08

Gabarito

Transcrição

Vamos criar a função `reset()`. Seu papel será substituir a classe do frame atual pelo frame inicial. Além disso, deve mudar o estado de `current` para 0. Não podemos esquecer de adicionar a função `reset` no objeto retornado por `createSprite()`:

```
var createSprite = function(selector) {

    var hasNext = function() {

        return current + 1 <= last;
    };

    var moveFrame = function(from, to) {

        $el.removeClass(from).addClass(to);
    };

    var nextFrame = function() {

        if (hasNext()) moveFrame(frames[current], frames[++current]);
    };

    var reset = function() {

        moveFrame(frames[current], frames[0])
        current = 0;
    };

    var $el = $(selector);

    var frames = [
        'frame1', 'frame2', 'frame3', 'frame4', 'frame5',
        'frame6', 'frame7', 'frame8', 'frame9'
    ];

    var current = 0;

    var last = frames.length - 1;
    $el.addClass(frames[current]);

    return { nextFrame: nextFrame, reset: reset };
}
```

Para testarmos a função `reset()`, vamos alterar `gallows/index.html` e deixar apenas a instrução que chama `createSprite()`. Como a variável esta no escopo global, vamos acessá-la através do terminal. Nosso HTML ficará assim:

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
```

```
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width">
    <link rel="stylesheet" href="css/bootstrap.min.css">
    <link rel="stylesheet" href="css/sprite.css">
    <link rel="stylesheet" href="css/lacuna.css">
    <title>Jogo da Forca</title>
</head>
<body>
    <div class="jumbotron">
        <h2 class="text-center">Jogo da forca</h2>
    </div>
    <div class="container">

        <!-- div com a classe sprite -->
        <div class="sprite"></div>
        <ul class="lacunas text-center"></ul>

        <div class="text-center">
        <input id="entrada"
            class="entrada"
            autofocus>
        </div>
    </div>
    <script src="js/jquery.min.js"></script>
    <script src="js/sprite.js"></script>
    <script>

        var sprite = createSprite('.sprite');
    </script>
</body>
</html>
```

Agora, no console do navegador, podemos fazer assim:

```
sprite.nextFrame();
sprite.nextFrame();
sprite.nextFrame();
sprite.reset();
sprite.nextFrame();
```

Agora só falta a função `isFinihed` , aquela que indica se o sprite chegou ao fim ou não.

```
var createSprite = function(selector) {

    var hasNext = function() {

        return current + 1 <= last;
    };

    var moveFrame = function(from, to) {

        $el.removeClass(from).addClass(to);
    };

    var nextFrame = function() {
```

```
        if (hasNext()) moveFrame(frames[current], frames[++current]);
    };

    var reset = function() {

        moveFrame(frames[current], frames[0])
        current = 0;
    };

    // nova função!

    var isFinished = function() {

        return !hasNext();
    };

    var $el = $(selector);

    var frames = [
        'frame1', 'frame2', 'frame3', 'frame4', 'frame5',
        'frame6', 'frame7', 'frame8', 'frame9'
    ];

    var current = 0;

    var last = sprites.length - 1;

    $el.addClass(frames[current]);

    // adicionou isFinihed no objeto retornado
    return { nextFrame: nextFrame, reset: reset, isFinished: isFinished };
}
```

Para testar, faremos a mesma coisa no console:

```
sprite.nextFrame();
sprite.nextFrame();
sprite.nextFrame();
sprite.nextFrame();
sprite.nextFrame();
sprite.nextFrame();
sprite.nextFrame();
sprite.nextFrame();
sprite.nextFrame();
sprite.isFinished();
```