

```

                                game

// Sources voor uitleg en code:
//
https://developer.mozilla.org/en-US/docs/Web/API/CanvasRenderingContext2D/drawImage
// http://keycode.info/
//
https://www.w3schools.com/graphics/tryit.asp?filename=trygame_default_gravity

//audio en sprites
// https://www.youtube.com/watch?v=EyGzKxb2zSg
// http://bestanimations.com/Fantasy/Dragons/Dragons2.html
// https://killishandra.deviantart.com/art/Golden-pern-dragon-sprite-252632979
// http://pixeljoint.com/forum/forum_posts.asp?TID=15971&PD=0

// DISCLAIMER spel werkt maar is niet af: gebruik de pijl toetsen om te bewegen

//variabele
var player;
var enemies = [];
var score;
var ambience;
var fireBall;

function startGame() {
    //objecten met de bijbehorende parameters
    player = new component(180, 180, "./sprites/dragonSprite1.gif", 10, 120,
        "image");
    fireBall = new component(60, 60, "./sprites/fireballSprite.gif", 10, 120,
        "image");
    background = new component(1600, 800, "./sprites/background.png", 0, 0,
"image");
    score = new component("30px", "Consolas", "black", 10, 40, "text")
    ambience = new sound("./audio/caveTune.mp3");
    //voert start uit
    myGameArea.start();
}

var myGameArea = {
    background: document.getElementById("gameBackground"),
    //de methode start geeft het canvas lengte en de breedte,
    //het canvas wordt het eerste childnode van het body element
    start: function() {
        this.background.width = 1600;
        this.background.height = 800;
        this.context = this.background.getContext("2d");
        ambience.play();
        document.body.insertBefore(this.background, document.body.childNodes[
            0]);
        this.frameNo = 0;
        this.interval = setInterval(updateGameArea, 20);
    },

```

```

                                game
clear: function() {
    this.context.clearRect(0, 0, this.background.width, this.background
        .height);
},
stop: function() {
    clearInterval(this.interval);
    ambience.stop();
    alert("Game over");
    myGameArea.start();
}
}

```

```

function sound(src) {
    this.sound = document.createElement("audio");
    this.sound.src = src;
    this.sound.setAttribute("preload", "auto");
    this.sound.setAttribute("controls", "none");
    this.sound.style.display = "none";
    document.body.appendChild(this.sound);
    this.play = function() {
        this.sound.play();
    }
    this.stop = function() {
        this.sound.pause();
    }
}

```

```

//constructor bevat eigenschappen en parameters die de
//component beheersen, namelijk width height x y en type
function component(width, height, color, x, y, type) {
    this.type = type;
    //als de component van het type image is dan wordt de
    //kleur de source
    if (type == "image") {
        this.image = new Image();
        this.image.src = color;
    }
    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 (type == "image") {
            ctx.drawImage(this.image,
                this.x,
                this.y,
                this.width, this.height);
        } else if (this.type == "text") {
            ctx.font = this.width + " " + this.height;
            ctx.fillStyle = color;

```

```

        game
        ctx.fillText(this.text, this.x, this.y);
    } else {
        ctx.fillStyle = color;
        ctx.fillRect(this.x, this.y, this.width, this.height);
    }
}
//zorgt ervoor dat de player beweegt
this.newPos = function() {
    this.x += this.speedX;
    this.y += this.speedY;
}
//zorgt ervoor dat wanneer de speler in contact raakt
// met de vijandige draak het spel stopt
//dit wordt gedaan door te kijken naar de x en de y waarde
//van de componenten
this.crashWith = function(enemieColision) {
    var playerleft = this.x;
    var playerright = this.x + (this.width);
    var playertop = this.y;
    var playerbottom = this.y + (this.height);
    var enemieleft = enemieColision.x;
    var enemieright = enemieColision.x + (enemieColision.width);
    var enemietop = enemieColision.y;
    var enemiebottom = enemieColision.y + (enemieColision.height);
    var crash = true;
    if ((playerbottom < enemietop) || (playertop > enemiebottom) ||
    (playerright <
        enemieleft) || (playerleft >
        enemieright)) {
        crash = false;
    }
    return crash;
}
}

```

```

//spawned de components
// elke 10 seconden worden
//er nieuwe vijanden gespawned
function updateGameArea() {
    var x, y;
    for (i = 0; i < enemies.length; i += 1) {
        if (player.crashWith(enemies[i])) {
            myGameArea.stop();
            return;
        }
    }
    myGameArea.clear();
    background.newPos();
    background.update();
    myGameArea.frameNo += 1;
}

```

```

                                game
if (myGameArea.frameNo == 1 || everyinterval(500)) {
    x = myGameArea.background.width;
    y = myGameArea.background.height - 200
    enemies.push(new component(180, 180, "./sprites/enemyDragon1.gif",
1500, 10,
        "image"));
    enemies.push(new component(180, 180, "./sprites/enemyDragon1.gif",
1600, 160,
        "image"));
    enemies.push(new component(180, 180, "./sprites/enemyDragon2.gif",
2000, 100,
        "image"));
    enemies.push(new component(180, 180, "./sprites/enemyDragon1.gif",
2100, 240,
        "image"));
    enemies.push(new component(180, 180, "./sprites/enemyDragon2.gif",
2500, 500,
        "image"));
    enemies.push(new component(180, 180, "./sprites/enemyDragon1.gif",
1600, 600,
        "image"));
}
for (i = 0; i < enemies.length; i += 1) {

    enemies[i].x += -3;
    enemies[i].newPos();
    enemies[i].update();
    player.newPos();
    player.update();
    fireBall.newPos();
    fireBall.update();
    score.text = "SCORE: " + myGameArea.frameNo;
    score.update();
}

function everyinterval(n) {
    if ((myGameArea.frameNo / n) % 1 == 0) {
        return true;
    }
    return false;
}
//eventlistener voor wanneer pijltjes worden ingedrukt
window.addEventListener("keydown", checkKeyPress, false);
if (checkKeyPress.keyCode == "32") {
    function fireBallBehaviour() {
        fireBall.x += 3;
        fireBall.update();
    }
}
//als de pijltjes toetsen worden ingedrukt dan beweegt het draakje
function checkKeyPress(key) {
    player.image.src = "./sprites/dragonsSprite2.gif";

```

```

                                game
if (key.keyCode == "38") {
    player.speedY = -3;
} else if (key.keyCode == "40") {
    player.speedY = 3;
} else if (key.keyCode == "37") {
    player.speedX = -3;
} else if (key.keyCode == "32") {
    fireBall.x += 3;
} else if (key.keyCode == "39") {
    player.speedX = 3;
} else if (key.keyCode == "80") {
    alert("Paused");
    ambience.stop();
}
//eventlistener voor wanneer pijltjes worden losgelaten
window.addEventListener("keyup", checkKeyRelease, false);
//zorgt ervoor dat als je de pijltjes loslaat de draak niet
//verder gaat, sprite verandert.
function checkKeyRelease(key) {
    player.image.src = "./sprites/dragonSprite1.gif";
    if (key.keyCode == "38") {
        player.speedX = 0;
        player.speedY = 0;
    } else if (key.keyCode == "40") {
        player.speedX = 0;
        player.speedY = 0;
    } else if (key.keyCode == "37") {
        player.speedX = 0;
        player.speedY = 0;
    } else if (key.keyCode == "39") {
        player.speedX = 0;
        player.speedY = 0;
    } else if (key.keyCode == "32") {
        fireBall.x += 3;
    }
}
}
}

```