**Collision:**

we want to check if there was a collision between the boat and the cannonball

We can get all the boats from the boats array and then get all the balls from the array and check if any of the ball touched the boats or not.

We already have a for loop written for **showCannonBalls**() function. Similarly, we’ll create another function called as **collisionWithBoat**(). This function will take only the index number value as a parameter. Inside the collisionWithBoat() function we’ll use a for loop on the length of the boat.

The Matter.js library has a function called **Matter.SAT.collides()**. This function takes 2 parameters and returns true if there was collision between the 2 bodies and false if there is no collision between them.

We’ll declare the collision variable and store the values we get from the Matter.SAT.collides() function in it. Now the collision variable will have either true or false as a value.

When the collision happens, when the cannon ball collides with the boat we need to **remove** the **ball** and **boat** from the **array** and the **world** in order to make them disappear from the canvas.

Now inside the Boat.js file we’ll write a **remove function** which will remove the boat from the world and from the array. This function will take the **index** of the boat to be removed as the parameter. Inside the function we’ll

use **Matter.World.remove()** to remove the boat from the world and

use **delete** method to delete the boat from the array.

We’ll add this code inside the **setTimeout**() function to execute the code after 2 seconds; setTimout() function executes a code after a certain time interval

We’ll write the similar remove() function inside the cannonball class as we also need to remove the cannonball from the screen.

Inside this condition we’ll call the boats[i].remove[i] function and call the Matter.World.remove() and delete balls[index] to remove balls from world and the array.

Finally call the collisionWithBoat() function along with the showCannonBalls() and pass ‘i’ for the index.

Inside the showCannonBalls() function, we’ll write a condition to check if the X position of the cannonball is more than or equal to the width of the screen or if the y position is greater than the height-50 which means little above the ground. If either of the two conditions is true then we’ll remove the ball using the remove function that we had written earlier.