//Codul din aplicatie, avand in vedere ca obtinerea programului Adobe Animate nu este atat de facila

import flash.display.MovieClip;

import punct;

import Rocket;

import flash.events.MouseEvent;

import flash.display.DisplayObject;

var vitezaX: Number;

var vitezaY: Number;

var accGrav: Number;

var elasticitate: Number;

var frecare: Number;

var cadrecurente: Number = 0;

var cPressed: Boolean = false;

var rocketSpeed: Number;

var leftCharges: int;

var fAngle: Number;

var puppetLimit: int; //limita de puncte

var FREEZE: Boolean;

var settings: Boolean;

var ajutor: Boolean;

var allFrames: int = 0; //numarul de cadre

var pOre: int = 0; //ceas

var pMinute: int = 0;

var pSecunde: int = 0;

var pointss: Boolean = false; //daca sunt sau nu activate punctele in cazul in care mingea e oprita

var airfric: Number; //frecarea cu aerul, 1 off, 0.99 on

var cadruurma: int = 1; //cadrul pentru urme

airfric = 1;

puppetLimit = 300;

function getName() {

var newName: String;

cadrecurente++;

newName = "minge" + String(cadrecurente);

return newName;

}

freezeBtn.addEventListener(MouseEvent.CLICK, changeFreeze);

function changeFreeze(event: MouseEvent) {

if (FREEZE == false) {

FREEZE = true;

} else {

if (inputP.currentFrame == 1) {

FREEZE = false;

settings = false;

setMenu.visible = false;

}

}

}

setMenu.visible = false;

settings = false;

setBtn.addEventListener(MouseEvent.CLICK, openSettings);

function openSettings(event: MouseEvent) {

if (settings == true) { // inchidem setarile

settings = false;

setMenu.visible = false;

if (inputP.currentFrame == 1) {

FREEZE = false;

}

} else { // deschidem setarile

settings = true;

FREEZE = true;

setMenu.visible = true;

setChildIndex(setMenu, numChildren - 1);

}

}

revBtn.addEventListener(MouseEvent.CLICK, quitmini);

function quitmini(event: MouseEvent) {

setMenu.minion = false;

inputP.visible = true;

outputP.visible = true;

helpBtn.visible = true;

freezeBtn.visible = true;

setBtn.visible = true;

setMenu.visible = true;

mybtn.visible = true;

revBtn.visible = false;

setChildIndex(setMenu, numChildren - 1);

setChildIndex(inputP, numChildren - 1);

}

helpMenu.visible = false;

ajutor = false;

helpBtn.addEventListener(MouseEvent.CLICK, openHelp);

function openHelp(event: MouseEvent) {

if (ajutor == false) {

ajutor = true;

helpMenu.visible = true;

setChildIndex(helpMenu, numChildren - 1);

setChildIndex(helpBtn, numChildren - 1);

helpBtn.x += 100;

helpBtn.y -= 8;

if (inputP.currentFrame == 15) {

inputP.gotoAndPlay(16);

}

FREEZE = true;

} else {

ajutor = false;

helpMenu.visible = false;

helpBtn.x -= 100;

helpBtn.y += 8;

inputP.play();

}

}

//initializam valorile

minge.x = 100;

minge.y = 100;

vitezaX = 0;

vitezaY = 0;

accGrav = 9.8 / 3;

elasticitate = 0.85;

frecare = 0.2;

FREEZE = true;

var variabila: int = 0;

function cadru() {

stage.addEventListener(Event.ENTER\_FRAME, ballmove);

function ballmove(e: Event) {

allFrames++;

setMenu.secSimulare.text = String(variabila / 25);

if (allFrames % 25 == 0) {

pSecunde++;

}

if (pSecunde == 60) {

pSecunde = 0;

pMinute++;

}

if (pMinute == 60) {

pMinute = 0;

pOre++;

}

setMenu.nrOre.text = String(pOre);

setMenu.nrMinute.text = String(pMinute);

setMenu.nrSecunde.text = String(pSecunde);

if (inputP.currentFrame == 15) {

setChildIndex(inputP, numChildren - 1);

setChildIndex(freezeBtn, numChildren - 1);

setChildIndex(mybtn, numChildren - 1);

setChildIndex(setBtn, numChildren - 1);

setChildIndex(helpBtn, numChildren - 1);

} else {

setChildIndex(inputP, 1);

}

if (settings) {

FREEZE = true;

}

if (accGrav == 0 && elasticitate == 1 && frecare == 0 && airfric == 1) {

minge.gotoAndStop(9);

} else {

if (setMenu.culoarecho.text == "Culoarea mingii") {

minge.gotoAndStop(setMenu.ballBtn0.currentFrame);

}

}

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

if (FREEZE == true) {

freezeBtn.rotation = 45;

if (setMenu.minion == false) {

return;

}

}

variabila++;

freezeBtn.rotation = 90;

if (int(setMenu.nrBalls.text) != puppetLimit) {

if (int(setMenu.nrBalls.text) < 0) {

setMenu.nrBalls.text = "0";

}

var toStart: int;

var toFinish: int;

toFinish = cadrecurente;

if (cadrecurente <= puppetLimit) {

toStart = 1;

} else {

toStart = cadrecurente - puppetLimit + 1;

}

for (var i: int = toStart; i <= toFinish; i++) {

removeChild(getChildByName("minge" + String(i)));

}

cadrecurente = 0;

puppetLimit = int(setMenu.nrBalls.text);

}

if (Math.abs(vitezaX) > 1 || Math.abs(vitezaY) > 1 || pointss) {

var punctNou = new punct;

addChild(punctNou);

punctNou.name = getName();

punctNou.x = minge.x;

punctNou.y = minge.y;

punctNou.gotoAndStop(cadruurma);

//setChildIndex(punctNou, 1);

if (cadrecurente > puppetLimit) {

removeChild(getChildByName(String("minge") + String(cadrecurente - puppetLimit)));

}

}

vitezaY += 0.5 \* accGrav;

vitezaX \*= airfric;

vitezaY \*= airfric;

if (Math.abs(vitezaX) < 1)

vitezaX = 0;

if (Math.abs(vitezaY) < 1 && airfric == 2) {

vitezaY = 0;

}

outputP.xOutput.text = String(Math.abs(vitezaX));

outputP.yOutput.text = String(Math.abs(vitezaY));

outputP.xpOutput.text = String(int(minge.x));

outputP.ypOutput.text = String(int(minge.y));

if (Math.abs(vitezaX) < 1) {

outputP.xOutput.text = "0";

}

if (Math.abs(vitezaY) < 1) {

outputP.yOutput.text = "0";

}

minge.x += vitezaX;

minge.y += vitezaY;

if (minge.y >= 335) { //jos

minge.y = 335;

var vitezaCurenta: Number;

vitezaCurenta = Math.sqrt(vitezaX \* vitezaX + vitezaY \* vitezaY);

if (vitezaX > 0) {

vitezaX = Math.sqrt(Math.max(0, vitezaCurenta \* vitezaCurenta - 2 \* frecare \* accGrav - vitezaY \* vitezaY));

}

if (vitezaX < 0) {

vitezaX = (-1) \* Math.sqrt(Math.max(0, vitezaCurenta \* vitezaCurenta - 2 \* frecare \* accGrav - vitezaY \* vitezaY));

}

vitezaY \*= (-1) \* elasticitate;

}

if (minge.x >= 615) { //dreapta

minge.x = 615;

vitezaX \*= (-1) \* elasticitate;

}

if (minge.y <= 25) { //sus

minge.y = 25;

vitezaCurenta = Math.sqrt(vitezaX \* vitezaX + vitezaY \* vitezaY);

if (vitezaX > 0) {

vitezaX = Math.sqrt(Math.abs(vitezaCurenta \* vitezaCurenta - 2 \* frecare \* Math.abs(accGrav) - vitezaY \* vitezaY));

}

if (vitezaX < 0) {

vitezaX = (-1) \* Math.sqrt(Math.abs(vitezaCurenta \* vitezaCurenta - 2 \* frecare \* Math.abs(accGrav) - vitezaY \* vitezaY));

}

vitezaY \*= (-1) \* elasticitate;

}

if (minge.x <= 25) { //stanga

minge.x = 25;

vitezaX \*= (-1) \* elasticitate;

}

if (leftCharges > 0) {

if (leftCharges == 1) {

newRocket.alpha = 0.4;

}

leftCharges--;

newRocket.x += rocketSpeed \* Math.sin(fAngle);

newRocket.y -= rocketSpeed \* Math.cos(fAngle);

}

}

}

revBtn.visible = false;

mybtn.addEventListener(MouseEvent.CLICK, restartball);

function restartball(event: MouseEvent) {

if (inputP.currentFrame == 1) {

FREEZE = false;

}

variabila = 0;

vitezaX = 0;

vitezaY = 0;

vitezaX = Number(inputP.xinput.text);

vitezaY = Number(inputP.yinput.text);

accGrav = Number(inputP.ginput.text) / 3;

frecare = Number(inputP.finput.text);

frecare = Math.min(frecare, 1000.00);

frecare = Math.max(frecare, 0.00);

elasticitate = Number(inputP.einput.text);

elasticitate = Math.min(elasticitate, 1.00);

elasticitate = Math.max(elasticitate, 0.00);

inputP.xpinput.text = String(Math.max(0, Number(inputP.xpinput.text)));

inputP.ypinput.text = String(Math.max(0, Number(inputP.ypinput.text)));

inputP.xpinput.text = String(Math.min(640, Number(inputP.xpinput.text)));

inputP.ypinput.text = String(Math.min(360, Number(inputP.ypinput.text)));

minge.x = Number(inputP.xpinput.text);

minge.y = Number(inputP.ypinput.text);

if (newRocket.visible == true && newRocket.alpha == 1) {

repeatLaunch();

}

if (inputP.angleMode == true) {

var palfa: Number;

var pr: Number;

palfa = Number(inputP.xinput.text);

pr = Number(inputP.yinput.text);

var prezup: Number;

var prezdown: Number;

prezup = Math.cos(palfa / 180 \* Math.PI) \* pr;

prezdown = Math.sin(palfa / 180 \* Math.PI) \* pr;

if (Math.abs(prezup - Math.floor(prezup)) < 0.01) {

prezup = Math.floor(prezup);

}

if (Math.abs(prezup - Math.ceil(prezup)) < 0.01) {

prezup = Math.ceil(prezup);

}

if (Math.abs(prezdown - Math.floor(prezdown)) < 0.01) {

prezdown = Math.floor(prezdown);

}

if (Math.abs(prezdown - Math.ceil(prezdown)) < 0.01) {

prezdown = Math.ceil(prezdown);

}

vitezaX = prezup;

vitezaY = prezdown;

}

if (inputP.currentFrame == 15) {

inputP.gotoAndPlay(16);

}

settings = false;

setMenu.visible = false;

FREEZE = false;

}

outputP.gotoAndStop(1);

cadru();

var newRocket = new Rocket;

addChild(newRocket);

newRocket.name = "Rachetuta";

newRocket.visible = false;

function launchRocket(pozrX: Number, pozrY: Number) {

newRocket.alpha = 1;

newRocket.x = pozrX;

newRocket.y = pozrY;

newRocket.scaleX = 0.25;

newRocket.scaleY = 0.25;

newRocket.visible = true;

rocketSpeed = Number(inputP.rrrinput.text);

if (rocketSpeed < 0.1) {

rocketSpeed = 1;

inputP.rrrinput.text = "1";

}

if (rocketSpeed > 400) {

rocketSpeed = 400;

inputP.rrrinput.text = "400";

}

var spawnX: Number;

var spawnY: Number;

var lastX: Number;

var lastY: Number;

spawnX = newRocket.x;

spawnY = newRocket.y;

lastX = minge.x;

lastY = minge.y;

var maxDist: Number;

var currentDist: Number;

var found: Boolean;

var rezX: Number;

var rezY: Number;

var lastVX: Number;

var lastVY: Number;

var framesNum: Number;

var finalNum: Number;

framesNum = 0;

finalNum = 0;

lastVX = vitezaX;

lastVY = vitezaY;

found = false;

maxDist = 0;

while (maxDist <= 750 && found == false) {

framesNum++;

currentDist = Math.sqrt((spawnX - lastX) \* (spawnX - lastX) + (spawnY - lastY) \* (spawnY - lastY));

if ((Math.abs(currentDist - maxDist) <= 25) && (found == false)) {

rezX = lastX;

rezY = lastY;

found = true;

finalNum = framesNum;

}

//am verificat, acum modificam

if (Math.abs(lastVX) < 1)

lastVX = 0;

lastVY += 0.5 \* accGrav;

lastVX \*= airfric;

lastVY \*= airfric;

lastX += lastVX;

lastY += lastVY;

if (lastY >= 335) {

var vitezaCurenta: Number;

lastY = 335;

vitezaCurenta = Math.sqrt(lastVX \* lastVX + lastVY \* lastVY);

if (lastVX > 0) {

lastVX = Math.sqrt(Math.max(0, vitezaCurenta \* vitezaCurenta - 2 \* frecare \* accGrav - lastVY \* lastVY));

}

if (lastVX < 0) {

lastVX = (-1) \* Math.sqrt(Math.max(0, vitezaCurenta \* vitezaCurenta - 2 \* frecare \* accGrav - lastVY \* lastVY));

}

lastVY \*= (-1) \* elasticitate;

}

if (lastX >= 615) {

lastX = 615;

lastVX \*= (-1) \* elasticitate;

}

if (lastY <= 25) {

lastY = 25;

vitezaCurenta = Math.sqrt(lastVX \* lastVX + lastVY \* lastVY);

if (lastVX > 0) {

lastVX = Math.sqrt(Math.abs(vitezaCurenta \* vitezaCurenta - 2 \* frecare \* Math.abs(accGrav) - lastVY \* lastVY));

}

if (lastVX < 0) {

lastVX = (-1) \* Math.sqrt(Math.abs(vitezaCurenta \* vitezaCurenta - 2 \* frecare \* Math.abs(accGrav) - lastVY \* lastVY));

}

lastVY \*= (-1) \* elasticitate;

}

if (lastX <= 25) {

lastX = 25;

lastVX \*= (-1) \* elasticitate;

}

maxDist += rocketSpeed;

}

if (found == false) {

rezY = minge.y;

rezX = minge.x;

}

fAngle = Math.atan2(rezY - spawnY, rezX - spawnX);

fAngle += Math.PI / 2;

newRocket.rotation = fAngle \* 180 / Math.PI;

var punctt = new punct;

leftCharges = finalNum;

if (rocketSpeed == 400) {

leftCharges = 0;

newRocket.x = minge.x;

newRocket.y = minge.y;

newRocket.alpha = 0.5;

}

}

function repeatLaunch() {

if (newRocket.visible == true) {

launchRocket(newRocket.x, newRocket.y);

}

}

var startTime: Number;

var framesNumber: Number = 0;

function fpsCounter(): void {

startTime = getTimer();

addEventListener(Event.ENTER\_FRAME, checkFPS);

}

function checkFPS(e: Event): void {

var currentTime: Number = (getTimer() - startTime) / 1000;

framesNumber++;

if (currentTime > 1) {

outputP.fpsText.text = (Math.floor((framesNumber / currentTime) \* 10.0) / 10.0);

startTime = getTimer();

framesNumber = 0;

}

}

fpsCounter();

function onKeyPress(evt: KeyboardEvent) {

if (evt.keyCode == 67) {

cPressed = true;

launchRocket(mouseX, mouseY);

}

}

function onKeyLet(evt2: KeyboardEvent) {

if (evt2.keyCode == 67) {

cPressed = false;

}

}

stage.addEventListener(KeyboardEvent.KEY\_UP, onKeyLet);

stage.addEventListener(KeyboardEvent.KEY\_DOWN, onKeyPress);