<!DOCTYPE html>

<html>

<head>

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

<style>

canvas {

border:1px solid #d3d3d3;

background-color: #f1f1f1;

}

</style>

</head>

<body onload="startGame()">

<script>

var myGamePiece;

var myObstacles = [];

var myScore;

function startGame() {

myGamePiece = new component(30, 30, "red", 10, 120);

myScore = new component("30px", "Consolas", "black", 280, 40, "text");

myGameArea.start();

}

var myGameArea = {

canvas : document.createElement("canvas"),

start : function() {

this.canvas.width = 480;

this.canvas.height = 270;

this.context = this.canvas.getContext("2d");

document.body.insertBefore(this.canvas, document.body.childNodes[0]);

this.frameNo = 0;

this.interval = setInterval(updateGameArea, 20);

},

clear : function() {

this.context.clearRect(0, 0, this.canvas.width, this.canvas.height);

},

stop : function() {

clearInterval(this.interval);

}

}

function component(width, height, color, x, y, type) {

this.type = type;

this.width = width;

this.height = height;

this.speedX = 0;

this.speedY = 0;

this.x = x;

this.y = y;

this.update = function() {

ctx = myGameArea.context;

if (this.type == "text") {

ctx.font = this.width + " " + this.height;

ctx.fillStyle = color;

ctx.fillText(this.text, this.x, this.y);

} else {

ctx.fillStyle = color;

ctx.fillRect(this.x, this.y, this.width, this.height);

}

}

this.newPos = function() {

this.x += this.speedX;

this.y += this.speedY;

}

this.crashWith = function(otherobj) {

var myleft = this.x;

var myright = this.x + (this.width);

var mytop = this.y;

var mybottom = this.y + (this.height);

var otherleft = otherobj.x;

var otherright = otherobj.x + (otherobj.width);

var othertop = otherobj.y;

var otherbottom = otherobj.y + (otherobj.height);

var crash = true;

if ((mybottom < othertop) || (mytop > otherbottom) || (myright < otherleft) || (myleft > otherright)) {

crash = false;

}

return crash;

}

}

function updateGameArea() {

var x, height, gap, minHeight, maxHeight, minGap, maxGap;

for (i = 0; i < myObstacles.length; i += 1) {

if (myGamePiece.crashWith(myObstacles[i])) {

myGameArea.stop();

return;

}

}

myGameArea.clear();

myGameArea.frameNo += 1;

if (myGameArea.frameNo == 1 || everyinterval(150)) {

y = myGameArea.canvas.width;

minWidth = 20;

maxWidth = 200;

width = Math.floor(Math.random()\*(maxWidth-minWidth+1)+minWidth);

minGap = 50;

maxGap = 200;

gap = Math.floor(Math.random()\*(maxGap-minGap+1)+minGap);

myObstacles.push(new component(width, 10, "green", 0, 0));

myObstacles.push(new component(y - width - gap, 10, "green", width + gap, 0))

}

for (i = 0; i < myObstacles.length; i += 1) {

myObstacles[i].speedY = 1;

myObstacles[i].newPos();

myObstacles[i].update();

}

myScore.text="SCORE: " + myGameArea.frameNo;

myScore.update();

myGamePiece.newPos();

myGamePiece.update();

}

function everyinterval(n) {

if ((myGameArea.frameNo / n) % 1 == 0) {return true;}

return false;

}

</script>

<p>The score will count one point for each frame you manage to "stay alive".</p>

</body>

</html>