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

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

const startButton = document.getElementById("startButton");

const pauseButton = document.getElementById("pauseButton");

const endButton = document.getElementById("endButton");

const tileSize = 20;

const rows = 15;

const cols = 15;

let jogoRodando = false;

let pausado = false;

let lastTime = 0;

const map = [

  [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],

  [0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 0],

  [0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0],

  [0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0],

  [0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0],

  [0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 0],

  [0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0],

  [0, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0],

  [0, 1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 0],

  [0, 1, 1, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 1, 0],

  [0, 0, 0, 1, 0, 1, 1, 1, 1, 1, 0, 0, 1, 0, 0],

  [0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 0, 1, 1, 0],

  [0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 0, 0, 1, 0],

  [0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0],

  [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],

];

let pacman = { x: 1, y: 1, dx: 0, dy: 0, direction: "right" };

let predador = { x: 13, y: 13, dx: 0, dy: -1, direction: "up" };

let moedas = [];

function criarMoedas() {

  moedas = [];

  for (let y = 0; y < rows; y++) {

    for (let x = 0; x < cols; x++) {

      if (map[y][x] === 1) {

        moedas.push({ x, y });

      }

    }

  }

}

function updatePredador() {

  const nextX = predador.x + predador.dx;

  const nextY = predador.y + predador.dy;

  if (

    nextX < 0 ||

    nextX >= cols ||

    nextY < 0 ||

    nextY >= rows ||

    map[nextY][nextX] === 0

  ) {

    mudarDirecaoPredador();

  } else {

    predador.x = nextX;

    predador.y = nextY;

  }

  if (predador.x === pacman.x && predador.y === pacman.y) {

    jogoRodando = false;

    alert("O predador capturou o Pac-Man! Fim de jogo!");

  }

}

function mudarDirecaoPredador() {

  const direcoes = [

    { dx: 0, dy: -1, direction: "up" },

    { dx: 0, dy: 1, direction: "down" },

    { dx: -1, dy: 0, direction: "left" },

    { dx: 1, dy: 0, direction: "right" },

  ];

  const oposta = getOppositeDirection(predador.direction);

  const filtradas = direcoes.filter((d) => d.direction !== oposta);

  for (let d of filtradas) {

    const nx = predador.x + d.dx;

    const ny = predador.y + d.dy;

    if (nx >= 0 && nx < cols && ny >= 0 && ny < rows && map[ny][nx] === 1) {

      predador.dx = d.dx;

      predador.dy = d.dy;

      predador.direction = d.direction;

      return;

    }

  }

  // Aleatória

  for (let i = 0; i < 10; i++) {

    const d = direcoes[Math.floor(Math.random() \* direcoes.length)];

    const nx = predador.x + d.dx;

    const ny = predador.y + d.dy;

    if (nx >= 0 && nx < cols && ny >= 0 && ny < rows && map[ny][nx] === 1) {

      predador.dx = d.dx;

      predador.dy = d.dy;

      predador.direction = d.direction;

      break;

    }

  }

}

function getOppositeDirection(direction) {

  switch (direction) {

    case "up":

      return "down";

    case "down":

      return "up";

    case "left":

      return "right";

    case "right":

      return "left";

    default:

      return direction;

  }

}

function update() {

  const nx = pacman.x + pacman.dx;

  const ny = pacman.y + pacman.dy;

  if (nx >= 0 && nx < cols && ny >= 0 && ny < rows && map[ny][nx] === 1) {

    pacman.x = nx;

    pacman.y = ny;

    moedas = moedas.filter((m) => m.x !== pacman.x || m.y !== pacman.y);

  }

  updatePredador();

}

function draw() {

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

  drawMap();

  drawMoedas();

  drawPacman();

  drawPredador();

}

function drawMap() {

  for (let y = 0; y < rows; y++) {

    for (let x = 0; x < cols; x++) {

      ctx.fillStyle = map[y][x] === 0 ? "blue" : "black";

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

    }

  }

}

function drawMoedas() {

  ctx.fillStyle = "yellow";

  moedas.forEach((m) => {

    ctx.beginPath();

    ctx.arc(

      m.x \* tileSize + tileSize / 2,

      m.y \* tileSize + tileSize / 2,

      5,

      0,

      Math.PI \* 2

    );

    ctx.fill();

  });

}

function drawPacman() {

  ctx.fillStyle = "yellow";

  ctx.beginPath();

  ctx.arc(

    pacman.x \* tileSize + tileSize / 2,

    pacman.y \* tileSize + tileSize / 2,

    tileSize / 2 - 2,

    0,

    Math.PI \* 2

  );

  ctx.fill();

}

function drawPredador() {

  ctx.fillStyle = "red";

  ctx.beginPath();

  ctx.arc(

    predador.x \* tileSize + tileSize / 2,

    predador.y \* tileSize + tileSize / 2,

    tileSize / 2 - 2,

    0,

    Math.PI \* 2

  );

  ctx.fill();

}

function gameLoop(timestamp) {

  if (!jogoRodando || pausado) return;

  const deltaTime = timestamp - lastTime;

  if (deltaTime > 150) {

    update();

    draw();

    lastTime = timestamp;

  }

  requestAnimationFrame(gameLoop);

}

startButton.onclick = () => {

  if (!jogoRodando) {

    pacman = { x: 1, y: 1, dx: 0, dy: 0, direction: "right" };

    predador = { x: 13, y: 13, dx: 0, dy: -1, direction: "up" };

    criarMoedas();

    jogoRodando = true;

    pausado = false;

    requestAnimationFrame(gameLoop);

  } else {

    pausado = false;

    requestAnimationFrame(gameLoop);

  }

};

pauseButton.onclick = () => {

  pausado = !pausado;

  if (!pausado) requestAnimationFrame(gameLoop);

};

endButton.onclick = () => {

  jogoRodando = false;

  pausado = false;

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

};

document.addEventListener("keydown", (e) => {

  switch (e.key) {

    case "ArrowUp":

      pacman.dx = 0;

      pacman.dy = -1;

      pacman.direction = "up";

      break;

    case "ArrowDown":

      pacman.dx = 0;

      pacman.dy = 1;

      pacman.direction = "down";

      break;

    case "ArrowLeft":

      pacman.dx = -1;

      pacman.dy = 0;

      pacman.direction = "left";

      break;

    case "ArrowRight":

      pacman.dx = 1;

      pacman.dy = 0;

      pacman.direction = "right";

      break;

  }

});