01

Jogo - PARTE 1

Transcrição

> *Começando deste ponto? Você pode fazer o DOWNLOAD (https://s3.amazonaws.com/caelum-online-public/carreira-js/stages/03-gallows.zip) completo do projeto do capítulo anterior e continuar seus estudos a partir deste capítulo.*

Nosso projeto fará uma separação marcante entre a lógica do jogo e sua apresentação (manipulação de DOM) e neste capítulo você precisará abordar o primeiro, a lógica do jogo. Não se preocupe caso você nunca tenha realizado essa divisão antes, pois a estrutura mínima necessária será passada para você. É importante que você não se afaste dessa estrutra. Já a implementação da lógica dentro dessa estrutura será de sua responsabilidade.

Crie o arquivo `gallows/js/jogo.js` e importe-o logo após do script `sprite.js` em `gallows/index.html`. Antes de liberá-lo para a implementação do código, siga os meus passos a seguir.

Estrutura: separação entre a lógica e sua apresentação

A primeira coisa que farei é usar a mesma estratégia que usamos com `sprite.js`, a diferença é que escreverei os nomes de todas as funções em português:

A API publica do nosso jogo, isto é, as funções que podem ser chamadas no objeto retornado por `criaJogo()` deverá ter essa estrutura:

```
var criaJogo = function() {

    // recebe a palavra secreta e deve atribuí-la à variável `palavraSecreta`. Vai para a próxi
    var setPalavraSecreta = function (palavra) {
        console.log('falta implementar');
    };

    // retorna as lacunas do jogo. Importante para quem for exibí-las.
    var getLacunas = function () {
        console.log('falta implementar');
    };

    // retorna a etapa atual do jogo
    var getEtapa = function () {
        console.log('falta implementar');
    };
};
```

Se você como programador achar que é necessário criar mais funções por uma questão de manutenção e legibilidade fique à vontade.

Excelente, mas ainda precisamos fazer mais coisa. Para que você entenda com clareza até onde eu quero chegar, abrirei no navegador o projeto com esta parte já implementada. A página só importa `jogo.js` e nada mais. É através do

console do navegador que interagiremos com nosso jogo.

Primeiro, vamos criar nosso jogo:

```
var jogo = criaJogo();
```

Excelente, temos uma variável `jogo` que guarda um objeto JavaScript que possui uma série de funções que dizem respeito ao jogo. Sabemos que um dos primeiro passos que o jogador executará é informar a palavra secreta. Esse passo é feito em nossa API de jogo através da chamada da função `jogo.setPalavraSecreta()` :

```
var jogo = criaJogo();
console.log(jogo.getEtapa()); // 1
jogo.setPalavraSecreta('calopsita');
```

Internamente, a função `setPalavraSecreta` deve atribuir o palavra recebida como parâmetro e guardá-la na variável `palavraSecreta` , além disso, deve criar as lacunas em branco. Lembre-se que o tamanho do array `lacunas` deve ter o mesmo tamanho da palavra secreta informada. Por fim, o jogo deve ir para a próxima etapa, que é a leitura dos chutes. Como você implementará essa parte fica ao seu critério, mas o importante é que siga a estrutura aqui definida. Inclusive, podemos testar se o array de lacunas foi criado corretamente através da função `getLacunas()` :

```
var jogo = criaJogo();
console.log(jogo.getEtapa()); // 1
jogo.setPalavraSecreta('calopsita');
console.log(jogo.getEtapa()); // 2
console.log(jogo.getLacunas()); // ['', '', '', '', '', '', '', '', '']
```

Ótimo, mas para que possamos acessar essas funções através de um objeto a função `iniciaJogo` deve retornar um objeto com essas funções. Por fim, a estrutura mínima de `inciaJogo()` ficará assim:

```
var criaJogo = function() {

    var setPalavraSecreta = function (palavra) {
        console.log('falta implementar');
    };

    var getLacunas = function () {
        console.log('falta implementar');
    };

    var getEtapa = function () {
        console.log('falta implementar');
    };

    return {

        setPalavraSecreta: setPalavraSecreta,
        getLacunas: getLacunas,
        getEtapa: getEtapa
    };
};
```

A lógica do nosso jogo ainda esta longe de terminar, mas isso é o suficiente para este vídeo. Você deve chegar até este ponto. Lembre-se que você poderá consultar minha resposta, mas faça isso depois de tentar resolver o problema primeiro.