

CRASHING TRAM



What is our GOAL for this MODULE?

We used our knowledge about the Angry Birds Game while creating the simulation of a crashing tram.

What did we ACHIEVE in the class TODAY?

- Learned to apply force.
- Learned about inertia and mass.
- Learned about the SAT theorem and implemented it.

Which CONCEPTS/ CODING BLOCKS did we cover today?

- Flag property
- Matter.Body.applyForce(body, position, force)

How did we DO the activities?

1. Use the same code from Angry Birds to create the ground.

```
JS Ground.js > Ground
1 class Ground {
2   constructor(x,y,width,height) {
3     var options = {
4       isStatic: true
5     }
6     this.body = Bodies.rectangle(x,y,width,height,options);
7     this.width = width;
8     this.height = height;
9     World.add(world, this.body);
10  }
11  display(){
12    var pos =this.body.position;
13    rectMode(CENTER);
14    fill("brown");
15    rect(pos.x, pos.y, this.width, this.height);
16  }
17  };
```

2. Create a similar class to the ground class and name it as Boggie and Rock class.

```
JS Rock.js x
JS Rock.js > Rock
1 class Rock{
2   constructor(x,y,width,height){
3     var options ={ isStatic: false}
4     this.body = Bodies.rectangle(x,y,width,height,options);
5
6     this.width = width;
7     this.height = height;
8     this.image=loadImage("images/rock1.png");
9     World.add(myWorld,this.body);
10  }
11  show()
12  {
13    var pos = this.body.position;
14
15    //this.body.position.x = mouseX;
16    //this.body.position.y = mouseY;
17    imageMode(CENTER);
18    fill('black');
19    stroke ("grey");
20    image(this.image,pos.x,pos.y,this.width, this.height);
21  }
22
23
24 }
25
```

```
JS Boggie.js • JS sketch.js
JS Boggie.js > ...
1
2
3 class Boggie{
4   constructor(x,y,w,h){
5   var options= {
6     isStatic:false
7   }
8   this.body= Bodies.rectangle(x,y,w,h,options);
9   this.w = w;
10  this.h=h;
11  this.image=loadImage("images/coach.png");
12  World.add(myWorld,this.body);
13  }
14  show(){
15    imageMode(CENTER);
16    image(this.image,this.body.position.x,this.body.position.y,this.w,this.h);
17  }
18  }
19 }
```

3. Replace the slingshot with the chain class.

```
JS Chain.js > Chain
1 class Chain{
2   constructor(bodyA,bodyB){
3     var options ={
4       bodyA: bodyA,
5       bodyB: bodyB,
6       stiffness: 0.5
7     }
8     this.chain = Constraint.create(options);
9     World.add(myWorld,this.chain);
10  }
11
12  show(){
13    var pos1 = this.chain.bodyA.position;
14    var pos2 = this.chain.bodyB.position;
15    stroke(0)
16    strokeWeight(5);
17    line(pos1.x,pos1.y,pos2.x, pos2.y);
18  }
19 }
```

4. Apply the concept of applyforce in matter.js.

rectangles, circles and other polygons) can be found in the module `Matter.Bodies`.

See the included usage [examples](#).

[Methods](#) | [Properties](#) | [Events](#)

Methods

`Matter.Body.applyForce(body, position, force)`

Applies a force to a body from a given world-space position, including resulting torque.

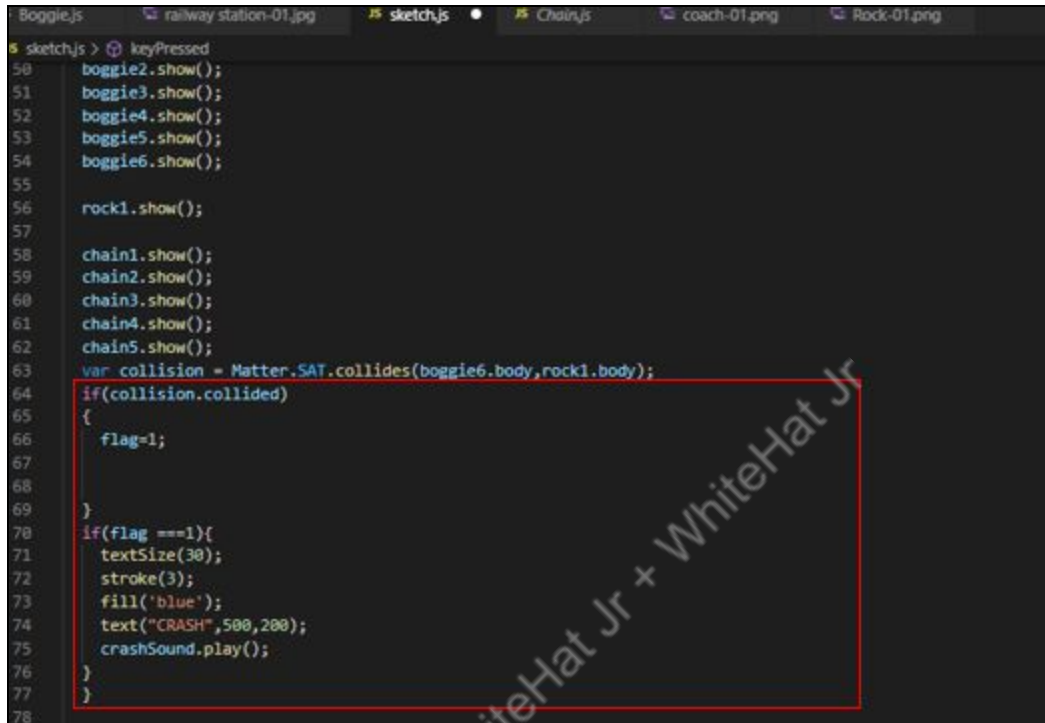
Parameters

<code>body</code>	<code>Body</code>
<code>position</code>	<code>Vector</code>
<code>force</code>	<code>Vector</code>

@ src/body/Body.js:640

```
sketchjs > draw
3
4
5 }
6
7 function draw() {
8   background(bg);
9   Engine.update(myEngine);
10
11   boggie1.show();
12   boggie2.show();
13   boggie3.show();
14   boggie4.show();
15   boggie5.show();
16   boggie6.show();
17
18   rock1.show();
19
20   chain1.show();
21   chain2.show();
22   chain3.show();
23   chain4.show();
24   chain5.show();
25   var collision = Matter.SAT.collides(boggie6.body, rock1.body);
```

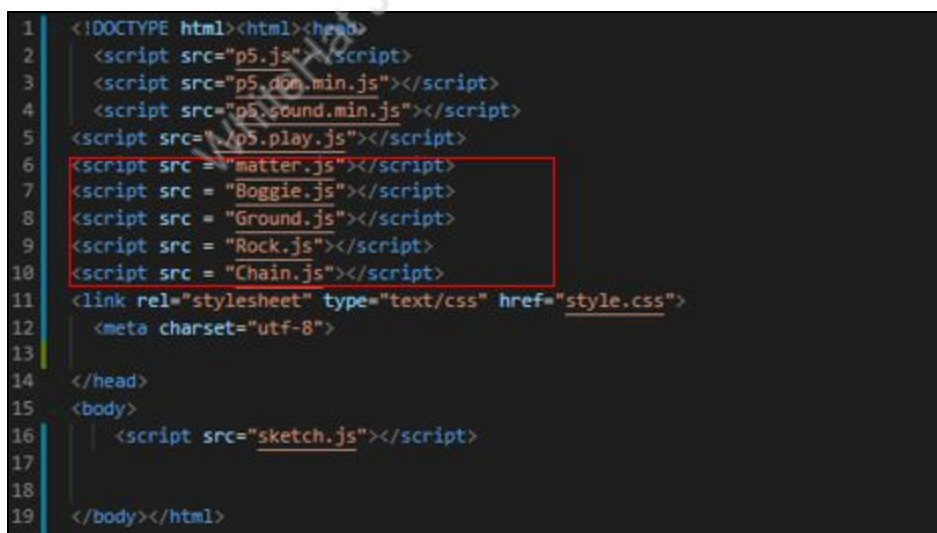
- Use the collided condition to check whether the two objects (tram and engine) have collided.



```

Boggie.js  railway station-01.jpg  sketch.js  Chain.js  coach-01.png  Rock-01.png
5 sketch.js > keyPressed
50 boggie2.show();
51 boggie3.show();
52 boggie4.show();
53 boggie5.show();
54 boggie6.show();
55
56 rock1.show();
57
58 chain1.show();
59 chain2.show();
60 chain3.show();
61 chain4.show();
62 chain5.show();
63 var collision = Matter.SAT.collides(boggie6.body, rock1.body);
64 if(collision.collided)
65 {
66     flag=1;
67
68 }
69
70 if(flag ==1){
71     textSize(30);
72     stroke(3);
73     fill('blue');
74     text("CRASH",500,200);
75     crashSound.play();
76 }
77 }
78
  
```

- Include the class file into index.html.



```

1 <!DOCTYPE html><html><head>
2   <script src="p5.js"></script>
3   <script src="p5.dom.min.js"></script>
4   <script src="p5.sound.min.js"></script>
5   <script src="p5.play.js"></script>
6   <script src = "matter.js"></script>
7   <script src = "Boggie.js"></script>
8   <script src = "Ground.js"></script>
9   <script src = "Rock.js"></script>
10  <script src = "Chain.js"></script>
11  <link rel="stylesheet" type="text/css" href="style.css">
12  <meta charset="utf-8">
13
14 </head>
15 <body>
16   <script src="sketch.js"></script>
17
18
19 </body></html>
  
```

What's NEXT?

In the next class, you will be learning about real-time databases.

EXTEND YOUR KNOWLEDGE:

1. This document contains a detailed description of matter.js. You can explore it to learn more about it: <https://brm.io/matter-js/docs/>

WhiteHat Jr + WhiteHat Jr + WhiteHat Jr