Aula JavaScript

***Canvas***

O ***canvas*** é um ***contexto*** de renderização ***dinâmica*** muito utilizado para ***animações***. Com ele podemos renderizar linhas, formas geométricas e até imagens e movimenta-las de forma dinâmica, sem precisar recarregar a página.

Se parece muito com o elemento ***<img>*** com a diferença de não possuir os atributos ***src*** e ***alt***. O elemento ***<canvas>*** tem apenas dois atributos - ***width*** e ***height***. Ambos são opcionais e podem ser aplicados utilizando as propriedades [***DOM***](https://developer.mozilla.org/en-US/docs/DOM) respectivas. Se não forem especificados, o canvas será iniciado com **300 pixels** de largura por **150 pixels** de altura. O elemento pode ser redimensionado por ***CSS***, mas durante a renderização a imagem é escalonada para caber no tamanho do layout.

Ele também será o local onde desenharemos o nosso jogo, é declarado da seguinte maneira:

<canvas width= “500” height = “500”></canvas>

Toda a renderização do ***canvas*** é feito por meio de ***script***, no nosso caso o JavaScript (ECMAScript 6), mas, antes, é necessário ter o acesso à ele no código, para isso precisamos declarar duas variáveis, uma refere-se ao elemento canvas e outro ao seu contexto (o contexto nos diz em qual dimensão estaremos desenhando).

let canvas = document.getElementById(“gameScreen”);

let context = canvas.getContext(“2d”);

Para desenhar, chamamos a variavel ***context*** e dizemos, respectivamente, seu objeto, a posição X e Y, sua largura (width) e altura (height):

 context.fillRect(40, 40, 100, 100);

Nesse caso, o objeto é um retângulo.

Podemos mudar o preenchimento colocando o seguinte comando antes do ***fillRec***:

context.fillStyle: “red”;

context.fillRect(40, 40, 100, 100);

Atenção: Todos os outros retângulos criados a partir desse ponto terão a cor vermelha, para mudar isso é só declarar outra cor antes do novo retângulo.

Sempre que formos desenhar algo no canvas, é necessário limpar ele primeiro, para isso:

context.clearRect(originX, originY, canvasWidth, canvasHeight);