/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

//Definimos las variables para la partida:

var imagenX=10; //X del jugador

var imagenY=50; //Y del jugador

var rightArrowPressed = false, leftArrowPressed = false, upArrowPressed = false, downArrowPressed = false; //booleanos para las teclas

var array=[]; //array de enemigos

var gest=0; //Variable auxiliar de "gestion\_enemigos"

var vidas=3; //Vidas del jugador.

var dificultad; //Variable de dificultad.

var limite; //Variable auxiliar de gestion\_enemigos

//Y acá cargamos las imágenes relativas tanto a los enemigos como al prota.

var img= new Image;

img.src = "img/pe.png"; //Imagen del personaje jugable.

var foto= new Image; //Imagen del cactus desde abajo.

foto.src="img/Imagen1.png";

var cactus= new Image;

cactus.src="img/Imagen5.png";

var yunque= new Image; //Imagen del yunque.

yunque.src="img/Imagen4.png";

var bola= new Image; //Imagen de la bola.

bola.src="img/Imagen2.png";

var final\_screen= new Image; //Imagen del fondo de los menús.

final\_screen.src="img/fin.jpg";

//Otras variables de carácter auxiliar:

var timeout;

var minutos=0;

var segundos=0;

var theChrono;

var puntos=0;

var thePuntos;

var theVidas;

var conten=false;

var auz;

var parar\_tiempo=false;

var play=false;

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

function comienzo(){ //Variable de inicio de los elementos básicos del juego.

window.document.addEventListener("keydown", handlerOne);

window.document.addEventListener("keyup", handlerTwo);

canvas=document.getElementById('Canvas');

canvas.style.background='#F5F5F5';

ctx = canvas.getContext('2d');

theChrono = document.getElementById("chrono");

thePuntos=document.getElementById("puntuacion");

theVidas=document.getElementById("nvidas");

timeout = window.setTimeout(temporizador, 1000);

auz=setInterval(movimiento, 1000/60);

}

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

function movimiento(){ //Actúa como update

if(play===false){

menu();

segundos=-1;

puntos=-50;

eleccion\_menu();

}

else{

desplazar();

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

ctx.drawImage(img, imagenX, imagenY,120,80);

gestion\_enemigos();

for(var a=0;a<array.length; a++){

array[a].render(ctx);

if (array[a].x + array[a].w <= 0) {

delete(array[a]);

array.splice(a, 1);

}

if (array[a].crashWith(imagenX,imagenY,50,50)){

conten=true;

}

}

if (vidas===0)

end();

}};

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

var handlerOne = function (event) { //Función del evento de pulsar botones.

switch (event.keyCode) {

case 39:

if (!rightArrowPressed) {

rightArrowPressed = true;

}

break;

case 37:

if (!leftArrowPressed) {

leftArrowPressed = true;

}

break;

case 38:

if (!upArrowPressed) {

upArrowPressed = true;

}

break;

case 40:

if (!downArrowPressed) {

downArrowPressed = true;

}

break;

default:

break;

}

};

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

var handlerTwo = function (event) { //Función del evento de soltar botones.

switch (event.keyCode) {

case 39:

rightArrowPressed = false;

break;

case 37:

leftArrowPressed = false;

break;

case 38:

upArrowPressed = false;

break;

case 40:

downArrowPressed = false;

break;

default:

break;

}

};

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

function desplazar(){ //Función de desplazamiento del jugador

if (rightArrowPressed && imagenX<=1590)

imagenX+=8;

if (leftArrowPressed && imagenX>=0)

imagenX-=8;

if (upArrowPressed && imagenY>=0)

imagenY-=8;

if (downArrowPressed && imagenY<=390)

imagenY+=8;

};

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

function cactuses (posicion,imagen){ //Las siguientes funciones actúan como clases de enemigos

this.w=148;

this.h=255;

this.x=1800;

this.y;

if (posicion)

this.y=250;

else

this.y=0;

this.render = function (ctx) {

this.x-=dificultad;

ctx.drawImage(imagen,this.x, this.y, this.w, this.h);

};

this.crashWith = function (x,y,width,height) {

// detect collision with the bounding box algorithm

var myleft = this.x;

var myright = this.x + this.w-20;

var mytop = this.y-20;

var mybottom = this.y + this.h-20;

var otherleft = x+25;

var otherright = x + width+20;

var othertop = y+30;

var otherbottom = y + height-20;

var crash = true;

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

(myleft > otherright)) {

crash = false;

}

return crash;

};

};

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

function balas (imagen){

this.w=130;

this.h=100;

this.x=1800;

this.y=8;

this.pos;

this.render = function (ctx) {

this.x-=dificultad;

if (this.y === 400)

this.pos=true;

else if (this.y === 8)

this.pos=false;

if (this.pos)

this.y-=8;

else if (this.pos===false)

this.y+=8;

ctx.drawImage(imagen,this.x, this.y, this.w, this.h);

};

this.crashWith = function (x,y,width,height) {

// detect collision with the bounding box algorithm

var myleft = this.x;

var myright = this.x + this.w-20;

var mytop = this.y-20;

var mybottom = this.y + this.h-20;

var otherleft = x+25;

var otherright = x + width+20;

var othertop = y+30;

var otherbottom = y + height-20;

var crash = true;

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

(myleft > otherright)) {

crash = false;

}

return crash;

};

};

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

function yunques (imagen){

this.w=155;

this.h=130;

this.x=1800;

this.y=0;

this.posi=false;

this.render = function (ctx) {

this.x-=dificultad;

if(this.x<=imagenX+40 && this.x>=imagenX-40)

this.posi=true;

if (this.y<370 && this.posi)

this.y+=12;

ctx.drawImage(imagen,this.x, this.y, this.w, this.h);

};

this.crashWith = function (x,y,width,height) {

// detect collision with the bounding box algorithm

var myleft = this.x;

var myright = this.x + this.w-20;

var mytop = this.y-20;

var mybottom = this.y + this.h-20;

var otherleft = x+25;

var otherright = x + width+20;

var othertop = y+30;

var otherbottom = y + height-20;

var crash = true;

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

(myleft > otherright)) {

crash = false;

}

return crash;

};

};

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

function generarEnemigos(){ //Función de generación de enemigos.

var i=Math.floor(Math.random() \*10);

if (i===0 || i===5 || i===7){

array.push(new cactuses(true,foto));}

if (i===1 || i===6 || i===8 || i===10){

array.push(new cactuses(false,cactus));}

if (i===2|| i===9){

array.push(new balas(bola));}

if (i===3){

array.push(new yunques(yunque));}

};

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

function gestion\_enemigos(){ //Todo lo relativo a la gestión de los mismos se halla aquí.

gest+=1;

if (gest>=limite)

gest=0;

if (gest===0){

var aleat=Math.random();

if(aleat<=0.9)

generarEnemigos();

}

};

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

function temporizador(){ //Manejo del cronómetro

if(parar\_tiempo===false){

segundos++;

if (segundos%3===0)

puntos+=50;

if (segundos > 59) {

minutos++;

segundos = 0;

}

if (conten){

conten=false;

vidas-=1;

}

theChrono.innerHTML = "Tiempo de juego: " + pad(minutos, 2) + ":" + pad(segundos, 2);

thePuntos.innerHTML = "Puntos: "+puntos;

theVidas.innerHTML = "Vidas: "+vidas;

timeout = window.setTimeout(temporizador, 1000);}

};

function pad(n, width, z) { //Con esta función podemos escribir los minutos y segundos por pantalla.

z = z || "0";

var s = n.toString();

return s.length >= width ? s : new Array(width - s.length + 1).join(z) + s;

};

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

function end() //Manejo de la pantalla de cierre.

{

parar\_tiempo=true;

ctx.fillStyle="black";

array = [];

clearInterval(auz);

window.document.removeEventListener("keydown", handlerOne);

window.document.removeEventListener("keyup", handlerTwo);

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

ctx.drawImage(final\_screen, canvas.height, 0,canvas.height+300,canvas.height);

ctx.font="bold 70px arial";

ctx.strokeStyle="red"; //color externo

ctx.fillStyle="yellow";

ctx.strokeText("FINAL DE LA PARTIDA",canvas.height+20,140);

ctx.fillText("FINAL DE LA PARTIDA",canvas.height+20,140);

ctx.font="bold 50px arial";

ctx.strokeStyle="yellow"; //color externo

ctx.fillStyle="red";

ctx.strokeText("PUNTUACIÓN: ",canvas.height+100,240);

ctx.fillText("PUNTUACIÓN: ",canvas.height+100,240);

ctx.strokeText("TIEMPO: ",canvas.height+100,300);

ctx.fillText("TIEMPO: ",canvas.height+100,300);

ctx.strokeStyle="yellow"; //color externo

ctx.fillStyle="blue";

ctx.strokeText(puntos,canvas.height+460,240);

ctx.fillText(puntos,canvas.height+460,240);

ctx.strokeText(minutos+" min "+segundos+" seg",canvas.height+330,300);

ctx.fillText(minutos+" min "+segundos+" seg",canvas.height+330,300);

}

function menu() //Manejo de la pantalla de inicio.

{

ctx.fillStyle="black";

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

ctx.drawImage(final\_screen, canvas.height, 0,canvas.height+300,canvas.height);

ctx.font="bold 40px arial";

ctx.strokeStyle="blue"; //color externo

ctx.fillStyle="yellow";

ctx.strokeText("Aythami Godoy Rodríguez, 46250752Y",canvas.height+50,140);

ctx.fillText("Aythami Godoy Rodríguez, 46250752Y",canvas.height+50,140);

ctx.font="bold 30px Times New Roman";

ctx.strokeStyle="blue"; //color externo

ctx.fillStyle="red";

ctx.strokeText("Fácil: pulsa flecha arriba ",canvas.height+250,230);

ctx.fillText("Fácil: pulsa flecha arriba ",canvas.height+250,230);

ctx.strokeText("Normal: pulsa flecha abajo ",canvas.height+250,300);

ctx.fillText("Normal: pulsa flecha abajo ",canvas.height+250,300);

ctx.strokeText("Difícil: pulsa flecha derecha ",canvas.height+250,370);

ctx.fillText("Difícil: pulsa flecha derecha ",canvas.height+250,370);

};

function eleccion\_menu(){ //Aprovechando el handler\_one, gastamos los booleanos para configurar la dificultad.

if (rightArrowPressed){

dificultad=24;

limite=25;

play=true;}

if (upArrowPressed){

dificultad=8;

limite=55;

play=true;}

if (downArrowPressed){

dificultad=16;

limite=35;

play=true;}

};