/\*\*ARGUINO + PROCESSING \*\*/

/\*\*PROGRAMA QUE MUEVE EL RECTANGULO CON POTENCIOMETRO\*\*/

import processing.serial.\*;

Serial port;

int leer; //recibe valor entre 0 y 500 del potenciometro en el arduino

int anchoVentana=350; //en size no identifica esta variable, toca cambiarla manual

int altoVentana=500;

int puntaje=0;

int vidas=3;

String mje;

int tiempoLimite=15000;

color nudePiel = #B3A48D;

color negroCasi = color(30, 28, 34);

//Cuadros

float posCuadro\_X=200;

int espacioEntreCuadros= 8;

int columnasCuadros= 2; //columnas

int filasCuadros= 2; // Filas

int espacioCielo= 100; //espacio entre borde sup ventana y 1ra fila de cuadros

float anchoCuadro= 20;//(anchoVentana-(columnasCuadros-2)\*espacioEntreCuadros)/columnasCuadros

float altoCuadro= 20;

color brickColors[]= {color(#4C5A56), color(#EAD7C9), color(255,255,0), color(0,255,0), color(0,0,255), color(156,0,255), color(255, 0,0), color(255,183, 0), color(255,255,0), color(0,255,0)};

//color brickColor= color(255, 255, 0); no se q es

ArrayList<Bloque> grupoDeCuadros= new ArrayList<Bloque>();

//Bola

int anchoBola= 16;

float posInicialBolaX=random(anchoVentana);

float posInicialBolaY=altoVentana/2;

color colorBola= color(#000000);

boolean haPerdido= false;

boolean haGanado= false;

Bola Lala= new Bola(posInicialBolaX, posInicialBolaY, anchoBola, colorBola);

//base

int baseX= 2; //ESTABA anchoVentana/2

int baseY= altoVentana-100;

int alturaBase= 12;

int anchoBase= 120;

color baseColor= color(255,255,255,0); //el 0 es de opacidad, representa 0% de opacidad

//255 es un 100% de opacidad

Bloque base= new Bloque(baseX, baseY, anchoBase,alturaBase, baseColor);

//La imagen q sera la base, el cigarrillo

PImage cigarrilloBase;

PFont fuente;

void setup() {

size(350, 500); //deben coincidir con los dos primeros valores de arriba

//saber cuantos puertos estan habilitados y saber elegir el q se trabajara con arduino

println(Serial.list());

/\*\*CONEXION PROCESSING ARDUINO\*\*/

//Serial.list()[2] corresponde al COM8

//COM8 es el puerto al q esta conectado el arduino

//se puede ver en arduino - herramientas - puerto

port= new Serial(this,Serial.list()[2],9600);

background(255);

cigarrilloBase= loadImage("cigarrillo.png");

fuente= createFont("Comfortaa-Regular.ttf",5);

iniciarCuadros();

}

void draw() {

background(#F04141);

dibujarCuadros();

if(!haPerdido&&!haGanado) dibujarBola();

dibujarBase();

updatePuntaje(false);

impTxtVidas();

int tiempoActual = millis();

if(vidas==0){

finalPerder();}else if(tiempoActual>tiempoLimite){

finalPerderPorTiempo();

};

}

//INICIALIZAR CUADROS

void iniciarCuadros() {

//1ra Columna

for(int i=0; i<3; i++){

color brickColor=negroCasi;

float posCuadro\_Y=espacioCielo+(altoCuadro+espacioEntreCuadros)\*i;

posCuadro\_X=5+(anchoCuadro+espacioEntreCuadros);

grupoDeCuadros.add(new Bloque(posCuadro\_X, posCuadro\_Y, anchoCuadro, altoCuadro, brickColor));

}

//2da Columna

for(int i=0; i<5; i++){

color brickColor=negroCasi;

float posCuadro\_Y=(espacioCielo+(altoCuadro+espacioEntreCuadros)\*i)-(altoCuadro+espacioEntreCuadros);

posCuadro\_X=5+anchoCuadro+(espacioEntreCuadros\*5);

grupoDeCuadros.add(new Bloque(posCuadro\_X, posCuadro\_Y, anchoCuadro, altoCuadro, brickColor));

}

//3ra Columna

for(int i=0; i<6; i++){

color brickColor=negroCasi;

float posCuadro\_Y=(espacioCielo+(altoCuadro+espacioEntreCuadros)\*i)-(altoCuadro+espacioEntreCuadros);

posCuadro\_X=5+anchoCuadro+(espacioEntreCuadros\*9);

grupoDeCuadros.add(new Bloque(posCuadro\_X, posCuadro\_Y, anchoCuadro, altoCuadro, brickColor));

}

//4ta Columna

for(int i=0; i<7; i++){

color brickColor=negroCasi;

float posCuadro\_Y=(espacioCielo+(altoCuadro+espacioEntreCuadros)\*i)-(altoCuadro+espacioEntreCuadros);

posCuadro\_X=5+anchoCuadro+(espacioEntreCuadros\*13);

grupoDeCuadros.add(new Bloque(posCuadro\_X, posCuadro\_Y, anchoCuadro, altoCuadro, brickColor));

}

//5ta Columna

for(int i=0; i<7; i++){

color brickColor=negroCasi;

float posCuadro\_Y=(espacioCielo+(altoCuadro+espacioEntreCuadros)\*i);

posCuadro\_X=5+anchoCuadro+(espacioEntreCuadros\*17);

grupoDeCuadros.add(new Bloque(posCuadro\_X, posCuadro\_Y, anchoCuadro, altoCuadro, brickColor));

}

//6ta Columna

for(int i=0; i<7; i++){

color brickColor=negroCasi;

float posCuadro\_Y=(espacioCielo+(altoCuadro+espacioEntreCuadros)\*i)-(altoCuadro+espacioEntreCuadros);

posCuadro\_X=5+anchoCuadro+(espacioEntreCuadros\*21);

grupoDeCuadros.add(new Bloque(posCuadro\_X, posCuadro\_Y, anchoCuadro, altoCuadro, brickColor));

}

//7ma Columna

for(int i=0; i<6; i++){

color brickColor=negroCasi;

float posCuadro\_Y=(espacioCielo+(altoCuadro+espacioEntreCuadros)\*i)-(altoCuadro+espacioEntreCuadros);

posCuadro\_X=5+anchoCuadro+(espacioEntreCuadros\*25);

grupoDeCuadros.add(new Bloque(posCuadro\_X, posCuadro\_Y, anchoCuadro, altoCuadro, brickColor));

}

//8va Columna

for(int i=0; i<5; i++){

color brickColor=negroCasi;

float posCuadro\_Y=(espacioCielo+(altoCuadro+espacioEntreCuadros)\*i)-(altoCuadro+espacioEntreCuadros);

posCuadro\_X=5+anchoCuadro+(espacioEntreCuadros\*29);

grupoDeCuadros.add(new Bloque(posCuadro\_X, posCuadro\_Y, anchoCuadro, altoCuadro, brickColor));

}

//9na Columna

for(int i=0; i<3; i++){

color brickColor=negroCasi;

float posCuadro\_Y=espacioCielo+(altoCuadro+espacioEntreCuadros)\*i;

posCuadro\_X=5+(anchoCuadro+espacioEntreCuadros\*33);

grupoDeCuadros.add(new Bloque(posCuadro\_X, posCuadro\_Y, anchoCuadro, altoCuadro, brickColor));

}

}

//DIBUJAR LOS CUADROS

void dibujarCuadros() {

for(int cantCuadros= grupoDeCuadros.size()-1; cantCuadros>=0; cantCuadros--) {

Bloque brick=grupoDeCuadros.get(cantCuadros);

brick.draw();

if(brick.collidesWith(Lala)) { //si al cuadro es chocado x bola, se elimina

grupoDeCuadros.remove(brick);

updatePuntaje(true);

}

}

}

//DIBUJAR LA BOLA

void dibujarBola() {

Lala.draw();

Lala.actualizarBola();

if(Lala.verColisionParedBola()) {

vidas--;

Lala.move(anchoVentana/2, altoVentana/2);

}

}

//dibujar la base y moverla con potenciometro

void dibujarBase() {

base.draw();

if(0<port.available()){// si estoy recibiendo valores por el puerto haga lo siguiente

//deme y guarde los datos del potenciometro

leer=port.read();

base.posBloque\_X=leer; //antes era mouseX

println(leer);

}

base.collidesWith(Lala);

//colocar imagen del cirgarrillo enla posicion de la base

image(cigarrilloBase,base.posBloque\_X,baseY-48,anchoBase,62); //(x,y,ancho,alto)

}

void mostrarTexto(String mje, int x, int y, boolean txtCentrado) {

fill(0);

textSize(20);

textFont(fuente,20); // el nro es el tamaño de a fuente pero no se recomienda cambiarlo porque se pixela

String nombreTxt= mje;

float posTxt\_X= x;

if (txtCentrado) {

float anchoTxt= textWidth(nombreTxt);

posTxt\_X= (anchoVentana-anchoTxt)/2;

}

int textY= y;

text(nombreTxt, posTxt\_X, textY);

}

void finalPerderPorTiempo() {

noStroke();

fill(#F2F2EB,127); //el 127 es de opacidad, representa 55% de opacidad

//la opacidad se representa de 0 a 255

//255 es un 100% de opacidad

rect(25, (altoVentana/2)-50, anchoVentana-50, 70, 7);

String mje="¡El tiempo se agota, No Fume!";

mostrarTexto(mje, 0, (altoVentana/2)-10, true);

haPerdido=true;

}

void finalPerder() {

noStroke();

fill(#F2F2EB,127); //el 127 es de opacidad, representa 55% de opacidad

//la opacidad se representa de 0 a 255

//255 es un 100% de opacidad

rect(25, (altoVentana/2)-50, anchoVentana-50, 70, 7);

String mje="¡¿Seguirá Fumando?!";

mostrarTexto(mje, 0, (altoVentana/2)-10, true);

haPerdido=true;

}

void updatePuntaje(boolean esNuevo) {

if (esNuevo) puntaje+=50; //cada q la bola toca un cuadro suma 50

mje="Puntaje:"+puntaje;

mostrarTexto(mje, 15, altoVentana-460, false);

//calcula total de puntos posibles y compara con puntaje

//49 son el total de cuadros que hice

if(puntaje==49\*50) { // si son iguales muestra letrero, columnasCuadros\*filasCuadros\*50

mostrarTexto("Nunca ganas con el cigarrillo",0,altoVentana/2,true);

haGanado=true;

}

}

void impTxtVidas() {

String mje;

mje="Vidas: " + vidas;

mostrarTexto(mje,((int)(anchoVentana-textWidth(mje))-15),altoVentana-460,false);

}

/\*\*\*\*\*\*\* BOLA \*\*\*\*\*\*\*\*\*\*/

class Bola {

float posBola\_X;

float posBola\_Y;

float anchoBola;

color colorBola;

float velY= 2.5;

float velX= 2.5;

//CONSTRUCTOR

//usar fuera de la clase:

//Bola nombreDeLaBola= new Bola(x, y, Width, Color);

Bola(float x, float y, int Ancho, color Color) {

posBola\_X= x;

posBola\_Y= y;

anchoBola= Ancho;

colorBola= Color;

}

//crear la elipse como tal con los datos de la bola

void draw() {

noStroke();

fill(colorBola);

ellipse(posBola\_X, posBola\_Y, anchoBola, anchoBola);

}

//desplazar bola segun velocidad;

void actualizarBola() {

posBola\_X+=velX;

posBola\_Y+=velY;

}

void move(int X, int Y) {

posBola\_X = X;

posBola\_Y = Y;

velY= 4;

velX= 4;

}

// bounce bola

boolean verColisionParedBola() {

if (posBola\_X>anchoVentana-anchoBola/2) {

velX=-abs(velX); //abs es encontrar el valor absoluto del nro, entonces el nro siempre sera positivo

} else if (posBola\_X<anchoBola/2) {

velX=abs(velX);

} if (posBola\_Y>altoVentana-anchoBola/2) {

velY=-abs(velY);

return true;

} else if (posBola\_Y<anchoBola/2) {

velY= abs(velY);

}

return false;

}

}

/\*\*\*\*\*\*\*\*\*\* BLOQUE \*\*\*\*\*\*\*\*\*\*\*\*\*\*/

class Bloque {

float posBloque\_X;

float posBloque\_Y;

float anchoBloque;

float altoBloque;

color colorBloque;

int maxGolpes= 1;

int golpes=maxGolpes;

//constructor

Bloque(float x, float y, float Width, float Height, color Color) {

posBloque\_X= x;

posBloque\_Y= y;

anchoBloque= Width;

altoBloque= Height;

colorBloque= Color;

}

//hacer q se muestren los bloques en pantalla

void draw() {

noStroke();

fill(colorBloque);

rect(posBloque\_X, posBloque\_Y, anchoBloque, altoBloque);

}

//Ubicar los bloques

//para centrar en X y Y

void move(int X, int Y) {

posBloque\_X=X-anchoBloque/2;

posBloque\_Y=Y-altoBloque/2;

//no dejar que se salga en eje X

if (posBloque\_X+anchoBloque>anchoVentana) {

posBloque\_X=anchoVentana-anchoBloque;

}

else if (posBloque\_X<0) {

posBloque\_X=0;

}

//no dejar que se salga en eje Y

if (posBloque\_Y+altoBloque>altoVentana) {

posBloque\_Y=altoVentana-anchoBloque;

}

else if (posBloque\_Y<0) {

posBloque\_Y=0;

}

}

//Identificar si algun bloque choca con la bola

// la bola cambia de velocidad automaticamente

boolean collidesWith(Bola b) {

//si bola choca con la parte inferior de un bloque

if ((b.posBola\_X+b.anchoBola/4>posBloque\_X && b.posBola\_X-b.anchoBola/4<posBloque\_X+anchoBloque)

&& (b.posBola\_Y-b.anchoBola/2<(posBloque\_Y+altoBloque) && b.posBola\_Y-anchoBola/2>posBloque\_Y)) {

println("Pego en un bloquee X debajo!!!");

b.velY=abs(b.velY);

golpes--;

return true;

}

//si bola choca con la parte superior de un bloque

if ((b.posBola\_X+b.anchoBola/4>posBloque\_X && b.posBola\_X-b.anchoBola/4<posBloque\_X+anchoBloque)

&& (b.posBola\_Y+b.anchoBola/2<posBloque\_Y+altoBloque && b.posBola\_Y+b.anchoBola/2>posBloque\_Y)) {

//println("Pego en un bloquee X debajo!!! ");

b.velY=-abs(b.velY);

golpes--;

return true;

}

//si bola choca con lado izq de un bloque

else if ((b.posBola\_Y+b.anchoBola/4>posBloque\_Y && b.posBola\_Y-b.anchoBola/4<posBloque\_Y+altoBloque)

&& (b.posBola\_X+b.anchoBola/2>posBloque\_X && b.posBola\_X+b.anchoBola/2<posBloque\_X+anchoBloque)) {

//println("Choco a lado IZQ");

b.velX=-abs(b.velX);

golpes--;

return true;

}

//si bola choca con lado derecho de un bloque

if ((b.posBola\_Y+b.anchoBola/4>posBloque\_Y && b.posBola\_Y-b.anchoBola/4<posBloque\_Y+altoBloque)

&& (b.posBola\_X-b.anchoBola/2<posBloque\_X+anchoBloque && b.posBola\_X-b.anchoBola/2>posBloque\_X)) {

//println("Choco a lado DER");

b.velX=abs(b.velX);

golpes--;

return true;

}

return false;

}

}