Topic	COLLISION DETECTION	
Class Description	**Students will also learn to detect the collision between boats and cannonballs. Students will also learn about nested loops and nested if conditions.**	
Class	**C26**	
Class time	**45 mins**	
Goal	• Detect collision between the cannonball and boat. • Write nested if conditions for collision algorithm.	

Resources Required	• Teacher Resources:
	  ○ Laptop with internet connectivity
	  ○ Visual Studio Code
	  ○ Earphones with mic
	  ○ Notebook and pen
	• Student Resources:
	  ○ Laptop with internet connectivity
	  ○ Visual Studio Code
	  ○ Earphones with mic
	  ○ Notebook and pen	

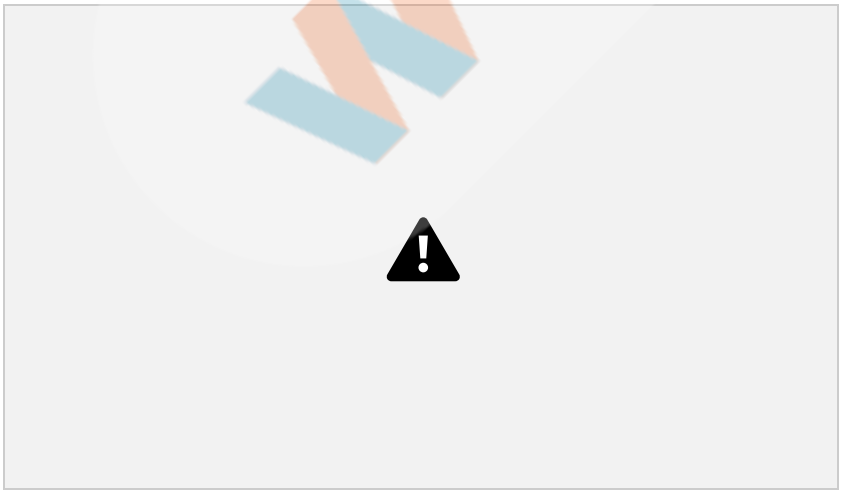
Class structure	**WARM-UP** **Teacher-led Activity** **Student-led Activity** **WRAP-UP**	**5 mins** **10 mins** **25 mins** **5 mins**

WARM-UP SESSION - 5 mins

Teacher starts slideshow  **from slides 1 to 9**

Refer to speaker notes and follow the instructions on each slide.

© 2020 - WhiteHat Education Technology Private Limited.
Note: This document is the original copyright of WhiteHat Education Technology Private Limited.
Please don't share, download or copy this file without permission.

Activity details	Solution/Guidelines
Hey <student's name>. How are you? It's great to see you! Are you excited to learn something new today? **Run the presentation from slide 1 to 4** **Following are the WARM-UP session deliverables:** • Connect the student • Previous class activity revision • WARM-UP Quiz Session	**ESR**: Hi, thanks; yes I am excited about it. Click on the slide show tab and present the slides.
QnA Session	
Question	**Answer**
Select the correct option to make the **board** body static in nature. A. isStatic : false B. isStatic : true C. Static : true D. isStatic = true	**B**
Select the correct option to create an **arrow** when the space key is pressed and only if the number of arrows is greater than '0'. Also decrease the **numberOfArrows** by 1 every time. 	**C**

© 2020 - WhiteHat Education Technology Private Limited.
Note: This document is the original copyright of WhiteHat Education Technology Private Limited.
Please don't share, download or copy this file without permission.

A.

```
/* if (keyCode === 23) {
    if (numberOfArrows > 0) {
      var posX = playerArcher.body.position.x;
      var posY = playerArcher.body.position.y;
      var angle = playerArcher.body.angle;
      var arrow = new PlayerArrow(posX, posY, 100, 10, angle);
      Matter.Body.setAngle(arrow.body, angle);
      playerArrows.push(arrow);
      numberOfArrows -= 1;
    }
} */
```

B.

```
/* if (keyCode === 32) {
    if (numberOfArrows > 0) {
     var posX = playerArcher.body.position.x;
     var posY = playerArcher.body.position.y;
     var angle = playerArcher.body.angle;
     var arrow = new PlayerArrow(posX, posY, 100, 10, angle);
     Matter.Body.setAngle(arrow.body, angle);
     playerArrows.push(arrow);
     numberOfArrows += 1;
    }
}*/
```

C.

```
/* if (keyCode === 32) {
    if (numberOfArrows > 0) {
      var posX = playerArcher.body.position.x;
      var posY = playerArcher.body.position.y;
      var angle = playerArcher.body.angle;
      var arrow = new PlayerArrow(posX, posY, 100, 10, angle);
      Matter.Body.setAngle(arrow.body, angle);
      playerArrows.push(arrow);
      numberOfArrows -= 1;
    }
} */
```

D.

```
/* if (keyCode === 32) {
    if (numberOfArrows > 0) {
     var posX = playerArcher.body.position.x;
     var posY = playerArcher.body.position.y;
     var angle = playerArcher.body.angle;
     var arrow = new PlayerArrow(posX, posY, 100, 10, angle);
     Matter.Body.setAngle(arrow.body, angle);
     playerArrows.push(arrow);
     numberOfArrows *= 1;
    }
}*/
```

Continue the WARM-UP session

Activity details	Solution/Guidelines

© 2020 - WhiteHat Education Technology Private Limited.
Note: This document is the original copyright of WhiteHat Education Technology Private Limited.
Please don't share, download or copy this file without permission.

Run the presentation from slide 5 to slide 9 to set the problem statement. **Following are the WARM-UP session deliverables:** • Enemies in the story • Pirates enemies • Moving Boat	Narrate the story by using hand gestures and voice modulation methods to bring in more interest in students.
Teacher ends slideshow 	
TEACHER-LED ACTIVITY - 10 mins	
Teacher Initiates Screen Share	
CHALLENGE • **Learn about the nested loops.** • **Use Matter.SAT.collides() function to detect collision.**	
Teacher Action	**Student Action**
Teacher downloads the code from Teacher Activity 1. Can you quickly tell me what we did in the last class?	**ESR:** • In the last class, we created a boat class. • Using this boat class we created a boat and added velocity to it. • We also wrote a **showBoats()** function using which we created many boats.

© 2020 - WhiteHat Education Technology Private Limited.
Note: This document is the original copyright of WhiteHat Education Technology Private Limited.
Please don't share, download or copy this file without permission.

Awesome! Remember we did the same to create the cannonballs earlier. We wrote a **showCannonball()** function to show the cannonballs. Can you tell me what exactly we do to get all the cannonballs from the ball array? So now we want to check if there was a collision between the boat and the cannonball. How can we do that?	**ESR:** ● We wrote a **for loop**. We iterated the loop on the length of the balls array. ● Got all the ball positions in the **i** variable and then pass this variable to the **showCannonBalls()** . **ESR:** 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.	

© 2020 - WhiteHat Education Technology Private Limited.
Note: This document is the original copyright of WhiteHat Education Technology Private Limited.
Please don't share, download or copy this file without permission.

```
function collisionWithBoat(index) {
  for (var i = 0; i < boats.length; i++) {

    }
}
```

Now we have access to the boats.

We first need to check if the **index** of the cannonball and **i** of boats are defined and not undefined, meaning that there is some value inside those arrays.

We'll write the if condition to do so. This condition will be inside the inner loop.

```
function collisionWithBoat(index) {
  for (var i = 0; i < boats.length; i++) {
    if (balls[index] !== undefined && boats[i] !== undefined) {

    }
  }
}
```

We have both values. Now we need to check if there was a collision between the boat and the cannonball.

Any idea how we can do that?

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.

ESR:
Varied.

© 2020 - WhiteHat Education Technology Private Limited.
Note: This document is the original copyright of WhiteHat Education Technology Private Limited.
Please don't share, download or copy this file without permission.

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.

*Teacher codes to declare the collision variable and use store the values of **Matter.SAT.collides()** function in it.*

```
function collisionWithBoat(index) {
  for (var i = 0; i < boats.length; i++) {
    if (balls[index] !== undefined && boats[i] !== undefined) {
      var collision = Matter.SAT.collides(balls[index].body, boats[i].body);

    }
  }
}
```

Can you tell me what we want to do 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.

ESR:
Varied.

© 2020 - WhiteHat Education Technology Private Limited.
Note: This document is the original copyright of WhiteHat Education Technology Private Limited.
Please don't share, download or copy this file without permission.

This will help later when we'll be adding images.
*Teacher codes to write the **remove()** function which takes index as the parameter.*

```
remove(index) {
  setTimeout(() => {
    Matter.World.remove(world, boats[index].body);
    delete boats[index];
  }, 2000);
}
```

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

remove function inside the **cannonball.js** file

```
remove(index) {
  Matter.Body.setVelocity(this.body, { x: 0, y: 0 });

  setTimeout(() => {
    Matter.World.remove(world, this.body);
    delete balls[index];
  }, 1000);
}
```

We'll use another **if** condition to check if the **collision.collided** is true.

If it is true then,

© 2020 - WhiteHat Education Technology Private Limited.
Note: This document is the original copyright of WhiteHat Education Technology Private Limited.
Please don't share, download or copy this file without permission.

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.

```
function collisionWithBoat(index) {
  for (var i = 0; i < boats.length; i++) {
    if (balls[index] !== undefined && boats[i] !== undefined) {
      var collision = Matter.SAT.collides(balls[index].body, boats[i].body);

      if (collision.collided) {
        boats[i].remove(i);

        Matter.World.remove(world, balls[index].body);
        delete balls[index];
      }
    }
  }
}
```

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

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

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.

© 2020 - WhiteHat Education Technology Private Limited.
Note: This document is the original copyright of WhiteHat Education Technology Private Limited.
Please don't share, download or copy this file without permission.

```
function showCannonBalls(ball, index) {
  if (ball) {
    ball.display();
    if (ball.body.position.x >= width || ball.body.position.y >= height - 50) {
      ball.remove(index);
    }
  }
}
```

So now we have written code to remove the ball and the boat when the collision happens from the array and the world.	
Teacher starts slideshow :Slide 10 to Slide 11	
Run the presentation slide to set the student activity context.	

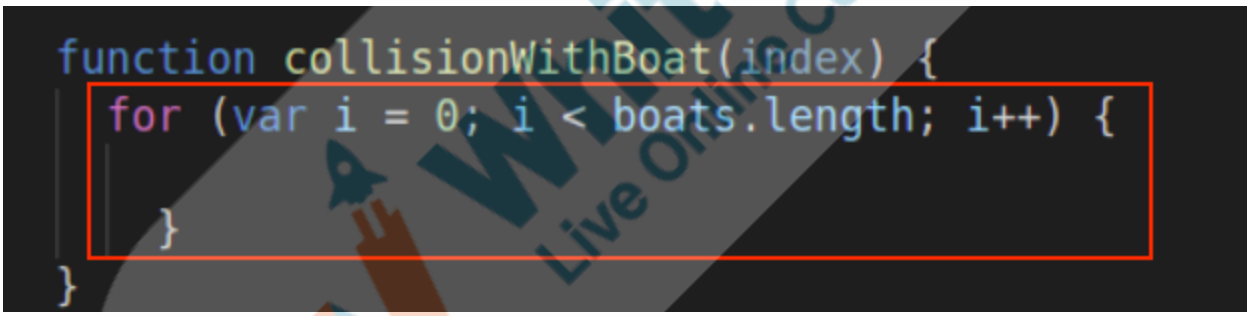
Teacher ends slideshow
Teacher Stops Screen Share
STUDENT-LED ACTIVITY - 25 mins
• **Ask the student to press the ESC key to come back to the panel.** • **Guide the student to start Screen Share.** • **The teacher gets into Fullscreen.**

© 2020 - WhiteHat Education Technology Private Limited.
Note: This document is the original copyright of WhiteHat Education Technology Private Limited.
Please don't share, download or copy this file without permission.

<table>
<tr><td colspan="2"><u>ACTIVITY</u>

• Detect collisions between cannonballs and the boats.</td></tr>
<tr><td>Teacher Action</td><td>Student Action</td></tr>
<tr><td>Guide the student to download the code from Student Activity 1
Guide the student to create a collisionWithBoat() function.</td><td>Student downloads code from Student Activity 1

student codes to create a collisionWithBoat() function.</td></tr>
<tr><td colspan="2">Write a for loop to get all the boats in the collisionWithBoat().

 
</td></tr>
<tr><td>Now we need to get all the boats from the array.

So we'll check if the value inside the boats array is not undefined using the if condition.

Guide the student to write the if condition.</td><td>The student codes to write an if condition.</td></tr>
<tr><td colspan="2">Create an if statement in collisionWithBoat() function.</td></tr>
</table>

© 2020 - WhiteHat Education Technology Private Limited.
Note: This document is the original copyright of WhiteHat Education Technology Private Limited.
Please don't share, download or copy this file without permission.

```
function collisionWithBoat(index) {
  for (var i = 0; i < boats.length; i++) {
    if (balls[index] !== undefined && boats[i] !== undefined) {

    }
  }
}
```

Before we check the collision, let's write a function **remove()** to remove the boats from the world and from the boats array. *Guide the student to code for the remove function inside the Boat.js file.*	*The student codes to write the remove function to remove boats from the world and array*

Use delete boats[index] to remove boats from array and world in Boat.js;

```
remove(index) {
  setTimeout(() => {
    Matter.World.remove(world, boats[index].body);
    delete boats[index];
  }, 2000);
}
```

Now we'll check if the collision is happening between the ball and boat and if they are colliding, then we want to remove them from the world and the array. Matter.js has a function called as **Matter.SAT.collides()**. This function takes the bodies on which we want to check the collision as parameters.	*The student codes to use the Matter.SAT.collides() function to detect the collision.* *Student can refer to Student Activity 2.*

© 2020 - WhiteHat Education Technology Private Limited.
Note: This document is the original copyright of WhiteHat Education Technology Private Limited.
Please don't share, download or copy this file without permission.

Create a for loop to detect collision file in collisionWithBoat() function of sketch.js .

```
function collisionWithBoat(index) {
  for (var i = 0; i < boats.length; i++) {
    if (balls[index] !== undefined && boats[i] !== undefined) {
      var collision = Matter.SAT.collides(balls[index].body, boats[i].body);

      if (collision.collided) {
        boats[i].remove(i);

        Matter.World.remove(world, balls[index].body);
        delete balls[index];
      }
    }
  }
}
```

Create a **remove()** function to remove cannonballs from the world and the array.

```
remove(index) {
  Matter.Body.setVelocity(this.body, { x: 0, y: 0 });

  setTimeout(() => {
    Matter.World.remove(world, this.body);
    delete balls[index];
  }, 1000);
}
```

Also, add a condition in the **showCannonBalls()** function to remove the cannonballs.

© 2020 - WhiteHat Education Technology Private Limited.
Note: This document is the original copyright of WhiteHat Education Technology Private Limited.
Please don't share, download or copy this file without permission.

```
function showCannonBalls(ball, index) {
  if (ball) {
    ball.display();
    if (ball.body.position.x >= width || ball.body.position.y >= height - 50) {
      ball.remove(index);
    }
  }
}
```

Awesome job. Now our game is almost complete.	

Teacher Guides Student to Stop Screen Share	
WRAP-UP SESSION - 5 Mins	
Teacher starts slideshow  **from slide 12 to 22**	
Activity details	**Solution/Guidelines**
Run the presentation from slide 12 to slide 22 **Following are the WARM-UP session deliverables:** • **Explain the facts and trivia** • **Next class challenge** • **Project for the day** • **Additional Activity**	Guide the student to develop the project and share with us
Quiz time - Click on in-class quiz	
Question	**Answer**
What is a nested loop? A. Nested loop is a loop inside a loop.	**A**

© 2020 - WhiteHat Education Technology Private Limited.
Note: This document is the original copyright of WhiteHat Education Technology Private Limited.
Please don't share, download or copy this file without permission.

B. Nested loop is a new type of for loop. C. Nested loop is a new type of while loop. D. Nested loops are two loops written side by side	

What does the following snippet of code do?
`//to remove ships from the world`
`remove(index) {`
`Matter.World.remove(world, boats[index].body);`
`delete boats[i]`
`}`
A.It only removes the boat body from the world.
B.It only removes the boat body from the array.
C.It removes the boats from the world and from the boat's array as well.
D. It removes boats from the game.
	C

Which function is used to detect the collision between two bodies? A.Matter.collides() B.Matter.SAT.collides() C.Matter.SAT.collided() D.Matter.SAT.collision()	**B**
End the quiz panel	
Teacher ends slideshow 	
FEEDBACK • **Appreciate the student for their efforts in the class.** • **Ask the student to make notes for the reflection journal along with the code they wrote in today's class.**	

Step 4: Wrap-Up (5 mins)	You get Hats Off for your excellent work!	*Make sure you have given at least 2 Hats Off during the class for:*

© 2020 - WhiteHat Education Technology Private Limited.
Note: This document is the original copyright of WhiteHat Education Technology Private Limited.
Please don't share, download or copy this file without permission.

	In the next class, let's add some proper animations to the boats to feel like traveling on water and adding sounds to make the game more interesting. We will be using the PISKEL app. This is only supported by Chrome, Firefox and Edge browsers, so, make sure to login using any of these browsers.	
*** This Project will take only 30 mins to complete. Motivate students to try and finish it immediately after the class.*** **PROJECT OVERVIEW** **EPIC ARCHERY STAGE 5** **Goal of the Project:** In Class 26, you learned to detect the collision between the cannonball and the boat using the Matter.SAT.collides(). You also learned to remove the boat and the ball from the game after the collision. *This is a continuation of Project 22, 23, 24 & 25. Make sure to complete that work before attempting this one.		**Note: You can assign the project to the student in class itself by clicking on the Assign Project button which is available under the projects tab.** *Students engage with the teacher over the project.*

© 2020 - WhiteHat Education Technology Private Limited.
Note: This document is the original copyright of WhiteHat Education Technology Private Limited.
Please don't share, download or copy this file without permission.

Story:
Archery is one of the oldest arts which is still practiced. After reading the information about Archery in a book, your friend Georgie wants to play Archery. To give him a virtual experience, you want to use your coding expertise and physics engine concepts to create an Archery game for him.

Detect the collision between the arrow and the targets. Remove the arrows if the collision happens.

I am very excited to see your project solution and I know you will do really well.

Bye Bye!

Teacher ends slideshow 

Teacher Clicks ✖ End Class

ADDITIONAL ACTIVITY

Additional Activities
Encourage the student to write reflection notes in their reflection journal using Markdown.

Use these as guiding questions:

- What happened today?
  - Describe what happened.
  - The code I wrote.

The student uses the Markdown editor to write their reflections in a reflection journal.

© 2020 - WhiteHat Education Technology Private Limited.
Note: This document is the original copyright of WhiteHat Education Technology Private Limited.
Please don't share, download or copy this file without permission.

• How did I feel after the class? • What have I learned about programming and developing games? • What aspects of the class helped me? What did I find difficult?	

Activity	Activity Name	Links
Teacher Activity 1	Boilerplate Class	https://github.com/pro-whitehatjr/PRO-C25-Reference
Teacher Activity 2	Teacher Reference	https://github.com/pro-whitehatjr/PRO-C26-Reference
Student Activity 1	Boilerplate code.	https://github.com/pro-whitehatjr/PRO-C25-Reference
Student Activity 2	Matter.SAT.Collide reference	https://brm.io/matter-js/docs/classes/SAT.html
Teacher Reference visual aid link	Visual aid link	https://curriculum.whitehatjr.com/Visual+Project+Asset/PRO_VD/PRO+V3+-C26-+withcues.html
Teacher Reference In-class quiz	In-class quiz	https://s3-whjr-curriculum-uploads.whjr.online/7689df5c-0112-442d-8226-76e7545653e1.pdf
Project Solution	Epic Archery Stage 5	https://github.com/pro-whitehatjr/V3_Project_solution_C26

© 2020 - WhiteHat Education Technology Private Limited.
Note: This document is the original copyright of WhiteHat Education Technology Private Limited.
Please don't share, download or copy this file without permission.