

Tema	Clasificación del Jugador	
Descripción de la clase	Los alumnos construyen una tabla de clasificación, que clasifica al jugador de acuerdo con su desempeño en el juego carreras de auto0073.	
Clase	C41	
Duración de la clase	45 minutos	
Objetivo	<ul style="list-style-type: none"> Almacenar la clasificación de los jugadores individuales en el juego como propiedad del jugador. Mostrar la clasificación del jugador cuando termina el juego. 	
Recursos requeridos	<ul style="list-style-type: none"> Recursos para Profesoras <ul style="list-style-type: none"> Laptop con conectividad a internet Auriculares con micrófono Cuaderno y bolígrafo Recursos para Alumnos <ul style="list-style-type: none"> Laptop con conectividad a internet Auriculares con micrófono Cuaderno y bolígrafo 	
Estructura de clase	Rompiendo el hielo. Actividad dirigida por la profesora. Actividad dirigida por el alumno. Conclusión	5 minutos 5 minutos 25 minutos 5 minutos
<p style="text-align: center;"><u>CONTEXTO</u></p> <ul style="list-style-type: none"> Papel de dar una clasificación en un juego de carreras 		
Pasos de la clase	Acción de la profesora	Acción del alumno
Paso 1: Rompiendo el hielo. (5 minutos)	Bienvenido a la clase Capstone. Hemos visto cómo agregar nuevas funcionalidades al juego, ha sido mucho más fácil debido a la estructura del código.	

	<p>¿Estás preparado para el desafío de hoy?</p> <p>Terminaremos nuestro juego hoy agregando una funcionalidad más importante.</p> <p>¿Puedes adivinar cuál?</p>	<p>REA: variado</p>
	<p>Construiremos la tabla de clasificación, que clasifica a los jugadores en función de cuándo terminaron el juego. Dirá quién llegó primero, segundo, tercero o cuarto.</p>	<p><i>El alumno escucha.</i></p>
	<p>Y, por supuesto, también puedes agregar los sonidos en el juego, para hacerlo aún más genial. ¿Estás listo para comenzar?</p>	<p>REA: ¡Sí!</p>
<p>La profesora comienza a compartir la pantalla</p>		
<p style="text-align: center;"><u>DESAFÍO</u></p> <ul style="list-style-type: none"> • Agregar una propiedad llamada "rank" – <i>clasificación</i>, para cada jugador. • Actualizar la clasificación a medida que cambia la distancia recorrida por cada jugador. 		
<p>Paso 2: Actividad dirigida por la profesora. (5 minutos)</p>	<p>¿Alguna idea sobre cómo empezar?</p>	<p>REA: variado</p>
	<p>Podemos comenzar agregando una propiedad más a cada jugador llamada – 'rank'.</p> <p>También podemos agregar algo en nuestra base de datos y código, que lleve el recuento de la cantidad de jugadores que han terminado el juego. ¿Puedes pensar en, cómo</p>	

	<p>podemos usar la propiedad de clasificación y la cantidad de jugadores que han terminado la carrera, para mostrar una tabla de clasificación?</p> <p>Nota: Podría haber muchas otras formas de mostrar una tabla de clasificación, y la profesora podría permitir que el alumno explore otras formas.</p>	<p>REA: variado</p>
	<p>Ok ... intentemos implementar la tabla de clasificación en nuestro juego, y mostrar la clasificación de cada jugador cuando finalice el juego.</p> <p>¿Te sientes seguro de que puedes hacerlo por tu cuenta?</p>	<p>REA: variado</p>
<p>La profesora deja de compartir la pantalla</p>		
	<p>Ahora es tu turno. Comparte tu pantalla conmigo.</p>	
<ul style="list-style-type: none"> • Pídele al alumno que presione la tecla ESC para volver al panel. • Guía al alumno, para que comience a compartir la pantalla. • Ingresa al modo de pantalla completa. 		
<p><u>ACTIVIDAD</u></p> <ul style="list-style-type: none"> • Muestra la clasificación cuando termina el juego para un jugador. 		

Paso 3: Actividad dirigida por el alumno. (25 minutos)	<p><i>Permite que el alumno piense en voz alta e intente escribir su propio código, para mostrar la clasificación de cada jugador y ejecutarlo.</i></p> <p><i>Está bien si el alumno falla o recibe errores en su código. Ayuda al alumno a pensar en voz alta, para tener claro lo que está haciendo.</i></p>	<p><i>El alumno intenta escribir su propio código, para mostrar la clasificación de los jugadores durante los primeros 10-15 minutos.</i></p>
	<p>Déjame ver si puedo ayudarte. Clonemos el código de la clase anterior.</p>	<p><i>El alumno clona la Actividad del alumno 1 o abre el código de la clase anterior.</i></p> <p><i>El alumno enciende el servidor web para iniciar el proyecto.</i></p>
	<p><i>Guía al alumno, para que cree una propiedad adicional para el objeto Player – jugador.</i></p>	<p><i>El alumno crea una propiedad adicional, para cada jugador llamada "rank" y la agrega al constructor de cada jugador.</i></p>

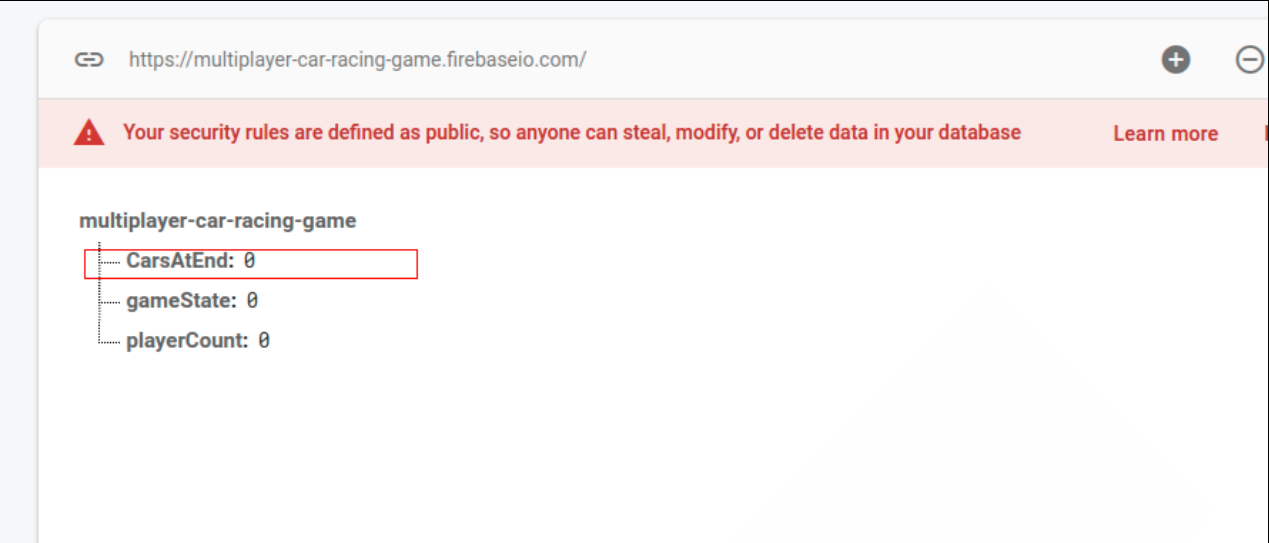
```

js > JS Player.js > Player > update
1  class Player {
2    constructor(){
3      this.index = null;
4      this.distance = 0;
5      this.name = null;
6      this.rank = null;
7    }
8
9    getCount(){
10     var playerCountRef = database.ref('playerCount');
11     playerCountRef.on("value",(data)=>{
12       playerCount = data.val();
13     })
14   }
15
16   updateCount(count){
17     database.ref('/').update({
18       playerCount: count
19     });
20   }
21
22   update(){
23     var playerIndex = "players/player" + this.index;
24     database.ref(playerIndex).set({
25       name:this.name,
26       distance:this.distance
27     });
28   }
29
30   static getPlayerInfo(){
31     var playerInfoRef = database.ref('players');
32     playerInfoRef.on("value",(data)=>{

```

Guía al alumno a cambiar la base de datos, para crear una nueva entrada para "CarsAtEnd" – autos al final.

El alumno, crea una nueva entrada en la base de datos llamada "CarsAtEnd" que almacena el número de coches que han llegado al punto final.

	<p><i>Ayuda al alumno a escribir las funciones: <code>getCarsAtEnd()</code> – obtén los autos al final, y una función estática <code>updateCarsAtEnd()</code> – actualiza los autos al final, para obtener y actualizar la cantidad de autos que han finalizado la carrera.</i></p> <p><i>¿Recuerdas qué son las "funciones estáticas"?</i></p> <p>REA: Las funciones estáticas, son funciones que son comunes a todos los objetos creados usando la clase.</p> <p><i>El alumno escribe una función estática dentro de la clase <code>Player</code> para <code>updateCarsAtEnd</code>.</i></p> <p><i>El alumno también escribe una función para <code>getCarsAtEnd</code>.</i></p> <p><i>El alumno asigna la clasificación actual del jugador dentro de <code>getCarsAtEnd</code>.</i></p>	
--	---	--

```

js sketch.js  JS Player.js  JS Game.js
js > JS Player.js > Player > update
25     name:this.name,
26     distance:this.distance
27   });
28   }
29
30   static getPlayerInfo(){
31     var playerInfoRef = database.ref('players');
32     playerInfoRef.on("value",(data)=>{
33       allPlayers = data.val();
34     })
35   }
36
37   getCarsAtEnd() {
38     database.ref('CarsAtEnd').on("value",(data)=>{
39       this.rank = data.val();
40     })
41   }
42
43   static updateCarsAtEnd(rank) {
44     database.ref('/').update({
45       CarsAtEnd:rank
46     })
47   }
48 }
49

```

Haz que el alumno use las funciones anteriores en su código, para actualizar la clasificación del jugador.

Cuando el jugador cruce el punto final, aumenta la clasificación del jugador en 1 y actualiza la entrada de la base de datos para carsAtEnd.

El alumno llama a la función getCarsAtEnd dentro del estado play - jugar.


```
js > JS Game.js > Game > play
39   car4.addImage("car4",car4_img);
40   cars = [car1, car2, car3, car4];
41 }
42
43 play(){
44   form.hide();
45
46   Player.getPlayerInfo();
47   Player.getCarsAtEnd();
48
49   if(allPlayers !== undefined){
50     background(rgb(198,135,103));
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 from the database to display the cars in y direction
```

Nota: Utiliza "player.getCarsAtEnd ()" en lugar de Player.getCarsAtEnd() ya que getCarsAtEnd() no es una función estática.


```

81         CameraId.position.y = cars[index-1].y;
82     }
83
84     //textSize(15);
85     //text(allPlayers[plr].name + ": " + allPlayers[plr].distance, 120,display_position)
86 }
87
88 }
89
90 if(keyIsDown(UP_ARROW) && player.index !== null){
91     player.distance +=10
92     player.update();
93 }
94
95 if(player.distance > 3860){
96     gameState = 2;
97     player.rank +=1
98     Player.updateCarsAtEnd(player.rank)
99 }
100
101 drawSprites();
102 }
103
104 end(){
105     console.log("Game Ended");
106     console.log(player.rank);
107 }
108 }
109

```

Haz que el alumno muestre la clasificación de cada jugador.

El alumno muestra la clasificación del jugador cuando termina el juego.

```

s > JS Game.js > Game > play
81      Carried.position.y = cars[index+1].y;
82    }
83
84    //textSize(15);
85    //text(allPlayers[plr].name + ": " + allPlayers[plr].distance, 120, disp
86  }
87
88  }
89
90  if(keyIsDown(UP_ARROW) && player.index !== null){
91    player.distance +=10
92    player.update();
93  }
94
95  if(player.distance > 3860){
96    gameState = 2;
97    player.rank +=1
98    Player.updateCarsAtEnd(player.rank)
99  }
100
101  drawSprites();
102  }
103
104  end(){
105    console.log("Game Ended");
106    console.log(player.rank);
107  }
108  }
109

```

Guía al alumno a ejecutar, probar y depurar el código.

El alumno ejecuta el código y prueba el código.

El alumno depura en caso de problemas.

La profesora guía al alumno para dejar de compartir la pantalla

RETROALIMENTACIÓN

- Anima al alumno a tomar notas de reflexión en formato markdown.
- Complementa al alumno por su esfuerzo en la clase.
- Repase el contenido de la lección.




Paso 4:
Conclusión
(5 minutos)

Ahora que tenemos la clasificación de cada jugador, podemos usarla de la forma que queramos cuando termine el juego. ¿Puedes pensar en formas en las que podrías usar la clasificación de los jugadores?

REA:

- Para mostrar la tabla de puntuación.

		– Para mostrar el rango del jugador
	<p>Impresionante. También puedes hacer que el juego sea más llamativo, agregando sonidos y otras características.</p> <p>¿Qué más características se te ocurren?</p>	<p>REA: variado</p>
	<p>Actualmente, nuestro coche solo se mueve en línea recta, puedes hacer que se mueva de lado. Puedes hacer que choque con otros coches y rebote. Hemos utilizado p5.play, para crear sprites en el juego. Podrías haber usado matter.js y crear personajes de autos que podrían interactuar con otros autos (u otros obstáculos) en el juego usando las leyes naturales de la física. Como se mencionó anteriormente, cuando estás creando tu propio juego, eres el DIOS del mundo que estás creando y puedes hacer lo que quieras. ¿Cuáles son algunas de las formas en que te gustaría modificar este juego para mejorarlo?</p>	<p>REA: variado</p>
	Felicidades por tu excelente trabajo.	<p><i>Asegúrate de dar al menos 2 felicitaciones durante la clase por:</i></p>

	<p>¡Increíble! Con esto daremos por terminada esta clase.</p> <p>Las clases 42 a 48, serán muy especiales en las que aprenderás más en profundidad sobre el diseño de juegos y cómo los diseñadores de juegos diseñan el mundo del juego. ¡Felicidades! Has establecido un nuevo punto de referencia. ¡Prepárate! Tu nuevo desafío está listo.</p> <p>Debes aplicar las construcciones de programación aprendidas durante las últimas clases, para crear la aplicación Recolector de Fruta.</p>	<div>Creatively Solved Activities  +10</div> <div>Great Question  +10</div> <div>Strong Concentration  +10</div>
<p><u>Nombre del proyecto:</u> <u>Recolector de fruta – 2.</u></p>	<p>Objetivo del proyecto:</p> <p>En la clase 41, continaste aprendiendo los conceptos sobre cómo calcular clasificaciones.</p> <p>En este proyecto, debes practicar lo que has aprendido en la clase y depurar el código, para arreglar el juego y mostrar la puntuación cuando los jugadores recolectan frutas.</p> <p>** Esta es una continuación del Proyecto 40, así que asegúrate de completarlo antes de realizar este proyecto. **</p> <p>Historia:</p> <p>Honey visitó la casa de sus abuelos, donde había una granja. Los agricultores estaban cortando frutas para cosecharlas. Su abuelo le dio la responsabilidad de recolectar todas</p>	<p><i>El alumno usa el editor markdown, para escribir su reflexión como un diario de reflexión.</i></p>

	<p>las frutas. Honey fue con su prima a la granja a recoger las frutas.</p> <p>¡Veamos quién recolecta más, Honey o su prima! En este proyecto, continuarás depurando el juego, por lo que se muestra la puntuación correcta para cada jugador.</p> <p>Estoy muy emocionada de ver la solución de tu proyecto y sé que a ambos les irá muy bien.</p> <p>¡Adiós!</p>	
<div> <div>✕ End Class</div> </div> <p>La profesora hace clic en</p>		
Actividades adicionales	<p><i>Anima 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"> • ¿Qué pasó hoy? <ul style="list-style-type: none"> - Describe lo que pasó - El código que escribí • ¿Cómo me sentí después de la clase? • ¿Qué he aprendido sobre programación y desarrollo de juegos? • ¿Qué aspectos de la clase me ayudaron? ¿Qué encontré difícil? 	<p><i>El alumno usa el editor markdown, para escribir su reflexión como un diario de reflexión.</i></p>

Actividad	Nombre de la actividad	Enlaces
Actividad de la profesora 1	Código de referencia de la profesora	https://github.com/whitehatjr/carRacingSta-ge3
Actividad del alumno 1	Código anterior del alumno	https://github.com/whitehatjr/carRacingSta-ge2.5
Solución del proyecto	Fruit Catcher-2 (Recogedor de frutas-2)	https://github.com/rupinwhitehatjr/5c24758b83bbadfe68a1e4fae7e90aca

