const canvas = document.getElementById("game");

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

class SnakePart {

constructor(x, y) {

this.x = x;

this.y = y;

}

}

let speed = 7;

let tileCount = 20;

let tileSize = canvas.width / tileCount - 2;

let headX = 10;

let headY = 10;

const snakeParts = [];

let tailLength = 2;

let appleX = 5;

let appleY = 5;

let inputsXVelocity = 0;

let inputsYVelocity = 0;

let xVelocity = 0;

let yVelocity = 0;

let score = 0;

const gulpSound = new Audio("gulp.mp3");

let previousXVelocity = 0;

let previousYVelocity = 0;

//game loop

function drawGame() {

xVelocity = inputsXVelocity;

yVelocity = inputsYVelocity;

//Was moving right and try to move left

if (previousXVelocity === 1 && xVelocity === -1) {

xVelocity = previousXVelocity;

}

//Was moving left and try to move right

if (previousXVelocity === -1 && xVelocity === 1) {

xVelocity = previousXVelocity;

}

//Was moving up and try to move down

if (previousYVelocity === -1 && yVelocity === 1) {

yVelocity = previousYVelocity;

}

//Was moving down and try to move up

if (previousYVelocity === 1 && yVelocity === -1) {

yVelocity = previousYVelocity;

}

previousXVelocity = xVelocity;

previousYVelocity = yVelocity;

changeSnakePosition();

let result = isGameOver();

if (result) {

document.body.removeEventListener("keydown", keyDown);

return;

}

clearScreen();

checkAppleCollision();

drawApple();

drawSnake();

drawScore();

if (score > 5) {

speed = 9;

}

if (score > 10) {

speed = 11;

}

setTimeout(drawGame, 1000 / speed);

}

function isGameOver() {

let gameOver = false;

if (yVelocity === 0 && xVelocity === 0) {

return false;

}

//walls

if (headX < 0) {

gameOver = true;

} else if (headX === tileCount) {

gameOver = true;

} else if (headY < 0) {

gameOver = true;

} else if (headY === tileCount) {

gameOver = true;

}

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

let part = snakeParts[i];

if (part.x === headX && part.y === headY) {

gameOver = true;

break;

}

}

if (gameOver) {

ctx.fillStyle = "white";

ctx.font = "50px Verdana";

if (gameOver) {

ctx.fillStyle = "white";

ctx.font = "50px Verdana";

var gradient = ctx.createLinearGradient(0, 0, canvas.width, 0);

gradient.addColorStop("0", " magenta");

gradient.addColorStop("0.5", "blue");

gradient.addColorStop("1.0", "red");

// Fill with gradient

ctx.fillStyle = gradient;

ctx.fillText("Game Over!", canvas.width / 6.5, canvas.height / 2);

}

ctx.fillText("Game Over!", canvas.width / 6.5, canvas.height / 2);

}

return gameOver;

}

function drawScore() {

ctx.fillStyle = "white";

ctx.font = "10px Verdana";

ctx.fillText("Score " + score, canvas.width - 50, 10);

}

function clearScreen() {

ctx.fillStyle = "black";

ctx.fillRect(0, 0, canvas.width, canvas.height);

}

function drawSnake() {

ctx.fillStyle = "green";

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

let part = snakeParts[i];

ctx.fillRect(part.x \* tileCount, part.y \* tileCount, tileSize, tileSize);

}

snakeParts.push(new SnakePart(headX, headY)); //put an item at the end of the list next to the head

while (snakeParts.length > tailLength) {

snakeParts.shift(); // remove the furthest item from the snake parts if have more than our tail size.

}

ctx.fillStyle = "orange";

ctx.fillRect(headX \* tileCount, headY \* tileCount, tileSize, tileSize);

}

function changeSnakePosition() {

headX = headX + xVelocity;

headY = headY + yVelocity;

}

function drawApple() {

ctx.fillStyle = "red";

ctx.fillRect(appleX \* tileCount, appleY \* tileCount, tileSize, tileSize);

}

function checkAppleCollision() {

if (appleX === headX && appleY == headY) {

appleX = Math.floor(Math.random() \* tileCount);

appleY = Math.floor(Math.random() \* tileCount);

tailLength++;

score++;

gulpSound.play();

}

}

document.body.addEventListener("keydown", keyDown);

function keyDown(event) {

console.log(inputsXVelocity, inputsYVelocity);

//up

if (event.keyCode == 38 || event.keyCode == 87) {

//87 is w

inputsYVelocity = -1;

inputsXVelocity = 0;

}

//down

if (event.keyCode == 40 || event.keyCode == 83) {

// 83 is s

inputsYVelocity = 1;

inputsXVelocity = 0;

}

//left

if (event.keyCode == 37 || event.keyCode == 65) {

// 65 is a

inputsYVelocity = 0;

inputsXVelocity = -1;

}

//right

if (event.keyCode == 39 || event.keyCode == 68) {

//68 is d

inputsYVelocity = 0;

inputsXVelocity = 1;

}

}

drawGame();