**PRO-C35**

Tema: Base de Datos en Tiempo Real

**Descripción:** Los alumnos aprenden la importancia de usar una base de datos en tiempo real, para crear juegos en modo multijugador. Los alumnos aprenden a conectarse, leer y escribir datos en una base de datos remota en tiempo real.

**Objetivos:** ● Crear una base de datos remota en tiempo real. ● Leer y escribir datos en una base de datos en tiempo real. ● Crear una pelota que se mueve sincrónicamente, en diferentes navegadores

¡Hola! ¡Bienvenidos de nuevo a la clase! ¿Cómo te sientes después de la última clase?.

Hasta ahora, hemos creado muchos juegos, tanto proyectos en clase como después de clase. Tenemos juegos tan simples y antiguos como Pong. Incluso, creamos juegos que usan motores físicos como el de Angry Birds.

Pero todos estos juegos que hemos creado son juegos para un jugador. ¿No sería genial si pudiéramos crear juegos multijugador, que tú y tus amigos pudieran jugar en diferentes computadoras, mientras están conectados?.

¡¡Vamos a empezar a hacer exactamente eso!!. ¡Vamos a comenzar a construir un juego de carreras de autos multijugador en las próximas clases!

Ya sabemos cómo diseñar un juego con múltiples personajes. También, podemos alojar el juego en línea y luego abrirlo en diferentes navegadores en diferentes computadoras.

**¿Qué nos impide entonces diseñar un juego multijugador?.** **REA:** El juego se encuentra en diferentes estados en los dos navegadores. Para un juego multijugador, necesitamos que los dos navegadores tengan el juego en el mismo estado al mismo tiempo. Todo en los dos navegadores debe ser ***sincrónico***. Actualmente, los juegos son asincrónicos e independientes entre sí.

¡Exactamente! Veamos este problema con un ejemplo. El profesor escribe el código (igual que el de la ***Actividad del Profesor 1***) para crear una pelota simple, que se pueda mover con las teclas de flecha.

*var ball;*

*function setup(){*

*createCanvas(500,500);*

*ball = createSprite(250,250,20,20);*

*ball.shapeColor = rgb(255,255,0); }*

*function draw(){*

*background(255, 87, 34);*

*if(keyDown(LEFT\_ARROW)){ changePosition(-1,0); }*

*else if(keyDown(RIGHT\_ARROW)){ changePosition(1,0); }*

*else if(keyDown(UP\_ARROW)){ changePosition(0,-1); }*

*else if(keyDown(DOWN\_ARROW)){ changePosition(0,1); }*

*drawSprites(); }*

*function changePosition(x,y){*

*ball.x = ball.x + x;*

*ball.y = ball.y + y; }*

También podemos agregar un margen a nuestro lienzo para colocarlo en el centro.

*canvas{*

*margin-left: 5%;*

*margin-right: 25%;*

*margin-top: 150px;*

*border: 7px solid lightskyblue; }*

Abramos la aplicación en dos navegadores diferentes. Movamos la pelota y veamos qué pasa. ¿Que ves?. **REA:** La pelota en los dos navegadores se mueve de forma independiente. Sus movimientos son asincrónicos.

Esto sucede porque la posición de la pelota, en cada navegador, es independiente de la posición del otro.

Pero, ¿qué pasaría si pudiéramos almacenar la posición de la pelota, en una base de datos común remota, y nuestra aplicación leyera la posición de la pelota de la base de datos, y la actualizara cuando cambiara?

Los servidores de base de datos son computadoras que están conectadas de forma remota a través de Internet, y mantienen sus datos que puede usar en sus aplicaciones.

Los dos navegadores leerán la posición de la pelota desde la base de datos remota común y siempre estarán sincronizados.

Así es como funcionan los juegos multijugador. Almacenan la posición del estado del juego en todo momento en una base de datos remota.

Todas las consolas/navegadores de los jugadores, leen el juego desde esta base de datos remota, y escriben en él cuando hacen algún cambio en el juego.

Hagamos una base de datos remota para nuestra sencilla aplicación en Internet en la nube. Esta base de datos remota almacenará el estado (posiciones) de la pelota, y nos permitirá leerla o escribir en ella en cualquier momento.

Usaremos la Base de Datos en Tiempo Real de Google, Firebase, para este propósito. El profesor guía al alumno sobre cómo crear una base de datos en tiempo real, y crea una variable llamada ball (pelota), que

almacena dos valores, x e y.

La base de datos se puede comparar con un formato de estructura de datos JSON. ball = { x: 250, y: 250 }

Ingresar a la página de **Firebase** (de Google) y crear un proyecto: **Juego de Carreras Multijugador.**

1. Deshabilitar Googl Analitics, de momento no se va a requerir
2. Visita la sección Realtime Database (Base de datos en Tiempo real) del tablero, y haz clic en 'Crear una base de datos’.
3. Crea la base de datos en modo de prueba por ahora. (El modo de prueba nos ayudará a comenzar rápidamente. Será menos seguro en este modo, aunque más adelante, aprenderemos cómo hacer que la base de datos sea más segura)
4. Haz clic en Add child, para crear nodos que puedan contener las posiciones x e y de la pelota.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Conectemos la base de datos de Firebase a nuestra aplicación. Tendremos que **importar algunas bibliotecas de Firebase,** que nos permitan **leer y escribir** en nuestra base de datos.

También, necesitaremos proporcionar algunos **ajustes de configuración** como: clave api, databaseURL, etc. que pueden **autenticar nuestra aplicación** y permitirle **escribir** en la base de datos.

**No es prudente hacer esto público**, porque entonces cualquiera podrá escribir en tu aplicación. Pero nuestra aplicación no contiene ninguna información crítica, por lo que podemos agregar esto directamente en nuestro archivo html.

El profesor muestra cómo **conectar** la aplicación a la base de datos Firebase, y cómo **inicializar la App** de Firebase con la configuración propia de Firebase.

**Paso 1:** Haz clic en "Descripción general del proyecto". Elige Agregar a la Web para comenzar.

Registra app: Juego de Carrera de Autos

Agrega el SDK de firebase

**Paso 2:** Obtén la clave de configuración de Firebase. Descripción general de/ Configuración del proyecto/ Juego de Carreras de Autos/ Firebase SDK snipped/ **CDN**

**Paso 3:** Agrega esto al archivo index.html junto con la **biblioteca src para la base de datos** de Firebase.

Dentro de **index.html**

*<script src="https://www.gstatic.com/firebasejs/6.3.4/firebase-app.js"></script>*

*<script src="https://www.gstatic.com/firebasejs/6.3.0/firebase-database.js"></script>*

*<script>*

*const firebaseConfig = {*

*apiKey: "AIzaSyDp6is5bv4OSTnExS8KNbrHhVL1umR3Ans",*

*authDomain: "juego-de-carreras-multij-16e34.firebaseapp.com",*

*databaseURL: "https://juego-de-carreras-multij-16e34-default-rtdb.firebaseio.com",*

*projectId: "juego-de-carreras-multij-16e34",*

*storageBucket: "juego-de-carreras-multij-16e34.appspot.com",*

*messagingSenderId: "20632588317",*

*appId: "1:20632588317:web:97c671c972d7bc015eb6f4"*

*};*

*firebase.initializeApp(firebaseConfig);*

*</script>*

El profesor muestra cómo obtener una referencia a la posición de la pelota en la base de datos. **.ref()** se usa para referirse a la ubicación del valor de la base de datos que nos interesa. **.on()** crea un oyente que sigue escuchando los cambios en la base de datos.

Cada vez que ocurre un cambio en los valores de la base de datos de la posición (referencia), se llama a la función readPosition.

Si hay algún error al leer los valores en la base de datos, se llama a la función showError.

Dentro de **sketch.js:**

*var ball, database;*

*var position;*

*function setup(){*

*database = firebase.database();*

*console.log(database);*

*createCanvas(500,500);*

*ball = createSprite(250,250,20,20);*

*ball.shapeColor = rgb(255,255,0); //Color amarillo en rgb*

*var ballPosition = database.ref('Pelota/posicion');*

*ballPosition.on("value", readPosition, showError);}*

En la función readPosition podemos leer la posición del valor en la base de datos. Asignamos los valores x e y de la posición de la pelota en la base de datos al sprite de la pelota.

También, necesitamos escribir el cambio de posición cuando se presionan las teclas de flecha. El profesor muestra cómo escribir en la base de datos en tiempo real.

**.set()** se usa para establecer el valor en la base de datos.

Básicamente, las presiones de flecha están cambiando los valores de posición de la pelota, solo en la base de datos. En nuestra aplicación, leemos esos valores y mostramos la pelota en esa posición.

Veamos qué sucede si ahora ejecutamos la aplicación en dos navegadores diferentes. El profesor muestra la aplicación en dos navegadores diferentes.

La pelota en ambos navegadores parece moverse en posiciones sincronizadas cuando se presionan las teclas de flecha. Sin embargo, **hay algunos errores: ¿Puedes notarlos?** **REA:** Si se presiona la tecla de flecha inmediatamente cuando se inicia la aplicación, la aplicación muestra un error.

Además, cuando se inicia la aplicación, la posición inicial es 250, 250 antes de que se sincronice con los valores de la base de datos de la posición de la pelota. ¿Puedes pensar por qué están presentes estos errores y cómo podemos solucionarlos? REA: variado

En **sketch.js**

*function draw(){*

*background(255, 87, 34);*

*if(keyDown(LEFT\_ARROW)){ writePosition(-1,0); // Antes changePosition(-1,0);}*

*else if(keyDown(RIGHT\_ARROW)){ writePosition(1,0); }*

*else if(keyDown(UP\_ARROW)){ writePosition(0,-1); }*

*else if(keyDown(DOWN\_ARROW)){writePosition(0,1);}*

*drawSprites();}*

function writePosition(x,y){

database.ref('Pelota/posicion').set({'x': position.x + x, 'y': position.y + y});}

*function readPosition(data){ //Función para leer la posición de la DB*

*position = data.val();*

*console.log("Posicion X:", position.x);*

*console.log("Posicion Y:", position.y);*

*ball.x = position.x;*

*ball.y = position.y; }*

*function showError(){ console.log("Error de escritura en la DB") }*

¿Por qué no intentas ver si puedes arreglarlos por tu cuenta?

Permita que el alumno ejecute la aplicación en dos **navegadores diferentes**, y verifique el movimiento de la pelota - si están sincronizados.

¿Cuál es el error que estás viendo?. REA: Si se presiona la tecla de flecha inmediatamente cuando se inicia la aplicación, la aplicación muestra un error. Además, cuando se inicia la aplicación, la posición inicial es 250, 250 antes de que se sincronice con los valores de la base de datos de la posición de la pelota.

¿Por qué crees que existe este error?. El alumno pasa algún tiempo pensando en corregir el error.

Estamos declarando la variable 'posición' en la línea 2, pero la aplicación tarda algún tiempo en leer los valores de la base de datos y asignarlos a este valor.

Hasta entonces, el valor de la posición no está definido, y el sprite de la pelota se muestra en el valor predeterminado de 250, 250 que usamos para crearlo.

Cuando presionamos la tecla de flecha inmediatamente al inicio de la aplicación, estamos tratando de escribir los valores de posición 'indefinidos' en la base de datos.

Podemos arreglar esto dibujando la pelota, o escribiendo en la base de datos solo cuando 'posición' NO ES IGUAL a indefinido.

Dentro de **sketch.js**

*background(255,0,255);*

*if(position !== undefined){*

*if(keyDown(LEFT\_ARROW)){*

**Conclusión:** Aprendiste a usar una base de datos remota en tiempo real en tu aplicación, y también hiciste un poco de depuración. ¿Cómo te sientes con la clase de hoy?.

Acabamos de empezar con el juego de carreras de autos multijugador, y la conexión a la base de datos es el primer concepto que hemos descubierto.

Aprenderemos más para hacer nuestro primer juego de carreras de autos multijugador.

**Reto:** Paseo en globo aereostático

**Objetivo:** Hoy, aprendiste cómo crear una base de datos remota en tiempo real; cómo leer, escribir y conectarte a una base de datos remota en tiempo real.

En este proyecto, tendrás que aplicar lo que has aprendido en la clase, y configurar una base de datos en tiempo real para tu juego. También agrega una imagen de fondo para el juego, y agrega un globo

aerostático y eventos keyPress.

**Historia:** Kanchan fue a un evento y allí vio un paseo en globo aerostático. Pero, debido a la falta de tiempo, se perdió el viaje.

Después de regresar a casa, planeó crear su propio globo aerostático virtual en el que podría viajar con sus primos. Pero ella no es buena codificando, ¿puedes ayudarla a crear esto?