Wrecking ball

What is our GOAL for this MODULE?

The goal of this module is to review the concepts used in the Angry Bird Game, while creating the Wrecking ball simulation.

What did we ACHIEVE in the class TODAY?

- We created a Wrecking ball simulation.

Which CONCEPTS/CODING BLOCKS did we cover today?

- Concept of classes
- Usage of matter.js library

How did we DO the activities?

1. We created a ground class.

Ground.js > Ground

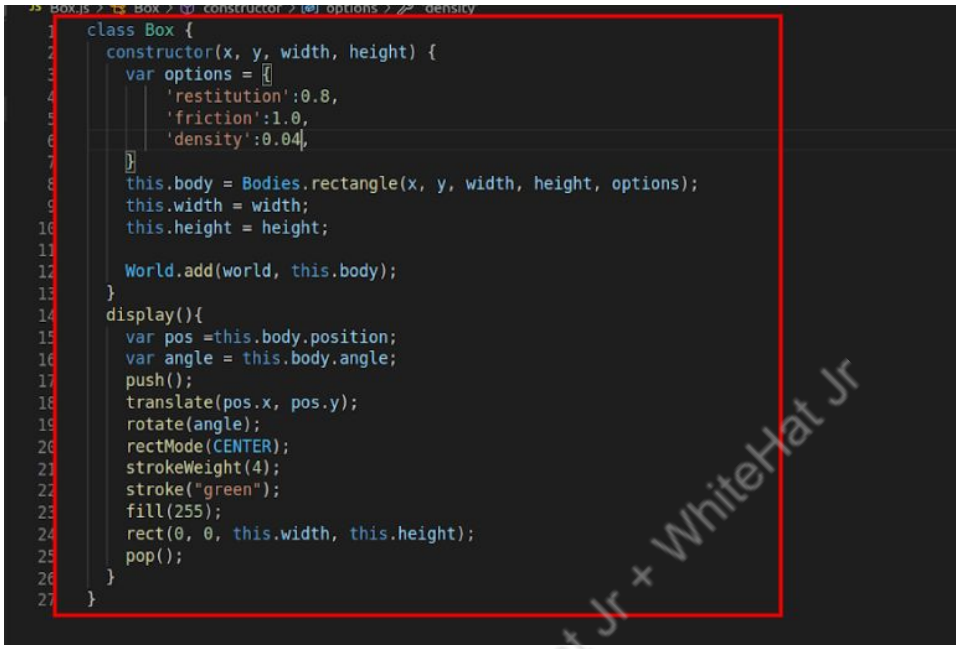
```
class Ground {
    constructor(x,y,width,height) {
      var options = {
          isStatic: true
      }
      this.body = Bodies.rectangle(x,y,width,height,options);
      this.width = width;
      this.height = height;
      World.add(world, this.body);
    }
    display(){
      var pos =this.body.position;
      rectMode(CENTER);
      fill("brown");
      rect(pos.x, pos.y, this.width, this.height);
    }
  };
```

2. We created a ground object using the ground class and displayed it.

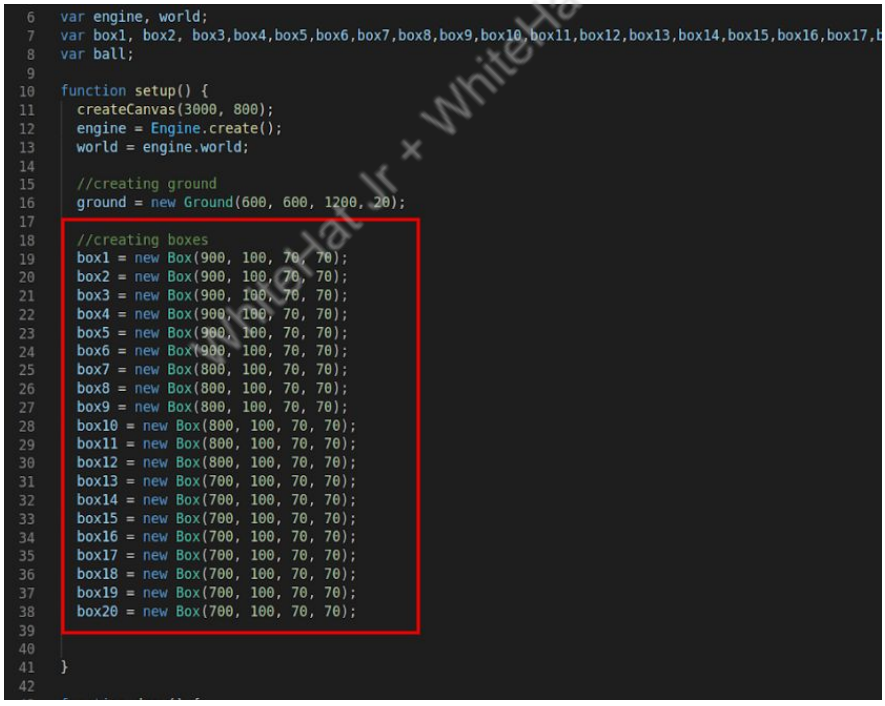
```
const Engine = Matter.Engine;
const World = Matter.World;
const Bodies = Matter.Bodies;
const Constraint = Matter.Constraint;

var engine, world;
var box1, box2, box3,box4,box5,box6,box7,box8,box9,box10,box11,box12,box13,box14,
var ball;

function setup() {
  createCanvas(3000, 800);
  engine = Engine.create();
  world = engine.world;

  //creating ground
  ground = new Ground(600, 600, 1200, 20);

  //creating boxes
  box1 = new Box(900, 100, 70, 70);
  box2 = new Box(900, 100, 70, 70);
  box3 = new Box(900, 100, 70, 70);
  box4 = new Box(900, 100, 70, 70);
  box5 = new Box(900, 100, 70, 70);
  box6 = new Box(900, 100, 70, 70);
  box7 = new Box(800, 100, 70, 70);
  box8 = new Box(800, 100, 70, 70);
  box9 = new Box(800, 100, 70, 70);
  box10 = new Box(800, 100, 70, 70);
  box11 = new Box(800, 100, 70, 70);
  box12 = new Box(800, 100, 70, 70);
  box13 = new Box(700, 100, 70, 70);
```

```
JS sketch.js > ...

}

function draw() {
  background(180);
  Engine.update(engine);

  //displaying the ground
  ground.display();

  //displaying the boxes
  box1.display();
  box2.display();
  box3.display();
  box4.display()
  box5.display()
  box6.display()
  box7.display()
  box8.display()
  box9.display()
  box10.display()
  box11.display()
  box12.display()
  box13.display()
  box14.display()
  box15.display()
  box16.display()
  box17.display()
  box18.display()
  box19.display()
  box20.display()
```

3. We also created the box class, Using the box classes we created multiple box objects and displayed them.



```
class Box {
  constructor(x, y, width, height) {
    var options = {
        'restitution':0.8,
        'friction':1.0,
        'density':0.04,
    }
    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("green");
    fill(255);
    rect(0, 0, this.width, this.height);
    pop();
  }
}
```



```
var engine, world;
var box1, box2, box3,box4,box5,box6,box7,box8,box9,box10,box11,box12,box13,box14,box15,box16,box17,b
var ball;

function setup() {
  createCanvas(3000, 800);
  engine = Engine.create();
  world = engine.world;

  //creating ground
  ground = new Ground(600, 600, 1200, 20);

  //creating boxes
  box1 = new Box(900, 100, 70, 70);
  box2 = new Box(900, 100, 70, 70);
  box3 = new Box(900, 100, 70, 70);
  box4 = new Box(900, 100, 70, 70);
  box5 = new Box(900, 100, 70, 70);
  box6 = new Box(900, 100, 70, 70);
  box7 = new Box(800, 100, 70, 70);
  box8 = new Box(800, 100, 70, 70);
  box9 = new Box(800, 100, 70, 70);
  box10 = new Box(800, 100, 70, 70);
  box11 = new Box(800, 100, 70, 70);
  box12 = new Box(800, 100, 70, 70);
  box13 = new Box(700, 100, 70, 70);
  box14 = new Box(700, 100, 70, 70);
  box15 = new Box(700, 100, 70, 70);
  box16 = new Box(700, 100, 70, 70);
  box17 = new Box(700, 100, 70, 70);
  box18 = new Box(700, 100, 70, 70);
  box19 = new Box(700, 100, 70, 70);
  box20 = new Box(700, 100, 70, 70);


}
```

```
function draw() {
  background(180);
  Engine.update(engine);

  //displaying the ground
  ground.display();

  //displaying the boxes
  box1.display();
  box2.display();
  box3.display();
  box4.display()
  box5.display()
  box6.display()
  box7.display()
  box8.display()
  box9.display()
  box10.display()
  box11.display()
  box12.display()
  box13.display()
  box14.display()
  box15.display()
  box16.display()
  box17.display()
  box18.display()
  box19.display()
  box20.display()


}
```

4. We then created the ball class and using this class we created a ball object and displayed it.

```
  box15 = new Box(700, 100, 70, 70);
  box16 = new Box(700, 100, 70, 70);
  box17 = new Box(700, 100, 70, 70);
  box18 = new Box(700, 100, 70, 70);
  box19 = new Box(700, 100, 70, 70);
  box20 = new Box(700, 100, 70, 70);

  //creating ball
  ball = new Ball(200, 200, 80, 80);

  //creating rope and attaching the ball
  rope = new Rope(ball.body, { x: 500, y: 50 });

}

function draw() {
  background(180);
  Engine.update(engine);

  //displaying the ground
  ground.display();

  //displaying the boxes
  box1.display();
  box2.display();
  box3.display();
```

```
66    box13.display()
67    box14.display()
68    box15.display()
69    box16.display()
70    box17.display()
71    box18.display()
72    box19.display()
73    box20.display()
74
75
76    //displaying the rope and ball
77    ball.display();
78    rope.display();
79
80  }
81
82
83  function mouseDragged() {
84    Matter.Body.setPosition(ball.body, { x: mouseX, y: mouseY });
85  }
86
```

5. Then we created the rope class and created a rope object and attached the ball to the rope objects and displayed it.

```
JS Rope.js > Rope > constructor
1   class Rope {
2     constructor(bodyA, pointB) {
3       var options = {
4         bodyA: bodyA,
5         pointB: pointB,
6         stiffness: 1.2,
7         length: 250,
8       };
9
10      this.pointB = pointB;
11      this.rope = Constraint.create(options);
12      World.add(world, this.rope);
13    }
14    attach(body) {
15      this.rope.bodyA = body;
16    }
17
18    fly() {
19      this.rope.bodyA = null;
20    }
21
22    display() {
23      if (this.rope.bodyA) {
24        var pointA = this.rope.bodyA.position;
25        var pointB = this.pointB;
26        push();
27
28        stroke(48, 22, 8);
29        strokeWeight(3);
30
31        line(pointB.x, pointB.y, pointA.x, pointA.y);
32
33        pop();
34      }
35    }
36  }
37
```

```
box15 = new Box(700, 100, 70, 70);
box16 = new Box(700, 100, 70, 70);
box17 = new Box(700, 100, 70, 70);
box18 = new Box(700, 100, 70, 70);
box19 = new Box(700, 100, 70, 70);
box20 = new Box(700, 100, 70, 70);

//creating ball
ball = new Ball(200, 200, 80, 80);

//creating rope and attaching the ball
rope = new Rope(ball.body, { x: 500, y: 50 });

}

function draw() {
  background(180);
  Engine.update(engine);

  //displaying the ground
  ground.display();

  //displaying the boxes
  box1.display();
  box2.display();
  box3.display();
```

```
  box13.display()
  box14.display()
  box15.display()
  box16.display()
  box17.display()
  box18.display()
  box19.display()
  box20.display()


  //displaying the rope and ball
  ball.display();
  rope.display();

}


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

6. We coded to move the ball with the mouse.

```
67  box14.display()
68  box15.display()
69  box16.display()
70  box17.display()
71  box18.display()
72  box19.display()
73  box20.display()
74
75
76  //displaying the rope and ball
77  ball.display();
78  rope.display();
79
80  }
81
82
83  function mouseDragged() {
84    Matter.Body.setPosition(ball.body, { x: mouseX, y: mouseY });
85  }
86
87
88
```

What's NEXT?

In the next class, we will be learning new concepts and building new projects.

EXTEND YOUR KNOWLEDGE:

You can explore more examples of matter.js and try to replicate them.
https://brm.io/matter-js/demo/#mixed