

Desenvolvendo um jogo educativo com a biblioteca javascript HTML5 Quintus

Autor: Henrique Barone

Este mini-jogo foi desenvolvido tendo por base a tecnologias web com uso da biblioteca javascript HTML5 Quintus.

Uma biblioteca é um conjunto de funções implementadas para agilizar o processo de programação de um desenvolvimento. Existem muitas bibliotecas em várias linguagens de programação como PHP, Rubby, Java e Python. Para javascript existem bibliotecas muito usadas como jQuery, mootools e Angular. E especificamente existem também dezenas de bibliotecas que funcionam com um mecanismo para criação de jogos.

Para este exemplo, usamos o HTML5 Quintus por ser uma das mais fáceis de se aprender

Para saber mais sobre o HTML5 Quintus acesse:

Antes de começarmos a estudar os exemplos passo-a-passo deste jogo, devemos nos certificar de termos:

- O programa para criação de mapas de fase Tiled (traduzido como Mosaico em algumas distribuições) que pode ser instalado pelo seu gerenciador de programas ou com o comando `sudo apt-get install tiled` no terminal

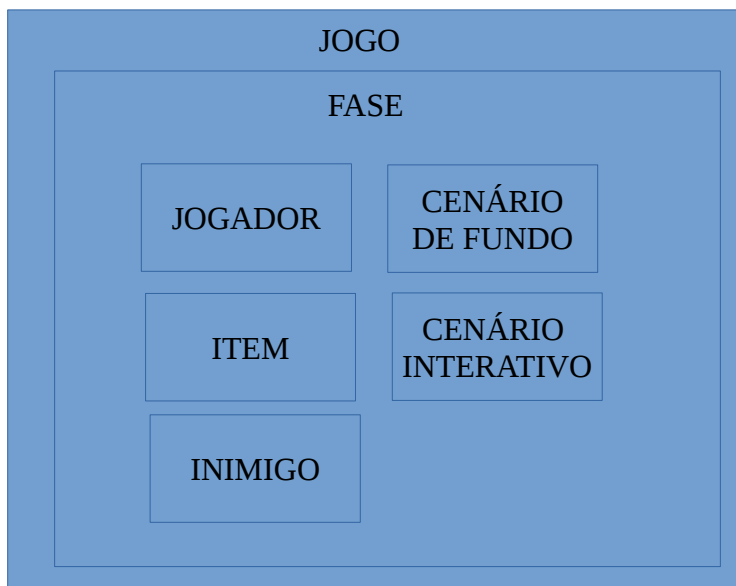
- O servidor web Apache 2 que no Linux pode ser instalado pelo seu gerenciador de programas ou com o comando `sudo apt-get install apache2` no terminal

Após instalado o Apache, abra o gerenciador de arquivos como administrador, Vá para Sistema de Arquivos e localize a pasta **var** dentro dessa pasta, há uma outra chamada **www**. É nessa última que devemos colar as pastas de exemplo do jogo.

Para rodar o jogo basta abrir o navegador e digitar na barra de endereços:

`http://localhost/nome_da_pasta`

Estrutura básica da montagem jogo



Estrutura básica de desenvolvimento com o HTML5 Quintus

- Cria e configura o mecanismo de jogo:

- **var Q = Quintus().include().setup();**
- Cria e configura o jogador como um derivado da classe sprite:
Q.sprite.extend("jogador", { });
- Cria e configura os inimigos como um derivado da classe sprite:
Q.sprite.extend("inimigo", { });
- Cria e configura os itens como um derivado da classe sprite:
- **Q.sprite.extend("caixa", { });**
- Cria e configura as cenas:
Q.scene();
- Carrega as mídias e inicia a primeira fase:
Q.load();

A sintaxe geral para criação de classes no quintus é:

Q.nome_da_classe.extend ("nome_do_elemento", { função });

No quintus criamos os elementos (jogador, inimigos, itens) através da classe **sprite**.

Portanto, temos:

Q.sprite.extend("nome_do_elemento", { função });

Material de apoio:

- video-aula de construção de fase no tiled e integração com Quintus:

<https://www.youtube.com/watch?v=YCVWAI7W6vo>