**PRO-C41**

Tema: Clasificación del Jugador

**Descripción:** Los alumnos construyen una tabla de clasificación, que clasifica al

jugador de acuerdo con su desempeño en el juego carreras de auto0073.

**Objetivos:** ● Almacenar la clasificación de los jugadores individuales en el

juego como propiedad del jugador. ● Mostrar la clasificación del jugador cuando termina el juego.

Bienvenido a la clase Capstone. Hemos visto cómo agregar nuevas funcionalidades al juego, ha sido mucho más fácil debido a la estructura del código.

Terminaremos nuestro juego hoy agregando una funcionalidad más importante. ¿Puedes adivinar cuál?

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.

Y, por supuesto, también puedes agregar los sonidos en el juego, para hacerlo aún más genial. ¿Estás listo

para comenzar?

¿Alguna idea sobre cómo empezar?. Podemos comenzar agregando una propiedad más a cada jugador

llamada – ‘rank’. 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 podemos usar la propiedad de clasificación y la cantidad de jugadores que han terminado la carrera, para mostrar una tabla de clasificación?

Nota: Podría haber muchas otras formas de mostrar una tabla de clasificación, y la profesora podría permitir que el alumno explore otras formas.

Ok ... intentemos implementar la tabla de clasificación en nuestro juego, y mostrar la clasificación de cada jugador cuando finalice el juego.

¿Te sientes seguro de que puedes hacerlo por tu cuenta?

Permite que el alumno piense en voz alta e intente escribir su propio código, para mostrar la clasificación de cada jugador y ejecutarlo.

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.

El alumno intenta escribir su propio código, para mostrar la clasificación de los jugadores durante los primeros 10-15 minutos.

Déjame ver si puedo ayudarte. Clonemos el código de la clase anterior.

El alumno clona la Actividad del alumno 1 o abre el código de la clase anterior. El alumno enciende el servidor web para iniciar el proyecto.

Guía al alumno, para que cree una propiedad adicional para el objeto Player – jugador.

El alumno crea una propiedad adicional, para cada jugador llamada "rank" y la agrega al constructor de

cada jugador. En Player.js

*this.rank = null;*

Guía al alumno a cambiar la base de Datos en firebase, 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.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Ayuda al alumno a escribir las funciones: getCarsAtEnd() – obtén los autos al final, y una función estática updateCarsAtEnd() – actualiza los autos al final, para obtener y actualizar la cantidad de autos que han finalizado la carrera.

¿Recuerdas qué son las "funciones estáticas"?. REA: Las funciones estáticas, son funciones que son comunes

a todos los objetos creados usando la clase.

El alumno escribe una función estática dentro de la clase Player para updateCarsAtEnd. En Player.js

*static updateCarsAtEnd(rank) {*

*database.ref('/').update({*

*CarsAtEnd:rank }) }*

El alumno también escribe una función para getCarsAtEnd. El alumno asigna la clasificación actual del jugador dentro de getCarsAtEnd.

*getCarsAtEnd() {*

*database.ref('CarsAtEnd').on("value",(data)=>{*

*this.rank = data.val(); })}*

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. Dentro de play() dentro de Game.js

*player.getCarsAtEnd();*

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

*if(player.distance > 3860){*

*gameState = 2;*

*player.rank +=1*

*Player.updateCarsAtEnd(player.rank) }*

Haz que el alumno muestre la clasificación de cada jugador. El alumno muestra la clasificación del jugador cuando termina el juego.

*end(){*

*console.log("Game Ended");*

*console.log(player.rank); }*

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

**Conclusión:** 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.

También puedes hacer que el juego sea más llamativo, agregando sonidos y otras características. ¿Qué más características se te ocurren?

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. Cómo 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?

**Desafío:** Nombre del Proyecto: Recolector de fruta – 2.

**Objetivo del proyecto:** En la clase 41, continuaste aprendiendo los conceptos sobre cómo calcular clasificaciones. 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.

\*\* Esta es una continuación del Proyecto 40, así que asegúrate de completarlo antes de realizar este proyecto.

**Historia:** 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 las frutas. Honey fue con su prima a la granja a recoger las frutas.

¡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. Estoy muy emocionada de ver la solución de tu proyecto y sé que a ambos les irá muy bien.

El alumno usa el editor **markdown**, para escribir su reflexión como un diario de reflexión.

● ¿Qué pasó hoy? ● 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?