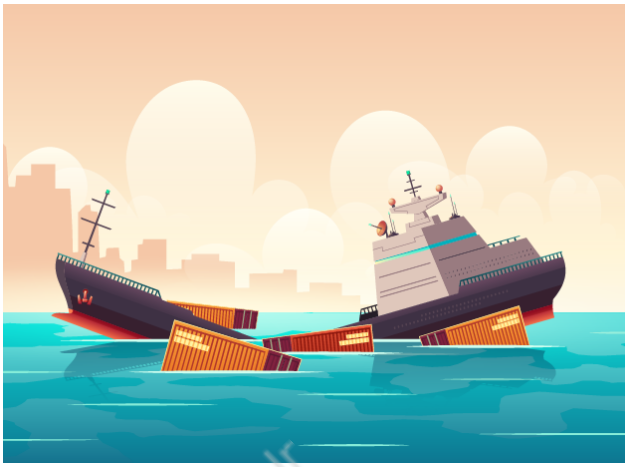
CREATING BOATS



What is our GOAL for this MODULE?

In this class, we learned to create a **Boat** class to create multiple boat objects using an array. We also learned to detect the collision between boats and cannonballs.

What did we ACHIEVE in the class TODAY?

- Created a **Boat** class.
- Created a new boat object and displayed it.
- Gave some velocity to the boat.
- Created multiple boats using **showBoats()**.
- Wrote code to check the collision between ball and boat.

Which CONCEPTS/ CODING BLOCKS did we cover today?

- OOPs concept
- **Matter.SAT.collides()**
- **remove()**

How did we DO the activities?

1. Create a **Boat class** as in our game the enemies are going to be the pirates and the pirates travel the sea on their boats.

Boat.js > Boat

```
class Boat {
  constructor(x, y, width, height, boatPos) {
    var options = {
      restitution: 0.8,
      friction: 1.0,
      density: 1.0
    };

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

    this.image = loadImage("./assets/boat.png");
    this.boatPosition = boatPos;
    World.add(world, this.body);
  }
```

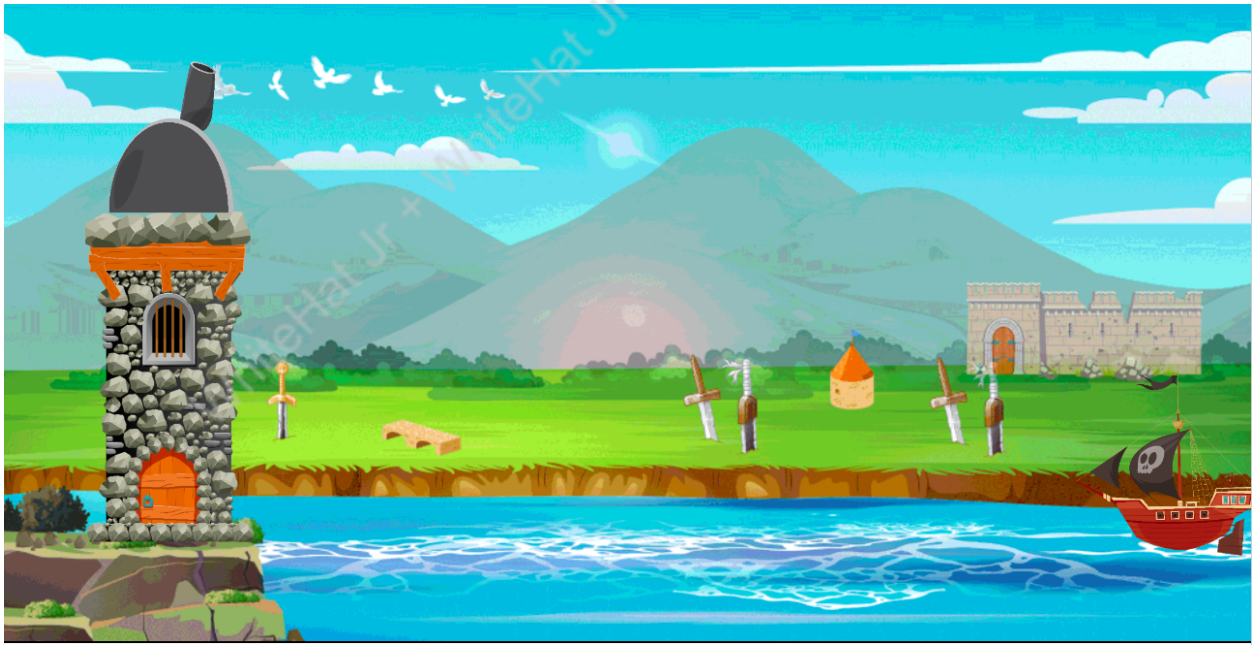
```
display() {
  var pos = this.body.position;

  push();
  translate(pos.x, pos.y);
  imageMode(CENTER);
  image(this.image, 0, this.boatPosition, this.width, this.height);
  pop();
}
```

2. Use the Boat class to create a new boat and display it in the **display()** function.

```
function setup() {
  canvas = createCanvas(1200, 600);
  engine = Engine.create();
  world = engine.world;
  angle = -PI / 4;

  ground = Bodies.rectangle(0, height - 1, width * 2, 1, { isStatic: true });
  World.add(world, ground);

  tower = Bodies.rectangle(160, 350, 160, 310, { isStatic: true });
  World.add(world, tower);

  cannon = new Cannon(180, 110, 130, 100, angle);
  boat = new Boat(width-79, height - 60, 170, 170,-80);
}
```

```
boat.display()
```

Output:



3. Create a new boat object in **Sketch.js.**

```
function setup() {
  canvas = createCanvas(windowWidth - 200, windowHeight - 150);
  engine = Engine.create();
  world = engine.world;
  angle = -PI / 4;
  ground = new Ground(0, height - 1, width * 2, 1);
  tower = new Tower(width / 2 - 650, height - 290, 250, 580);
  cannon = new Cannon(width / 2 - 600, height / 2 - 220, 120, 40, angle);

  boat = new Boat(width, height - 100, 200, 200, -100);

}
```

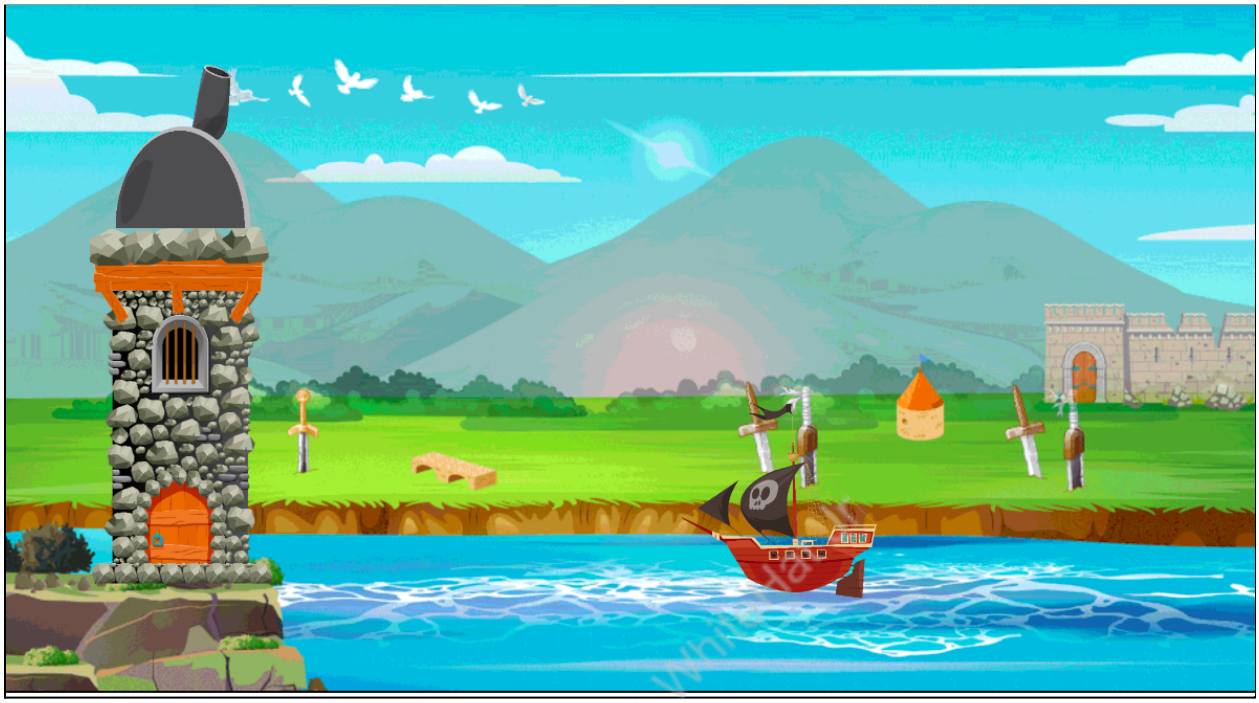
- Display the boat Inside function **draw().**

```
boat.display()
```

4. Give velocity to the boat using **Matter.Body.setVelocity()** to travel towards **Tower.**

```
Matter.Body.setVelocity(boat.body,{x:-0.9, y:0})
boat.display()
```

OUTPUT:



5. Create multiple boats via an empty boats array .

```
var canvas, angle, tower, ground, cannon,
var balls = [];
var boats = [];
```

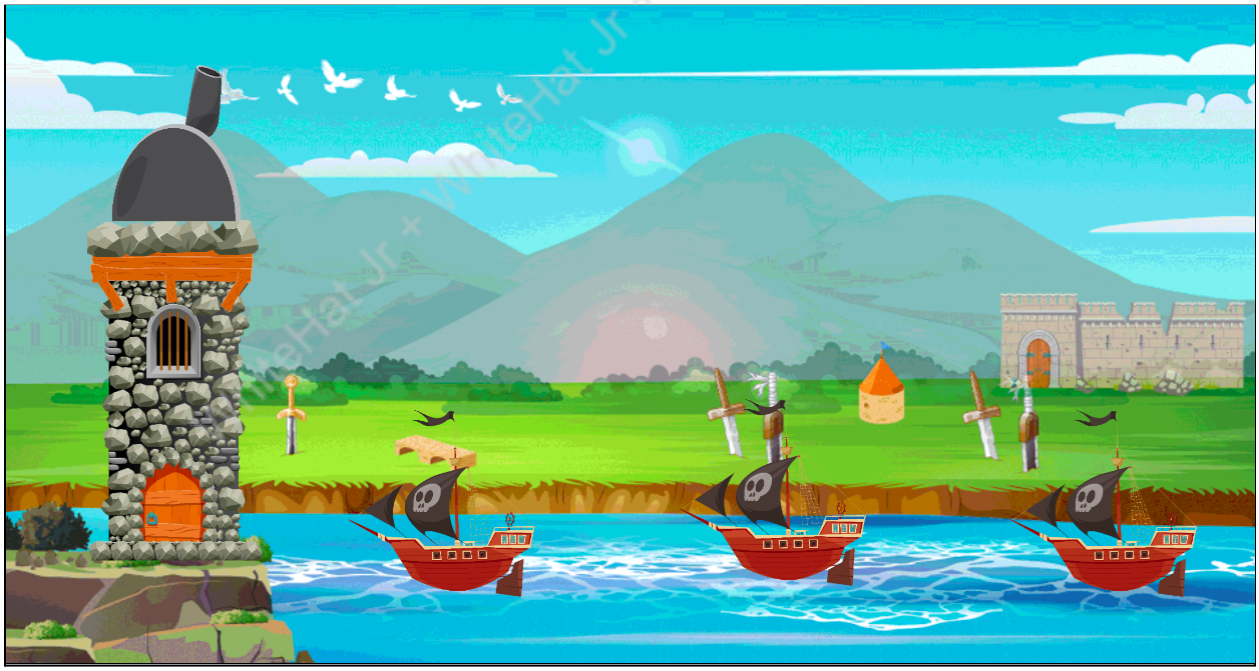
6. Create multiple boats using **showBoats()** same as that we did to create multiple cannonballs in **sketch.js**.

```
function showBoats() {
  if (boats.length > 0) {
    if (
      boats[boats.length - 1] === undefined ||
      boats[boats.length - 1].body.position.x < width - 300
    ) {
      var positions = [-40, -60, -70, -20];
      var position = random(positions);
      var boat = new Boat(width, height - 100, 170, 170, position);

      boats.push(boat);
    }

    for (var i = 0; i < boats.length; i++) {
      if (boats[i]) {
        Matter.Body.setVelocity(boats[i].body, {
          x: -0.9,
          y: 0
        });

        boats[i].display();
      }
    }
  } else {
    var boat = new Boat(width, height - 60, 170, 170, -60);
    boats.push(boat);
  }
}
```


7. Call function **showBoats()** in the function **draw().**

```
push();
imageMode(CENTER);
image(towerImage,tower.position.x, tower.position.y, 160, 310);
pop();


showBoats();

for (var i = 0; i < balls.length; i++) {
  showCannonBalls(balls[i]);
}

cannon.display();
```

OUTPUT:



What's next?

In the next class, we'll learn to create multiple boats.

EXTEND YOUR KNOWLEDGE

1. Bookmark the following link to know more about setting velocity to physics engine bodies: https://brm.io/matter-js/docs/classes/Body.html#method_setVelocity