**PRO-C38**

Tema: Camara de Juego y Tamaño de Pantalla

**Descripción:** Los alumnos usan los datos de la base de datos, para diseñar el juego de carreras de autos, cuando el juego está en play (jugar); y usan la Cámara del Juego, para enfocar el juego en el jugador activo que está jugando. Los alumnos también ajustan el lienzo del juego, al tamaño del dispositivo de visualización.

**Objetivos:** ● Mostrar el lienzo del juego y todos los elementos del juego, con respecto al tamaño de la pantalla del dispositivo. ●  Utilizar los datos de la base de datos, para diseñar un juego de carreras de coches cuando el juego esté en estado play. ●  Enfocar la cámara del juego en el jugador activo.

¿Recuerdas en dónde nos quedamos en la sesión de la última clase?. **REA:** ¡Sí! Habíamos diseñado el juego dentro de la base de datos, y estábamos mostrando la información del jugador y la distancia que recorrían en la pantalla.

¡Sí! Hoy usaremos los datos de la base de datos, y usaremos sprites para diseñar el juego de carreras en acción. Pero antes de eso, hablemos de otro problema.

En casi todas las clases, hemos estado diseñando nuestro juego en un tamaño de pantalla de 400X400. Pero los diferentes dispositivos, tienen diferentes tamaños de pantalla.

¿Cómo hacemos que nuestro juego se muestre en pantalla completa, en cualquier dispositivo?

Hoy dedicaremos tiempo a aprender a hacer eso. ¡Además, aprenderemos acerca de las Cámaras del Juego!

¿Sabes que cada juego tiene una cámara desde la que ves el juego? La cámara del juego, puede cambiar para mostrarte el mismo juego desde diferentes ángulos - al igual que podrías ver un partido de cricket, o fútbol desde diferentes ángulos de la cámara.

Asegúrate de que gameState y playerCount sean 0.

Hemos estado diseñando nuestro juego con un tamaño de pantalla de 400 como ancho, y 400 como alto en todos nuestros juegos. Pero, ¿cuál es el problema con este enfoque?. **REA:** - No podemos crear un juego utilizando todo el tamaño de la pantalla del dispositivo. - Los diferentes dispositivos tendrán diferentes tamaños, y podría ser un problema usar siempre 400 X 400.

En P5, el uso de 'displayWidth' y 'displayHeight' capturará automáticamente el tamaño del dispositivo en el que se esté ejecutando el programa. Podemos usarlo en nuestro código.

La profesora usa 'displayWidth' y 'displayHeight' en el código para crear el lienzo para llenar el navegador.

*(La profesora puede usar 'displayWidth-20' o 'displayHeight-30' para que los márgenes del lienzo sean visibles).* Las posiciones de los elementos del formulario, también deben ajustarse para que los botones, el cuadro de entrada de datos, etc. sean visibles en los lugares correctos.

¿Cuáles son las diferentes posiciones de los elementos que deben ajustarse?. **Nota:** La entrada y los botones también tienen tamaño. Debemos tener eso en cuenta mientras intentamos colocarlos en la pantalla.

**Sketch.js**

*canvas = createCanvas(displayWidth -20, displayHeight-30);*

**Form.js**

*display(){*

*var title = createElement("h1");*

*title.html("Car Racing Game");*

*title.position(displayWidth-190,0);*

*this.input.position(displayWidth/2,displayHeight/2);*

*this.button.position(displayWidth/2+70, displayHeight/2+70);*

*this.button.mousePressed(()=>{*

*this.input.hide();*

*this.button.hide();*

*player1.name = this.input.value*

*playerCount+=1;*

*player1.index = playerCount;*

*player1.update();*

*player1.updateCount(playerCount);*

*this.greeting.html("Welcome to my life " + player1.name);*

*this.greeting.position(displayWidth/2,displayHeight/2); }); }}*

Ahora tenemos una pantalla completa en nuestro juego.

Anteriormente, habíamos diseñado el juego para que entrara en estado de 'juego' tan pronto como 4 jugadores iniciaran sesión.

En playState, pudimos ver el nombre y la distancia recorrida por cada jugador en la base de datos, pero ahora, podemos construir el juego con cuatro autos controlados por 4 jugadores. ¿Te gustaría intentarlo?

Ahora creemos un juego de carreras de autos simple dentro de la función play() en Game.js.

Antes de eso, en nuestro archivo sketch.js, creemos 4 variables de autos: car1, car2, car3 y car4 (auto1, auto2, auto3, auto4).

También crea una matriz 'cars' que almacenará los 4 autos.  
¿Recuerdas las matrices y cómo podemos almacenar elementos dentro de una matriz?. **REA:** Las matrices son tipos especiales de objetos donde podemos almacenar datos. Podemos almacenar datos dentro de una matriz, asignando un índice a un elemento.

En el estado del juego 'start' -inicio, del juego, crea el objeto para todas las variables del coche y guárdalas en la matriz.

**Sketch.js**

*var car1, car2, car3, car4, cars;*

**Game.js**

car1 = createSprite(100,200);

*car2 = createSprite(200,200);*

*car3 = createSprite(300,200);*

*car4 = createSprite(400,200);*

*cars = [car1, car2, car3, car4];*

Para cada jugador en allPlayers -todos los jugadores, dibujemos el sprite del auto en una posición en particular en la pantalla.

Para la posición x del sprite, los dibujaremos a distancias iguales entre sí. Para la posición y del sprite, usaremos los datos de distancia de la base de datos.

Démosle un color diferente al jugador activo en el navegador. *El alumno escribe la condición if para colorear el jugador activo en rojo.*

*El alumno ejecuta el código en el navegador iniciando sesión para que cada usuario verifique el resultado.*

*(Asegúrate de que gameState y playCount sean 0).*

*El alumno presiona la tecla de flecha hacia arriba en diferentes pestañas, para ver cómo se mueven los sprites del auto.*

***Game.js***

*car1 = createSprite(100,200,50,50);*

*car2 = createSprite(300,200,50,50);*

*car3 = createSprite(300,200,50,50);*

*car4 = createSprite(300,200,50,50);*

*cars = [car1, car2, car3, car4];*

*play(){*

*form1.hide();*

*Player.getPlayerInfo();*

*if(allplayers !== undefined){*

*var index = 0; var x=0;*

*var y;*

*for(var plr in allplayers){*

*index = index + 1;*

*x = x +200;*

*y = displayHeight - allplayers[plr].distance;*

*cars[index-1].x = x;*

*cars[index-1].y = y;*

*if(index === player1.index){*

*cars[index - 1].shapeColor = rgb(205, 92, 92);*

*}}}*

*if(keyIsDown(UP\_ARROW) && player1.index !== null){*

*player1.distance +=10;*

*player1.update(); }*

*drawSprites(); }*

¿Cuál es el problema del juego en este momento?. **REA:** Cuando los sprites se han movido más allá de la pantalla, no podemos verlos. Todos los jugadores ven lo mismo en el juego.

Aquí es donde entra el concepto de una cámara del juego. La Cámara de Juego nos permite cambiar cómo y desde dónde estamos viendo el juego.

Queremos que la cámara del juego se enfoque en el auto de cada jugador. Podemos establecer la posición de la cámara en el juego de manera diferente para cada jugador.

*Guíe al alumno para que establezca la posición de la cámara para cada jugador en el juego.*

*Aquí estamos configurando la posición x de la cámara, y la posición y según el jugador.*

*camera.position.x = displayWidth/2;*

*camera.position.y = cars[index-1].y*

Con esto, cerramos la clase de hoy. ¿Cómo te sientes?.

¿Podrías resumir el aprendizaje de la clase de hoy? . **REA:** - Aprendimos a mostrar una aplicación que se ajusta al tamaño de la pantalla. - Aprendimos a usar una cámara del juego.