Relatório do trabalho de modelagem em fenômenos físicos no Processing

O nosso trabalho é inspirado no jogo Angry Birds, e criamos assim três “pássaros” que tem funcionalidades parecidas com as que os pássaros nos jogos originais possuem, e com o objetivo comum de acertar o porco verde.

O primeiro é o pássaro vermelho, que se comporta igual a um projétil convencional ao ser lançado, sem nenhum tipo de poder extra.

x0 = 1.0

y0 = 5.0

vx = 5.0

vy = -5.0

t0 = millis()

oldt = t0/1000.0

*Aqui, são definidas as variáveis de início: posição inicial (x0 e y0), velocidades iniciais em cada eixo, e definida a função tempo.*

def setup():

size(1200,600)

def draw():

global oldt, x0, y0, vx, vy

t = (millis()-t0)/1000.0

x = 100\*(x0 + vx\*t)

y = 100\*(y0 + vy\*t+ 5\*t\*t)

background(169, 216, 229)

stroke(0)

fill(240, 0, 30)

ellipse(x,y, 13, 13)

fill(0,255,0)

ellipse(685,585,30,30)

oldt = t

*Aqui, fica definida a trajetória do projétil, com o movimento no eixo horizontal sendo uniforme e no eixo vertical uniformemente variado. A primeira elipse definida é o projétil, e a segunda é o “porco” a ser atingido.*

def mouseClicked():

global oldt, t0

t0 = millis()

oldt = t0/1000.0

*Aqui, fica definido que ao clicar no mouse, a simulação reseta.*

Depois, fizemos o código para o pássaro amarelo, que aumenta a sua velocidade quando qualquer tecla do teclado é pressionada.

x = 1.0

y = 500.0

vx = 200.0

vy = -200.0

t0 = millis()/1000

g = 200

*De novo, condições iniciais. Entretanto, foi necessário realizar uma mudança de escala para que a simulação ocorresse como desejado, por isso o valores estão tão elevados.*

def setup():

size(1200,600)

def draw():

global x, y, vx, vy, g, t0

t = millis()/1000.0

dt=t-t0

t0=t

y = (y + vy\*dt)

vy = vy + g\*dt

x = (x + vx\*dt)

background(169, 216, 229)

stroke(0)

fill(240, 240, 0)

ellipse(x,y, 13, 13)

fill(0,255,0)

ellipse(685,585,30,30)

if keyPressed == True:

vx = 1.5\*vx

vy = 1.5\*vy

*Define-se aqui a posição da partícula a partir de sua posição anterior, adicionando apenas o valor da velocidade multiplicado pelo tempo. Depois configuramos a velocidade vertical para ser a velocidade anterior acrescida do valor da aceleração multiplicado pelo tempo. A velocidade horizontal é constante. Também fica programado que, no caso de haver qualquer tecla pressionada, as duas velocidades aumentam em 50% o seu valor.*

Por fim, fizemos o pássaro branco, que quando clicamos com o mouse põe um ovo e recebe um leve impulso na vertical.

x0 = 1.0

y0 = 5.0

vx = 6.0

vy = -6.0

t0 = millis()

oldt = t0/1000.0

u = -10

k=-10

v = 0

import copy

*Aqui, novamente são definidas as condições iniciais. Os valores de u, k e v foram definidos para que seja possível construir a elipse que será o ovo posteriormente. Importamos também a biblioteca copy, que será utilizada na hora de lançar o ovo.*

def setup():

size(1200,600)

def draw():

global oldt, x0, y0, vx, vy,u,k,v

t = (millis()-t0)/1000.0

x = 100\*(x0 + vx\*t)

y = 100\*(y0 + vy\*t+ 5\*t\*t)

background(169, 216, 229)

stroke(0)

fill(255)

ellipse(x,y, 13, 13)

fill(0,255,0)

ellipse(685,585,30,30)

oldt = t

if mousePressed:

u = copy.copy(x)

k = copy.copy(y)

vy = 1.1\*vy

v = 100\*(k + 5\*t\*t)

fill(255)

ellipse(u,v,10,15)

v = 100\*(k + 5\*t\*t)

*Nesse código, quando o mouse está sendo pressionado, são criadas cópias dos valores das posições horizontal e vertical do pássaro no instante em que clicamos, e também há um aumento de 10% na velocidade vertical. Com as cópias, iremos definir a elipse que será o ovo. Ele cai em queda livre desde o momento que é posto, por isso consideramos o seu movimento na vertical desta forma, com a posição inicial sendo k e aceleração sendo a aceleração da gravidade.*