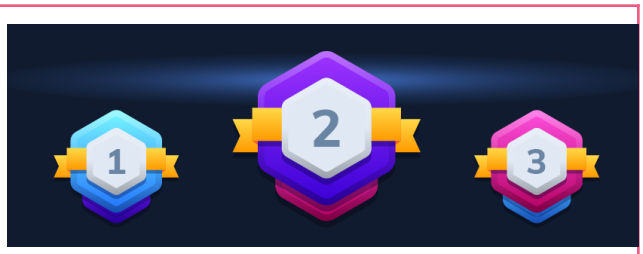
POSICIÓN DE LOS JUGADORES



¿Cuál fue nuestro objetivo para este módulo?

Construimos un mecanismo de posicionamiento que clasificó al jugador de acuerdo con su desempeño en el juego de carreras de autos.

¿Qué logramos en clase el día de hoy?

- Consultamos la base de datos creada para actualizar y leer la posición del jugador de la base de datos.
- Definimos una línea de meta para que los jugadores terminen el juego.
- Cambiamos **gameState** a 2.
- Utilizamos **swal()** para mostrar un mensaje emergente para la posición.

¿Qué conceptos y bloques de código vimos hoy?

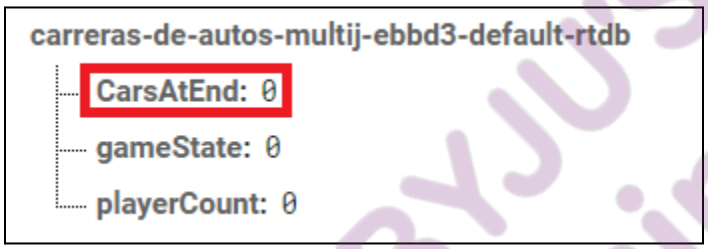
- **Swal()** de SweetAlert.

¿Cómo hicimos las actividades?

1. Creamos una propiedad en el **constructor()** de **player.js**.

```
class Player {
  constructor() {
    this.name = null;
    this.index = null;
    this.positionX = 0;
    this.positionY = 0;
    this.rank = 0;
    this.fuel = 185;
    this.life = 185;
    this.score = 0;
  }
```

2. Agregamos un campo **"CarsAtEnd"** en la base de datos.



3. Escribimos un método **getCarsAtEnd()** para leer la posición de la base de datos.
    - Escribimos una función **estática updateCarsAtEnd()** para actualizarla.

```
getCarsAtEnd(){
  database.ref('carsAtEnd').on("value",(data)=>{
    this.rank = data.val()
  })
}

static updateCarsAtEnd(rank) {
  database.ref("/").update({
    carsAtEnd: rank
  });
}
```

- Llamamos a **getCarsAtEnd()** en la función **play()** de **game.js**

4. Creamos una **const finishLine** que declara que la línea de meta es **100 px** menor que la longitud de **trackImage**.
    - Escribimos una **condición if** para comparar la posición Y del auto del jugador con la línea de meta.
    - Cuando el auto cruzó la línea de meta, **gameState** cambió a **2 (END)**
    - Llamamos a la función **showRank().**
    - Al mismo tiempo, actualizamos los datos del jugador en la base de datos.

```
const finshLine = height * 6 - 100;

    if (player.positionY > finshLine) {
      gameState = 2;
      player.rank += 1;
      Player.updateCarsAtEnd(player.rank);
      player.update();
      this.showRank();
    }
```

5. Creamos un método **showRank()** para mostrar la posición del jugador usando **swal()** en **game.js**.
    - La función **swal()** acepta propiedades como:
        - título:
        - texto:
        - URL de la imagen:
        - tamaño de la imagen:
        - Texto del botón de confirmación.:

```
showRank() {
  swal({
    title: `¡Impresionante!${"\n"}Posición${"\n"}${player.rank}`,
    text: "Cruzaste la línea de meta con éxito",
    imageUrl:
```

```
"https://raw.githubusercontent.com/vishalgaddam873/p5-multiplayer-car-race-game/master/assets/cup.png",
    imageSize: "100x100",
    confirmButtonText: "Ok"
  });
 }
```

6. Agregamos la biblioteca **sweetalert.css** en **index.html** para usar la función **swal()**.

```
<!-- Sweet Alert c40-->
<script
  src="https://code.jquery.com/jquery-3.5.1.min.js"
  integrity="sha256-9/aliU8dGd2tb6OSsuzixeV4y/faTqgFtohetphbbj0="
  crossorigin="anonymous"
></script>
<script src="./lib/sweetalert.min.js"></script>
<link rel="stylesheet" type="text/css" href="./lib/sweetalert.css" />
```

¿Qué sigue?

En la siguiente clase, crearemos una barra de combustible y una barra de vida. También mostraremos el mensaje de fin del juego.

AMPLÍEN SU CONOCIMIENTO:

1. Obtengan más información sobre SweetAlert en: https://sweetalert.js.org/guides/