

ANIMAÇÃO

**Qual é o nosso OBJETIVO neste MÓDULO?**

Nesta aula, aprendemos a usar o conceito de colisão para detectar e adicionar uma imagem de explosão (BOOM!) quando ocorre uma colisão entre os carros.

O que APRENDEMOS na aula de HOJE?

- Reduzir a vida de um jogador quando os carros colidirem entre si.
- Alterar a animação do carro quando o tempo de vida do jogador for reduzido a 0.
- Desabilitar o controle dos carros, uma vez que a animação seja alterada.

Quais CONCEITOS/BLOCOS DE CÓDIGO estudamos hoje?

- Colisão.
- Afastar o carro após a primeira colisão.
- Desativar o controle sem alterar o **gameState**.

Como FIZEMOS as atividades?

1. Carregamos a imagem para a explosão em **sketch.js**.

```
sketch.js > ...
1  var canvas;
2  var backgroundImage, car1_img, car2_img, track;
3  var fuelImage, powerCoinImage, lifeImage;
4  var obstacle1Image, obstacle2Image; //C
5  var database, gameState;
6  var form, player, playerCount;
7  var allPlayers, car1, car2, fuels, powerCoins, obstacle1, obsta
8  var cars = [];
9  var blastImage;
10 var currentLife = 0; //C42// SA
11
12 function preload() {
13   backgroundImage = loadImage("./assets/background.png");
14   car1_img = loadImage("./assets/car1.png");
15   car2_img = loadImage("./assets/car2.png");
16   track = loadImage("./assets/track.jpg");
17   fuelImage = loadImage("./assets/fuel.png");
18   powerCoinImage = loadImage("./assets/goldCoin.png");
19   lifeImage = loadImage("./assets/life.png");
20   obstacle1Image = loadImage("./assets/obstacle1.png"); // C41
21   obstacle2Image = loadImage("./assets/obstacle2.png"); // C41
22   blastImage = loadImage("./assets/blast.png"); //C42 //SA
23 }
```

2. Adicionamos a imagem no método **start()** de **Game.js**.

```
start() {  
    player = new Player();  
    playerCount = player.getCount();  
  
    form = new Form();  
    form.display();  
  
    car1 = createSprite(width / 2 - 50, height - 100);  
    car1.addImage("car1", car1_img);  
    car1.scale = 0.07;  
  
    car1.addImage("blast",blastImage);  
  
    car2 = createSprite(width / 2 + 100, height - 100);  
    car2.addImage("car2", car2_img);  
    car2.scale = 0.07;  
  
    car2.addImage("blast",blastImage);  
  
    cars = [car1, car2];  
}
```

3. Alteramos a animação do carro com base no valor de **player.life**.
- O método **play()** está sendo executado em cada **frameCount** da função **draw()**.
 - Obtivemos uma contagem de **vida** atualizada para o jogador em cada **frameCount**.
 - Usamos a propriedade **player.life** para alterar a animação do carro.

```
//índice da array
var index = 0;
for (var plr in allPlayers) {
    //adicione 1 ao índice para cada loop
    index = index + 1;

    //use os dados do banco de dados para exibir o
    var x = allPlayers[plr].positionX;
    var y = height - allPlayers[plr].positionY;

    var currentLife = allPlayers[plr].life;
    if (currentLife <=0){
        cars[index - 1].changeImage("blast");
        cars[index-1].scale = 0.3;
    }

    cars[index - 1].position.x = x;
    cars[index - 1].position.y = y;
}
```

- 1.
4. Desativamos o controle após **player.life = 0**:
 - Criamos uma propriedade na função **constructor()** de **Game.js** para verificar o status da explosão.
 - Mantivemos **this.blast = false** inicialmente.

```
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;  
  
    this.leftKeyActive = false;  
    this.blast = false;  
  }  
}
```

- Alteramos **this.blast** para **true**, quando **player.life <= 0**.

```
if (index === player.index) {  
  stroke(10);  
  fill("red");  
  ellipse(x, y, 60, 60);  
  
  this.handleFuel(index);  
  this.handlePowerCoins(index);  
  this.handleObstacleCollision(index);  
  
  if (player.life <= 0) {  
    this.blast = true;  
    this.playerMoving = false;  
  }  
}
```

5. Adicionamos a condição **if** em **handlePlayerControl()** para ler o controle pela tecla de seta quando **this.blast** for **false**.

```
handlePlayerControls() {  
  if (!this.blast) {  
    if (keyIsDown(UP_ARROW)) {  
      this.playerMoving = true;  
      player.positionY += 10;  
      player.update();  
    }  
  
    if (keyIsDown(LEFT_ARROW) && player.positionX > width / 3 - 50) {  
      this.leftKeyActive = true;  
      player.positionX -= 5;  
      player.update();  
    }  
  
    if (keyIsDown(RIGHT_ARROW) && player.positionX < width / 2 + 300) {  
      this.leftKeyActive = false;  
      player.positionX += 5;  
      player.update();  
    }  
  }  
}
```

6. Criamos uma nova função **handleCarACollisionwithCarB(index)**.

```
handleCarACollisionWithCarB(index) {  
  if(index === 1){  
    if (cars[index - 1].collide(cars[1])){  
      if (this.leftKeyActive){  
        player.positionX += 100;  
      } else {  
        player.positionX -= 100;  
      }  
  
      if (player.life > 0) {  
        player.life -= 185 / 4;  
      }  
  
      player.update();  
    }  
  }  
  
  if(index === 2){  
    if (cars[index - 1].collide(cars[0])){  
      if (this.leftKeyActive){  
        player.positionX += 100;  
      } else {  
        player.positionX -= 100;  
      }  
  
      if (player.life > 0) {  
        player.life -= 185 / 4;  
      }  
  
      player.update();  
    }  
  }  
}
```

- E a chamamos no método **play()**

```
cars[index - 1].position.x = x;
cars[index - 1].position.y = y;

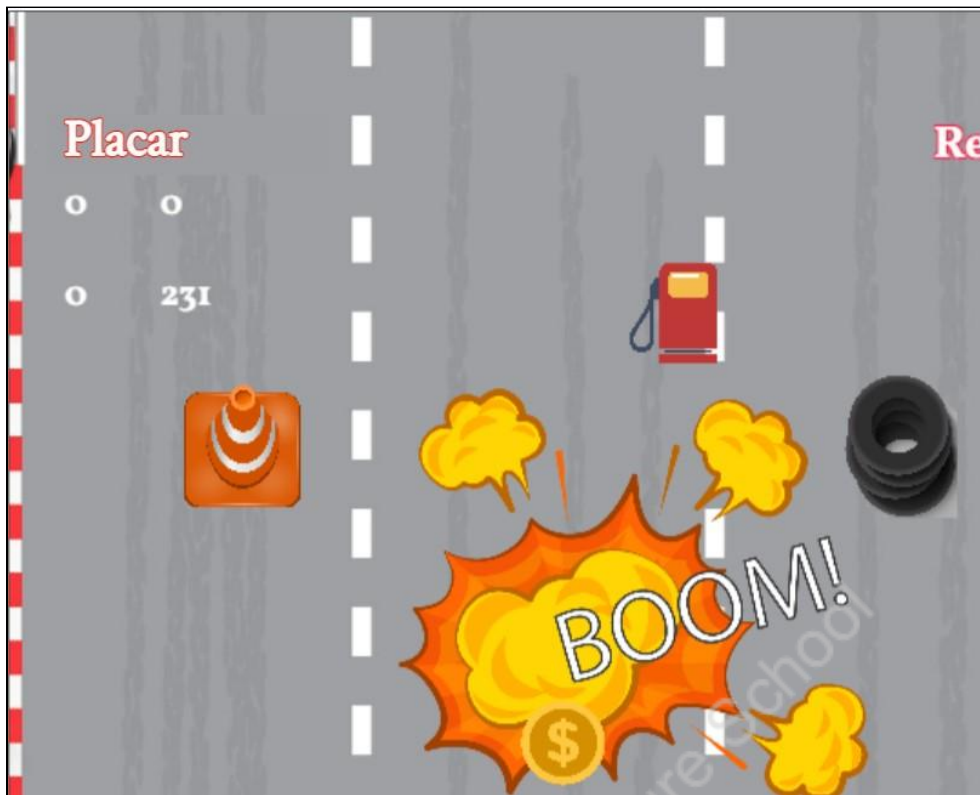
if (index === player.index) {
  stroke(10);
  fill("red");
  ellipse(x, y, 60, 60);

  this.handleFuel(index);
  this.handlePowerCoins(index);
  this.handleObstacleCollision(index);
  this.handleCarACollisionWithCardB(index);

  if (player.life <= 0) {
    this.blast = true;
    this.playerMoving = false;
  }
}
```

1.

RESULTADO:



Qual é o PRÓXIMO passo?

Na próxima aula, resolveremos as dúvidas dos projetos.

AMPLIE SEUS CONHECIMENTOS

Salve o seguinte link para saber mais sobre a Anatomia do Desenvolvimento de Jogos, criada pela Mozilla e por contribuidores individuais:

<https://developer.mozilla.org/en-US/docs/Games/Anatomy>

Sobre MDN por [Mozilla Contributors](#) está licenciado sob [CC-BY-SA 2.5](#).