**PRO-C34**

Tema: Bola de demolición

**Descripción:** Construir la simulación de la Bola de Demolición, revisando

los conceptos del Juego Angry Birds.

**Objetivos:** ● Construir la simulación de la Bola de Demolición, revisando

los conceptos del Juego Angry Birds.

¿Recuerdas algunos conceptos que aprendiste en las últimas clases?: **REA:** - Variables - Programación condicional – Bucles – Sprites - Objetos y funciones construidos sobre los objetos

Mientras hablamos, Angry Birds fue uno de los juegos que fue creado por programadores para teléfonos inteligentes de Apple, y futuras tabletas. También, fue uno de los juegos más jugados de esa época. Ahora que hemos visto cómo crear un juego de Angry Birds usando el motor físico, intentemos hacer una simulación de una bola de demolición

¿Sabes qué es una bola de demolición?: REA: Las bolas de demolición son las que se utilizan para romper grandes edificios.

En la clase de hoy, crearemos una simulación de una bola de demolición, mientras repasaremos todos los conceptos que aprendimos en la clase anterior:

Si recuerdas, en el Juego Angry Birds, descompusimos o dividimos todo el juego en componentes más pequeños: Bird (Ave), Pigs (Cerdos), SlingShot (Resortera), Obstacles, etc. Esto se llama descomposición. Antes de trabajar en cualquier proyecto complejo, un programador divide la tarea compleja en tareas más

pequeñas y sencillas en las que puede pensar cómo abordarla.

Aquí también, todo el proyecto es bastante complejo. ¿Cómo lo dividirías en componentes más simples?: REA: - Cuerda – Bolas – Cajas Opcionalmente, podemos agregar movimientos al Arrastrar (Drag) el ratón.

Comencemos a codificar para programar cada uno de los pequeños componentes del juego. Observarás cómo se construye toda la simulación, resolviendo los pequeños componentes del juego.

Creamos a clase Ground.js

class Ground {

*constructor (x,y,width,height){*

*var options = {isStatic: true}*

*this.body = Bodies.rectangle(x,y,width,height,options);*

*this.width = width;*

*this.height = height;*

*World.add(world,this.body); }*

*display(){*

*var pos = this.body.position;*

*rectMode(CENTER);*

*fill("yellow");*

*rect(pos.x, pos.y, this.width, this.height); }}*

En index.html

*<script src="Ground.js"></script>*

En sketch.js

*const Engine = Matter.Engine;*

*const World = Matter.World;*

*const Bodies = Matter.Bodies;*

*const Constraint = Matter.Constraint;*

*var engine, world;*

*var ground1;*

*function setup(){*

*createCanvas (1200,700);*

*engine = Engine.create();*

*world = engine.world;*

*ground1 = new Ground(600,690,1200,20); }*

*function draw(){*

*background("red");*

Engine.update(engine);

ground1.display();}

Ayude al alumno a recordar cómo se crearon las cajas de los obstáculos, de la misma manera, ayude al alumno a crear varias cajas.

Box.js

*class Box{*

*constructor(x,y,width,height){*

*var options = {restitution: 0.7, density:0.5, friction: 0.7}*

*this.body = Bodies.rectangle(x,y,width,height,options);*

*this.width = width;*

*this.height = height;*

*World.add(world,this.body); }*

*display(){*

*var pos = this.body.position;*

*var angle = this.body.angle;*

*push();*

*translate(pos.x, pos.y);*

*rotate(angle);*

*rectMode(CENTER);*

*strokeWeight(4);*

*stroke("blue");*

*fill("green");*

*rect(0,0,this.width, this.height);*

*pop();} }*

Agregándolo en Index.html

*<script src="Box.js"></script>*

Dentro de sketch.js

*var box1,box2,box3,box4,box5,box6,box7;*

*box1 = new Box(800,670,50,50);*

*box2 = new Box(850,670,50,50);*

*…*

*box14 = new Box(850,380,50,50);*

*box1.display();*

*…*

*box14.display();*

Ayude al alumno a crear la bola. En Ball.js

*class Ball{*

*constructor(x,y,width,height){*

*var options = {restitution: 0.7, friction:0.7, density:0.7}*

*this.body = Bodies.rectangle(x,y,width,height,options);*

*this.width = width;*

*this.height = height;*

*World.add(world, this.body); }*

*display(){*

*var pos = this.body.position;*

*var angle = this.body.angle;*

*push()*

*translate(pos.x, pos.y);*

*rotate(angle);*

*strokeWeight(4);*

*stroke("purple");*

*fill("white");*

*ellipse(0,0,this.width, this.height);*

*pop(); } }*

En index.html

*<script src="Ball.js"></script>*

En sketch.js

*var ball1;*

*ball1 = new Ball(600,600,45,45);*

*ball1.display();*

Ayude al alumno a recordar cómo se creó la resortera; ayude al alumno a crear la cuerda y a sujetar la bola a la cuerda.

El alumno escribe un código para crear una cuerda y unirla a la bola.

*class Rope{*

*constructor(bodyA, pointB){*

*var options = {bodyA: bodyA, pointB: pointB, stiffness: 1.2, length: 250}*

*this.pointB = pointB;*

*this.body = Constraint.create(options);*

*World.add(world, this.body); }*

*attach(){ this.body.bodyA = body; }*

*fly(){ this.body.bodyA = null; }*

*display(){*

*if(this.body.bodyA){*

*var pointA = this.body.bodyA.position;*

*var pointB = this.pointB;*

*push();*

*strokeWeight(4);*

*stroke("black");*

*line(pointB.x, pointB.y, pointA.x, pointA.y);*

*pop(); } } }*

En index.html

*<script src="Rope.js"></script>*

En sketch.js

*var rope1;*

*rope1 = new Rope(ball1.body, {x:500, y:250});*

*rope1.display();*

Ayude al alumno a hacer que la bola se mueva arrastrando el ratón.

*function mouseDragged(){ Matter.Body.setPosition(ball1.body,{x: mouseX, y: mouseY}); }*