Material utilizado relativo ao curso - Introdução a criação de websites com HTML5 e CSS3 - da instituição Digital Innovation One (DIO). Módulo: Recriando o jogo da cobrinha com JavaScript.

O atributo ‘defer’ da tag script nos documentos HTML faz com que o script seja carregado somente após a pagina ter sido carregada.

Primeiramente, cria-se a página HTML adicionando um canvas a ela.

<!DOCTYPE *html*>

<html *lang*="pt\_br">

    <head>

        <meta *charset*="UTF-8" />

        <meta *name*="viewport" *content*="width=device-width, initial-scale=1.0" />

        <link *rel*="stylesheet" *href*="./css/style.css" />

        <script *defer* *src*="./scripts/main.js"></script>

        <title>snekGame</title>

    </head>

    <body>

        <canvas *id*="snake" *width*="512" *height*="512"></canvas>

    </body>

</html>

Agora partindo para o JS, é sempre interessante associar o canvas da página a uma variável para ter um acesso mais fácil a ela; o mesmo é v;adio para o contexto deste.

const canvas = document.getElementById('snake') //associa o canvas à variável canvas

const context = canvas.getContext('2d') //associa o contexto do canvas à variável 'context'

let box = 32 //tamanho do pixel

function renderBG() {

    context.fillStyle = 'lightgreen'

    context.fillRect(0, 0, 16 \* box, 16 \* box)

} //renderiza o background do jogo

renderBG();

Feito isto, cria-se a cobrinha criando uma função análoga a ‘renderBG()’, a função ‘renderSnake()’.

Após isso deve-se pensar em como a cobrinha irá se comportar. A melhor maneirar de trabalhar com ela é pensando nela como uma array, em que elementos são adicionados e apagados. Portanto cria-se um array com estas características, onde cada parte do corpo da cobrinha será representado por um elemento no array e cada elemento terá um valor x e um valor y.

Após isto, cria-se a função para alterar a posição da cobrinha (playgame()) e então uma para controlar a direção para qual esta irá se mover (cd(event)), que escuta o event que é a tecla pressionada (keydown).

E dentro de playgame deve-se inserir algumas regras para que o jogo corra como esperado, como por exemplo o fato de que a cobrinha deve aparecer do outro lado da tela quando cruzar o limite desta.

const canvas = document.getElementById("snake"); //associa o canvas à variável canvas

const context = canvas.getContext("2d"); //associa o contexto do canvas à variável 'context'

let box = 32; //tamanho do pixel

let snake = []; //Array responsável por manter elementos do corpo da cobrinha

let direction = "right"; //Guarda a atual direção de movimento da cobrinha

snake[0] = {

    x: 8 \* box,

    y: 8 \* box,

}; //Primeiro elemento da cobrinha

function renderBG() {

    context.fillStyle = "lightgreen";

    context.fillRect(0, 0, 16 \* box, 16 \* box);

} //renderiza o background do jogo

function renderSnake() {

    for (i = 0; i < snake.length; i++) {

        context.fillStyle = "green";

        context.fillRect(snake[i].x, snake[i].y, box, box);

    }

} //Renderiza a cobrinha

document.addEventListener("keydown", cd); //esta função 'escuta' um evento na página, no caso o pressionar de uma tecla (keydown)

function cd(*event*) {

    if (*event*.keyCode == 37 && direction != "right") direction = "left";

    if (*event*.keyCode == 38 && direction != "down") direction = "up";

    if (*event*.keyCode == 39 && direction != "left") direction = "right";

    if (*event*.keyCode == 40 && direction != "up") direction = "down";

} //muda a direçao da cobrinha

function playGame() {

    if (snake[0].x > 15 \* box && direction == "right") snake[0].x = 0;

    if (snake[0].x < 0 && direction == "left") snake[0].x = 16 \* box;

    if (snake[0].y > 15 \* box && direction == "down") snake[0].y = 0;

    if (snake[0].y < 0 && direction == "up") snake[0].y = 16 \* box;

    renderBG();

    renderSnake();

    let snakeX = snake[0].x;

    let snakeY = snake[0].y;

    if (direction == "right") snakeX += box;

    if (direction == "left") snakeX -= box;

    if (direction == "up") snakeY -= box;

    if (direction == "down") snakeY += box;

    snake.pop();

    let newHead = {

        x: snakeX,

        y: snakeY,

    };

    snake.unshift(newHead);

    console.log("aqui passou");

}

let game = setInterval(playGame, 200);

Após isto cria-se a função que faz a comida aparecer e adiciona-se as condições para que ela cresça quando comer e uma que faça o jogo acabar quando ela morder o próprio corpo.