//Comentarios Anteriorres

/\*

//-------------------------------------------

const canvas = document.querySelector("canvas");

const ctx = canvas.getContext("2d");

const score =document.querySelector(".score--value")

const finalScore = document.querySelector(".final-score > span")

const menu = document.querySelector(".menu-screen")

const buttonPlay = document.querySelector(".btn-play")

const size = 30

const snake = [

{x: 270, y: 240},

{x: 300, y: 240},

{x: 330, y: 240},

{x: 360, y: 240},

{x: 390, y: 240},

{x: 420, y: 240}

]

const incrementScore = () => {

score.innerText = parseInt(score.innerText) + 10

}

const randomNumber = (min, max) =>{

return Math.floor(Math.random() \* (max - min) + min)

}

const randomPosition = () => {

const number = randomNumber(0, canvas.width - size)

return Math.round(number / 30 \* 30)

}

const width = canvas.width = window.innerWidth;

const height = canvas.height = window.innerHeight;

---------------------------------------------------

const canvas = document.querySelector("canvas")

const ctx = canvas.getContext("2d")

const score = document.querySelector(".score--value")

const finalScore = document.querySelector(".final-score > span")

const menu = document.querySelector(".menu-screen")

const buttonPlay = document.querySelector(".btn-play")

const audio = new Audio("../assets/audio.mp3")

const size = 30

const initialPosition = { x: 270, y: 240 }

let snake = [initialPosition]

const incrementScore = () => {

score.innerText = +score.innerText + 10

}

const randomNumber = (min, max) => {

return Math.round(Math.random() \* (max - min) + min)

}

const randomPosition = () => {

const number = randomNumber(0, canvas.width - size)

return Math.round(number / 30) \* 30

}

const randomColor = () => {

const red = randomNumber(0, 255)

const green = randomNumber(0, 255)

const blue = randomNumber(0, 255)

return `rgb(${red}, ${green}, ${blue})`

}

const food = {

x: randomPosition(),

y: randomPosition(),

color: randomColor()

}

let direction, loopId

const drawFood = () => {

const { x, y, color } = food

ctx.shadowColor = color

ctx.shadowBlur = 6

ctx.fillStyle = color

ctx.fillRect(x, y, size, size)

ctx.shadowBlur = 0

}

const drawSnake = () => {

ctx.fillStyle = "#ddd"

snake.forEach((position, index) => {

if (index == snake.length - 1) {

ctx.fillStyle = "white"

}

ctx.fillRect(position.x, position.y, size, size)

})

}

const moveSnake = () => {

if (!direction) return

const head = snake[snake.length - 1]

if (direction == "right") {

snake.push({ x: head.x + size, y: head.y })

}

if (direction == "left") {

snake.push({ x: head.x - size, y: head.y })

}

if (direction == "down") {

snake.push({ x: head.x, y: head.y + size })

}

if (direction == "up") {

snake.push({ x: head.x, y: head.y - size })

}

snake.shift()

}

const drawGrid = () => {

ctx.lineWidth = 1

ctx.strokeStyle = "#191919"

for (let i = 30; i < canvas.width; i += 30) {

ctx.beginPath()

ctx.lineTo(i, 0)

ctx.lineTo(i, 600)

ctx.stroke()

ctx.beginPath()

ctx.lineTo(0, i)

ctx.lineTo(600, i)

ctx.stroke()

}

}

const chackEat = () => {

const head = snake[snake.length - 1]

if (head.x == food.x && head.y == food.y) {

incrementScore()

snake.push(head)

audio.play()

let x = randomPosition()

let y = randomPosition()

while (snake.find((position) => position.x == x && position.y == y)) {

x = randomPosition()

y = randomPosition()

}

food.x = x

food.y = y

food.color = randomColor()

}

}

const checkCollision = () => {

const head = snake[snake.length - 1]

const canvasLimit = canvas.width - size

const neckIndex = snake.length - 2

const wallCollision =

head.x < 0 || head.x > canvasLimit || head.y < 0 || head.y > canvasLimit

const selfCollision = snake.find((position, index) => {

return index < neckIndex && position.x == head.x && position.y == head.y

})

if (wallCollision || selfCollision) {

gameOver()

}

}

const gameOver = () => {

direction = undefined

menu.style.display = "flex"

finalScore.innerText = score.innerText

canvas.style.filter = "blur(2px)"

}

const gameLoop = () => {

clearInterval(loopId)

ctx.clearRect(0, 0, 600, 600)

drawGrid()

drawFood()

moveSnake()

drawSnake()

chackEat()

checkCollision()

loopId = setTimeout(() => {

gameLoop()

}, 300)

}

gameLoop()

document.addEventListener("keydown", ({ key }) => {

if (key == "ArrowRight" && direction != "left") {

direction = "right"

}

if (key == "ArrowLeft" && direction != "right") {

direction = "left"

}

if (key == "ArrowDown" && direction != "up") {

direction = "down"

}

if (key == "ArrowUp" && direction != "down") {

direction = "up"

}

})

buttonPlay.addEventListener("click", () => {

score.innerText = "00"

menu.style.display = "none"

canvas.style.filter = "none"

snake = [initialPosition]

})

//Criando a maça na tela, definindo posição em canvas

\*ctx.fillStyle = "#FF7F00"

\*ctx.fillRect(250, 25, 20, 20)

\*/