsDown(UP_ARROW) && player.index !== null){
15         player.distance +=10
16         player.update();
17     }
18
19     if(player.distance > 3860){
20         gameState = 2;
21     }
22
23     drawSprites();
24 }
25
26 end(){
27     console.log("Game Ended");
28 }




```

### La Profesora Guía al Alumno para Dejar de Compartir Pantalla

#### RETROALIMENTACIÓN

- Anime al alumno a tomar notas de reflexión en markdown.
- Felicite al alumno por sus esfuerzos durante la clase.
- Repasen el contenido aprendido.

<b>Paso 4:</b> <b>Conclusión</b> <b>(5 minutos)</b>	¿Viste lo fácil que fue agregar los gráficos, y las nuevas funcionalidades en nuestro Juego de Carreras de Autos?	<b>REA:</b> Si.
	Fue fácil escribir las nuevas funcionalidades con tanta fluidez porque hemos dedicado mucho tiempo a la estructura de nuestro código, definiendo las clases, nombrando cada función, propiedad y variables con claridad.	<i>El alumno escucha.</i>

	¿Qué crees que aún falta en el juego?	<b>REA:</b> - 'Juego Terminado' en la pantalla - Tabla de Clasificación - Cómo reconocer qué auto pertenece al jugador - Sonidos
	¡Felicidades por tu excelente trabajo!  ¡Agregaremos estas características en la próxima clase, y haremos que este juego sea aún más asombroso!	Asegúrese de haber regalado al menos 2 sombreros durante la clase por:   
<b><u>Nombre del Proyecto: Juego del Corredor Infinito-2</u></b>	<b>Objetivo del Proyecto:</b>  En la clase 39, aprendiste cómo agregar imágenes para el fondo y para los sprites. También escribiste una condición para finalizar el juego.  En este proyecto, tendrás que agregar las imágenes y las condiciones de finalización del juego en tu juego anterior, que creaste en la Clase 38.	<i>Los alumnos se involucran con la profesora en el proyecto.</i>

	<p><b>** Esta es una continuación del Proyecto 38, así que asegúrate de completarlo antes de realizar este proyecto. **</b></p> <p><b>Historia:</b></p> <p>Ya has creado tu juego, y has agregado el componente de la cámara a lo largo del eje X. Ahora, asegurémonos de que el juego tenga mejores imágenes y, finalmente, una funcionalidad de final de juego en este proyecto.</p> <p>¿Estás listo para el desafío?</p> <p>Estoy muy emocionada de ver la solución de tu proyecto y sé que lo harás realmente bien.</p> <p>¡Adiós!</p>	
<p><b>La Profesora hace Clic en</b> <a href="#">✕ Finalizar Clase</a></p>		
<b>Actividades Adicionales</b>	<p><i>Anime al alumno a escribir notas de reflexión en su diario de reflexión utilizando markdown.</i></p> <p>Úsalos como preguntas de orientación:</p> <ul style="list-style-type: none"> <li>• ¿Qué pasó hoy? <ul style="list-style-type: none"> <li>- Describe lo que pasó</li> <li>- Código que escribí</li> </ul> </li> <li>• ¿Cómo me sentí después de la clase?</li> </ul>	<p><i>El alumno usa el editor markdown para escribir su reflexión como un diario de reflexión.</i></p>



	<ul style="list-style-type: none"> <li>• ¿Qué he aprendido sobre programación y desarrollo de juegos?</li> <li>• ¿Qué aspectos de la clase me ayudaron? ¿Qué encontré difícil?</li> </ul>	
--	---	--

Actividad	Nombre de la Actividad	Enlaces
Actividad de la Profesora 1	Código de Referencia	<a href="https://github.com/whitehatjr/carRacingStage2">https://github.com/whitehatjr/carRacingStage2</a>
Actividad del Alumno 1	Actividad previa de la clase	<a href="https://github.com/whitehatjr/CarRacingStage1.5/">https://github.com/whitehatjr/CarRacingStage1.5/</a>
Solución del Proyecto	JUEGO DEL CORREDOR INFINITO-2	Al tratarse de un proyecto abierto, no existe una solución específica