Evan Nichols

EECS 368

Homework 2

*Description*

The thingsclass.js drives the interaction and animation of index.html. It uses two Promises to procure from the user a number of balls and a strength of gravity for the animation.

A Promise.all() function takes the two chalk.entrybox() functions in as arguments. Once fulfilled, a function reads the return values of the Promises (in this case, two integer values), and uses them to instantiate a new Things object.

The animation and "gravity" effect on the balls is handled within the move function of the Circle object. In lines 25-32 of thingsclass.js, we see how position changes based on initial velocity and gravity. The x position changes each step with the xvel, and the y position changes with both yvel and gravity. Every time a ball hits a wall, energy is lost. The if statements account for these energy losses as xvel and yvel are fractionally decreased with each wall bounce.

*Index.html*

<html>

<head>

<script src="http://code.jquery.com/jquery-1.10.2.min.js"></script>

<script src="https://www.promisejs.org/polyfills/promise-7.0.1.min.js"></script>

<script src="chalk.20151006.js"></script>

<script src="thingsclass.js"></script>

</head>

<body>

<div id="chalk"></div>

</body>

</html>

*thingsclass.js*

var Circle = function(a,b,r,x,y,g,col) {

var xpos = a;

var ypos = b;

var xvel = x;

var yvel = y;

var grav = g;

var rad = r;

this.toString = function() {

return "Circle { x = " + x + ", y = " + y + ", r = " + r + "}";

}

this.draw = function(gc) {

gc.beginPath();

gc.arc(xpos,ypos,rad, 0, 2 \* Math.PI, false);

gc.fillStyle = col;

gc.fill();

}

this.move = function(sz) {

//need position to change based off initial velocity and passed acceleration.

xpos += xvel;

if (xpos < rad) { xpos = rad; xvel = -xvel\*0.5; };

if (xpos > sz - rad) { xpos = sz - rad; xvel = -xvel\*0.5; };

yvel += grav;

ypos += yvel;

if (ypos < rad) { ypos = rad; yvel = -yvel\*0.8; };

if (ypos > sz - rad) { ypos = sz - rad; yvel = -yvel\*0.8; };

}

};

var jayhawks = ["#0022B4","#E8000D","#FFC82D"];

var randomColor = function()

{

return jayhawks[Math.floor(Math.floor(Math.random() \* jayhawks.length))];

}

var Things = function(sz,count, grav) {

var shapes = []; // new Array();

var that = this;

for (var k = 0; k < count; k++) {

shapes[k] = new Circle(

Math.random() \* sz,

Math.random() \* sz,

10,

10 \* (Math.random() - 0.5),

10 \* (Math.random() - 0.5),

grav,

randomColor()

);

}

this.step = function(gc) {

for (var k = 0; k < shapes.length; k++) {

shapes[k].move(sz);

}

gc.clearRect (0,0,sz,sz);

for (var k = 0; k < shapes.length; k++) {

shapes[k].draw(gc);

}

chalk.delay(20).then(function (){ that.step(gc); });

} //end step

this.addThing = function(x,y) {

shapes.push(

new Circle(

Math.random() \* sz,

Math.random() \* sz,

10,

10 \* (Math.random() - 0.5),

10 \* (Math.random() - 0.5),

grav,

randomColor()

)

);

}

};

main = function()

{

chalk.println("How many balls?");

var pX = chalk.entrybox();

chalk.println("Strength of gravity?");

var pY = chalk.entrybox();

var sz = 400;

Promise.all([pX,pY]).then(function (arr) {

var gc = chalk.canvas(sz,sz);

var x = arr[0];

var y = arr[1];

chalk.println("Gravity is: "+y);

var uni = new Things(sz,parseInt(x),parseInt(y));

uni.step(gc);

//uni.addThing(parseInt(x),parseInt(y));

});

}