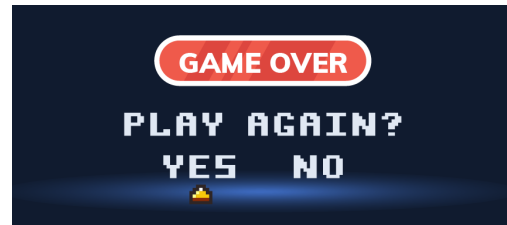


FIN DEL JUEGO



### ¿Cuál fue nuestro objetivo para este módulo?

Construimos una barra de progreso para las propiedades de combustible y de vida del jugador. Además, construimos un mensaje emergente de fin del juego.

### ¿Qué logramos en clase el día de hoy?

- Creamos una barra de progreso para el combustible y la vida.
- Actualizamos la función **handleFuel()** para mostrar el estatus del combustible.
- Usamos **swal()** para mostrar un mensaje emergente al ganar/acabar el juego.

### ¿Qué conceptos y bloques de código vimos hoy?

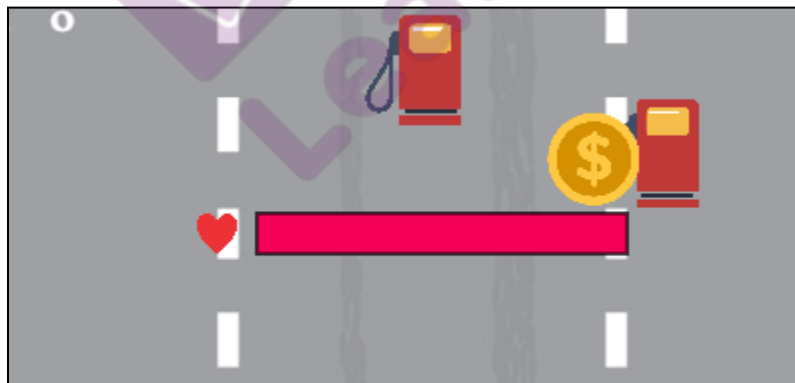
- Creamos una barra de progreso.
- **Swal()** de SweetAlert.
- Condiciones de fin del juego y ganar el juego.
- Mover el auto sin presionar ninguna tecla (usando una bandera).

### ¿Cómo hicimos las actividades?

1. Creamos una barra de progreso para la vida y el combustible.
  - Creamos **showLife()** para las monedas de poder.
  - Asignamos la posición según el tamaño de la pantalla.
  - Llamamos a esta función en el método **play()**.

```
showLife() {  
    push();  
    image(lifeImage, width / 2 - 130, height - player.positionY - 400, 20, 20);  
    fill("white");  
    rect(width / 2 - 100, height - player.positionY - 400, 185, 20);  
    fill("#f50057");  
    rect(width / 2 - 100, height - player.positionY - 400, player.life, 20);  
    noStroke();  
    pop();  
}
```

OUTPUT:



- Creamos **showFuelBar()** para el combustible dentro de game.js.
- Llamamos a esta función en el método **play()**.

```
showFuelBar() {  
  push();  
  image(fuelImage, width / 2 - 130, height - player.positionY - 100, 20, 20);  
  fill("white");  
  rect(width / 2 - 100, height - player.positionY - 100, 185, 20);  
  fill("#ffc400");  
  rect(width / 2 - 100, height - player.positionY - 100, player.fuel, 20);  
  noStroke();  
  pop();  
}
```

OUTPUT:



2. Reducimos el combustible:
  - Revisamos si el auto del jugador se movía o no.
  - Para ello, creamos una propiedad **this.playerMoving** en el **constructor()** de **game.js**.

```
class Game {  
  constructor() {  
    this.resetTitle = createElement("h2");  
    this.resetButton = createButton("");  
  
    this.leadeboardTitle = createElement("h2");  
  
    this.leader1 = createElement("h2");  
    this.leader2 = createElement("h2");  
    this.playerMoving = false;  
  }  
}
```

- Esto funcionaba como una bandera para dar la señal **true** o **false**.
- **False** indicaba que el auto no se mueve y **true** indicaba que el auto se mueve.
- Actualizamos **this.playerMoving** a verdadero en **handlePlayerControls()**.

```
handlePlayerControls() {  
  if (keyIsDown(UP_ARROW)) {  
    this.playerMoving = true;  
    player.positionY += 10;  
    player.update();  
  }  
}
```

3. Modificamos **handleFuel(index)** para reducir el combustible cuando el auto se mueve y cuando está vacío, el juego debe terminar.

```
handleFuel(index) {  
  // Adding fuel  
  cars[index - 1].overlap(fuels, function(collector, collected) {  
    player.fuel = 185;  
    //collected is the sprite in the group collectibles that triggered  
    //the event  
    collected.remove();  
  });  
  
  // Reducing Player car fuel  
  if (player.fuel > 0 && this.playerMoving) {  
    player.fuel -= 0.3;  
  }  
  
  if (player.fuel <= 0) {  
    gameState = 2;  
    this.gameOver();  
  }  
}
```

4. Creamos la función **gameOver()** que usa **swal()** para mostrar el mensaje emergente de fin del juego.

```
gameOver() {  
  swal({  
    title: `Fin del juego`,  
    text: "¡Ups! ¡Perdiste la carrera!",  
    imageUrl:  
      "https://cdn.shopify.com/s/files/1/1061/1924/products/Thumbs_Down_Sign_Emoji_Icon_ios10_grande.png",  
    imageSize: "100x100",  
    confirmButtonText: "Gracias por jugar"  
  });  
}
```

5. Escribimos una condición para mover al auto lentamente, incluso cuando la flecha arriba del teclado no es presionada.
  - Usamos la propiedad **this.playerMoving**.

```
if (this.playerMoving) {  
  player.positionY += 5;  
  player.update();  
}
```

### ¿Qué sigue?

En la siguiente clase, haremos a nuestro juego más desafiante al agregar algunos obstáculos y hacer colisionar al auto con ellos.

### AMPLÍA TU CONOCIMIENTO:

1. Obtén más información sobre SweetAlert en: <https://sweetalert.js.org/guides/>