02

O alvo aleatório!

Transcrição

Vamos criar o `programa5.html` e nele teremos nossa já conhecida função `desenhaCirculo` e `limpaTela` . A diferença é que dessa vez `desenhaCirculo` recebe como parâmetro uma cor para que possamos desenhar círculos de cores diferentes.

```
<canvas width="600" height="400"></canvas>

<script>

    var tela = document.querySelector('canvas');
    var pincel = tela.getContext('2d');

    pincel.fillStyle = 'lightgray';
    pincel.fillRect(0, 0, 600, 400);

    function desenhaCirculo(x, y, raio, cor) {

        pincel.fillStyle = cor;
        pincel.beginPath();
        pincel.arc(x, y, raio, 0, 2 * Math.PI);
        pincel.fill();
    }

    function limpaTela() {

        pincel.clearRect(0, 0, 600, 400);
    }

    // testando a função
    desenhaCirculo(200, 200, 10, 'red');

</script>
```

O primeiro passo é construirmos nosso alvo. Para isso, precisaremos criar três circulos cada um com o raio maior do que o outro, desenhando do maior para o menor:

```
<canvas width="600" height="400"></canvas>

<script>

    var tela = document.querySelector('canvas');
    var pincel = tela.getContext('2d');

    pincel.fillStyle = 'lightgray';
    pincel.fillRect(0, 0, 600, 400);

    function desenhaCirculo(x, y, raio, cor) {

        pincel.fillStyle = cor;
```

```
        pincel.beginPath();
        pincel.arc(x, y, raio, 0, 2 * Math.PI);
        pincel.fill();
    }

    function limpaTela() {

        pincel.clearRect(0, 0, 600, 400);
    }

    desenhaCirculo(200,200, 30, 'red');
    desenhaCirculo(200,200, 20, 'white');
    desenhaCirculo(200,200, 10, 'red');

</script>
```

Com isso já somos capazes de desenhar nosso alvo. Contudo, você deve ter reparado que o menor círculo é raio 10 e os depois são acrescidos de 10 e de 20 respectivamente. Podemos declarar a variável `raio` inicializando-a com `10`. Mais tarde, se quisermos mudar o tamanho do alvo, mudamos em um local apenas:

```
<canvas width="600" height="400"></canvas>

<script>

    var tela = document.querySelector('canvas');
    var pincel = tela.getContext('2d');

    pincel.fillStyle = 'lightgray';
    pincel.fillRect(0, 0, 600, 400);

    var raio = 10;

    function desenhaCirculo(x, y, raio, cor) {

        pincel.fillStyle = cor;
        pincel.beginPath();
        pincel.arc(x, y, raio, 0, 2 * Math.PI);
        pincel.fill();
    }

    function limpaTela() {
        pincel.clearRect(0, 0, 600, 400);
    }

    desenhaCirculo(200,200, raio + 20, 'red');
    desenhaCirculo(200,200, raio + 10, 'white');
    desenhaCirculo(200,200, raio, 'red');

</script>
```

Desenhando um alvo

Veja que as três chamadas à `desenhaCirculo` são feitas para criar um alvo, sendo assim, vamos isolá-las em uma função chamada `desenhaAlvo`:

```
<canvas width="600" height="400"></canvas>

<script>

    var tela = document.querySelector('canvas');
    var pincel = tela.getContext('2d');

    pincel.fillStyle = 'lightgray';
    pincel.fillRect(0, 0, 600, 400);

    var raio = 10;

    function desenhaCirculo(x, y, raio, cor) {

        pincel.fillStyle = cor;
        pincel.beginPath();
        pincel.arc(x, y, raio, 0, 2 * Math.PI);
        pincel.fill();
    }

    function limpaTela() {
        pincel.clearRect(0, 0, 600, 400);
    }

    function desenhaAlvo() {

        desenhaCirculo(200,200, raio + 20, 'red');
        desenhaCirculo(200,200, raio + 10, 'white');
        desenhaCirculo(200,200, raio, 'red');
    }

    desenhaAlvo();

</script>
```

Contudo, nossa função `desenhaAlvo` tem que ser capaz de desenhar um alvo em diferentes coordenadas x e y, sendo assim, vamos fazer com que a função `desenhaAlvo` receba como parâmetro as coordenadas:

```
<canvas width="600" height="400"></canvas>

<script>

    var tela = document.querySelector('canvas');
    var pincel = tela.getContext('2d');

    pincel.fillStyle = 'lightgray';
    pincel.fillRect(0, 0, 600, 400);

    var raio = 10;

    function desenhaCirculo(x, y, raio, cor) {

        pincel.fillStyle = cor;
        pincel.beginPath();
        pincel.arc(x, y, raio, 0, 2 * Math.PI);
```

```
        pincel.fill();
    }

    function limpaTela() {
        pincel.clearRect(0, 0, 600, 400);
    }

    function desenhaAlvo(x, y) {

        desenhaCirculo(x,y, raio + 20, 'red');
        desenhaCirculo(x,y, raio + 10, 'white');
        desenhaCirculo(x,y, raio, 'red');
    }

    desenhaAlvo(200, 200);

</script>
```

Gerando posições aleatória para nosso alvo

Excelente, mas queremos que as coordenadas x e y passadas para `desenhaAlvo` sejam aleatórias. Contudo, devem respeitar a faixa de 0 à 600 para o eixo x e 0 à 400 para o eixo y. Vamos isolar a lógica que gera o número aleatório na função `sorteiaPosicao` :

```
<canvas width="600" height="400"></canvas>

<script>

    var tela = document.querySelector('canvas');
    var pincel = tela.getContext('2d');

    pincel.fillStyle = 'lightgray';
    pincel.fillRect(0, 0, 600, 400);

    var raio = 10;

    function desenhaCirculo(x, y, raio, cor) {

        pincel.fillStyle = cor;
        pincel.beginPath();
        pincel.arc(x, y, raio, 0, 2 * Math.PI);
        pincel.fill();
    }

    function limpaTela() {
        pincel.clearRect(0, 0, 600, 400);
    }

    function desenhaAlvo(x, y) {

        desenhaCirculo(x,y, raio + 20, 'red');
        desenhaCirculo(x,y, raio + 10, 'white');
        desenhaCirculo(x,y, raio, 'red');
    }
```

```
    function sorteiaPosicao(maximo) {
        return Math.floor(Math.random() * maximo);
    }

    desenhaAlvo(200, 200);

</script>
```

No lugar de usarmos a já conhecida `Math.round` para arredondar o resultado, vamos usar dessa vez `Math.floor` . Esta última arredonda o número para baixo. Daí, para termos um número de 0 até um valor máximo, multiplicamos o resultado de `Math.random()` vezes o valor máximo passado para a função.

Vamos agora plotar aleatoriamente o alvo na tela:

```
<canvas width="600" height="400"></canvas>

<script>

    var tela = document.querySelector('canvas');
    var pincel = tela.getContext('2d');

    pincel.fillStyle = 'lightgray';
    pincel.fillRect(0, 0, 600, 400);

    var raio = 10;

    function desenhaCirculo(x, y, raio, cor) {

        pincel.fillStyle = cor;
        pincel.beginPath();
        pincel.arc(x, y, raio, 0, 2 * Math.PI);
        pincel.fill();
    }

    function limpaTela() {

        pincel.clearRect(0, 0, 600, 400);
    }

    function desenhaAlvo(x, y) {

        desenhaCirculo(x,y, raio + 20, 'red');
        desenhaCirculo(x,y, raio + 10, 'white');
        desenhaCirculo(x,y, raio, 'red');
    }

    function sorteiaPosicao(maximo) {
        return Math.floor(Math.random() * maximo);
    }

    var xAleatorio = sorteiaPosicao(600);
    var yAleatorio = sorteiaPosicao(400);
    desenhaAlvo(xAleatorio, yAleatorio);

</script>
```

Experimente recarregar a página algumas vezes. Veja que a cada recarregamento o alvo será plotado em um local diferente do nosso canvas!

Atualizando a posição do alvo na tela

Precisamos plotar um alvo a cada um minuto, apagando o anterior para que seja exibido apenas um alvo por vez na tela.

Para isso, vamos criar a função `atualizaTela` que se encarregará de desenhar um alvo usando coordenadas aleatórias. Aliás, essa mesma função será passada para `setInterval` para que seja executada a cada um segundo:

```
<canvas width="600" height="400"></canvas>

<script>

    var tela = document.querySelector('canvas');
    var pincel = tela.getContext('2d');

    pincel.fillStyle = 'lightgray';
    pincel.fillRect(0, 0, 600, 400);

    var raio = 10;

    function desenhaCirculo(x, y, raio, cor) {

        pincel.fillStyle = cor;
        pincel.beginPath();
        pincel.arc(x, y, raio, 0, 2 * Math.PI);
        pincel.fill();
    }

    function limpaTela() {
        pincel.clearRect(0, 0, 600, 400);
    }

    function desenhaAlvo(x, y) {

        desenhaCirculo(x,y, raio + 20, 'red');
        desenhaCirculo(x,y, raio + 10, 'white');
        desenhaCirculo(x,y, raio, 'red');
    }

    function sorteiaPosicao(maximo) {
        return Math.floor(Math.random() * maximo);
    }

    function atualizaTela() {

        limpaTela(); // tem que limpar antes de desenhar um novo alvo
        var xAleatorio = sorteiaPosicao(600);
        var yAleatorio = sorteiaPosicao(400);
        desenhaAlvo(xAleatorio, yAleatorio);
    }

    setInterval(atualizaTela, 1000);
```

```
</script>
```