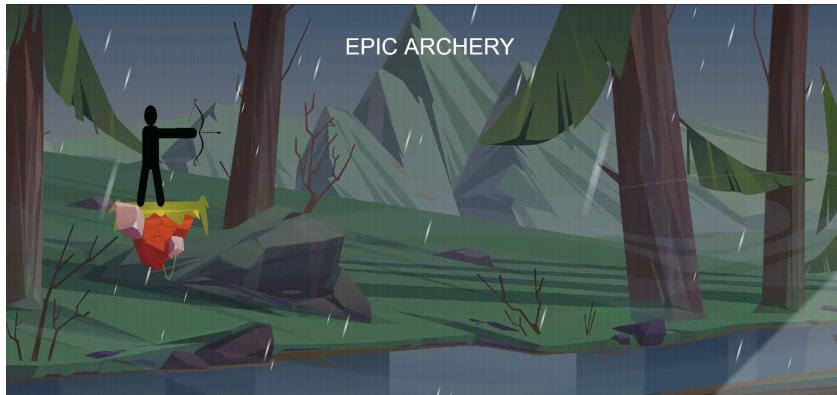


Topic	CREATING BOATS				
Class Description	<b>Students will learn to create a Boat class to create multiple boat objects using an array.</b>				
Class	<b>C25</b>				
Class time	<b>45 mins</b>				
Goal	<ul style="list-style-type: none"> <li>● Create a new boat class.</li> <li>● Create multiple boat objects using the boat class.</li> <li>● Detect collision between the cannonball and boat.</li> </ul>				
Resources Required	<ul style="list-style-type: none"> <li>● Teacher Resources: <ul style="list-style-type: none"> <li>○ Laptop with internet connectivity</li> <li>○ Visual Studio Code</li> <li>○ Earphones with mic</li> <li>○ Notebook and pen</li> </ul> </li> <li>● Student Resources: <ul style="list-style-type: none"> <li>○ Laptop with internet connectivity</li> <li>○ Visual Studio Code</li> <li>○ Earphones with mic</li> <li>○ Notebook and pen</li> </ul> </li> </ul>				
Class structure	<b>WARM-UP</b> <b>Teacher-led Activity</b> <b>Student-led Activity</b> <b>WRAP-UP</b>		<b>5 mins</b> <b>10 mins</b> <b>25 mins</b> <b>5 mins</b>		
<b>WARM-UP SESSION - 5 mins</b>					
 <p><b>Teacher starts slideshow</b> from slides 1 to 8  Refer to speaker notes and follow the instructions on each slide.</p>					

Activity details	Solution/Guidelines
<p><i>Hey &lt;student's name&gt;. How are you? It's great to see you! Are you excited to learn something new today?</i></p> <p><b>Run the presentation from slide 1 to 4</b>  <b>Following are the WARM-UP session deliverables:</b></p> <ul style="list-style-type: none"> <li>• Connect the student</li> <li>• Previous class activity revision</li> <li>• WARM-UP Quiz Session</li> </ul>	<p><b>ESR:</b> Hi, thanks; yes I am excited about it.</p> <p>Click on the slide show tab and present the slides.</p>
<b>QnA Session</b>	
Question	Answer
What will the following code block do?	D
<pre>if (keyIsDown(UP_ARROW) &amp;&amp; angle &gt; -103){     angle -=1;     Matter.Body.setAngle(this.body, angle); }</pre>	<p>A. It will set an angle for the body when the right arrow key is pressed.  B. It will set an angle for the body when the left arrow key is pressed.  C. It will set an angle for the body when the down arrow key is pressed.  D. It will set an angle for the body when the up arrow key is pressed.</p>
Select the correct block of code to call display( ) function to display each arrow.	C



A.

```
for (var i; i < playerArrows.length; i++) {
    if (playerArrows[i] !== undefined) {
        playerArrows[i].display();
    }
}
```

B.

```
for (var i = 0, i++) {
    if (playerArrows[i] !== undefined) {
        playerArrows[i].display();
    }
}
```

C.

```
for (var i = 0; i < playerArrows.length; i++) {
    if (playerArrows[i] !== undefined) {
        playerArrows[i].display();
    }
}
```

D.

```
for (var i) {
    if (playerArrows[i] !== undefined) {
        playerArrows[i].display();
    }
}
```

**Continue the WARM-UP session**

Activity details	Solution/Guidelines
------------------	---------------------

<p><b>Run the presentation from slide 5 to slide 8 to set the problem statement.</b></p> <p><b>Following are the WARM-UP session deliverables:</b></p> <ul style="list-style-type: none"> <li>• Enemies in the story</li> <li>• Pirates enemies</li> <li>• Moving Boat</li> </ul>	<p>Narrate the story by using hand gestures and voice modulation methods to bring in more interest in students.</p>
<p></p> <p><b>Teacher ends slideshow</b></p>	
<p><b>TEACHER-LED ACTIVITY - 10 mins</b></p>	
<p><b>Teacher Initiates Screen Share</b></p>	
<p><b><u>CHALLENGE</u></b></p> <ul style="list-style-type: none"> <li>• Create a boat class.</li> <li>• Create a boat object using the boat class and add movement to the boat.</li> </ul>	
Teacher Action	Student Action
<p>You must have played different kinds of games.</p> <p>Every game has some kind of villainous character that we need to defeat in order to win.</p> <p>Can you tell me what all different kinds of enemies have you seen in the games?</p>	<p><b>ESR:</b> <i>The student talks about the games he/she has played.</i></p>
<p>In our case, can you tell what those characters are?</p>	<p><b>ESR:</b> <i>Student talks about the different game enemies he/she has seen.</i></p> <p><b>ESR:</b> It should be the pirates.</p>

Yes! So today we'll be creating the enemy characters for our game.

Let us quickly review the code we wrote from the last class.

*Teacher downloads the template code from [Teacher Activity 1](#) and runs it in the VS code Editor.*

What is the first thing that we need to do to create the enemy?

In our game, the enemies are going to be the pirates and the pirates travel the sea on their boats so we'll create the pirate ships/boats.

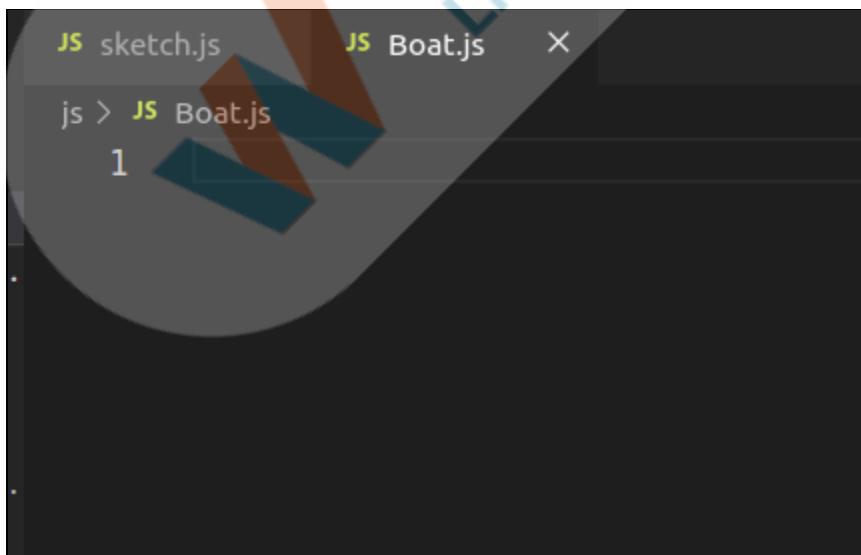
Let's first create a **Boat.js** file and add it to the **index.html** file.

*Student reads through the code and explains what each block of code does.*

**ESR:**

We'll first need to create the class.

1. Create a new file **Boat.js**



2. Add in **index.html**

```
<script src=".js/Cannon.js"></script>
<script src=".js/CannonBall.js"></script>
<script src=".js/Boat.js"></script>
```

First, we'll create the **constructor()**. In the **constructor()** we'll pass the values **x, y positions, height and width**.

Here we'll add one more variable and that is the **boatPos** which will store the position of the boat.

We'll then create the rectangle and pass the x, y, height and width to it. We'll get the boat position and store it in **this.boatPos**. We are using **this.boatPos** variable to get the random positions of the boat from the code.

Then we also have the boat image in the assets so let's load it in **this.image**.

Finally, we'll add the body to the world.

*<The teacher codes to create the constructor() in Boat.js.>*

*Student observes/  
listens/asks questions if any.*

```
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);
    }
}
```

Now inside the **display()** function we'll first get the angle and the position of the body.

- Then we'll call the **push()** function to save the current drawing style settings.
- Call the translate function and pass the **pos.x** and **pos.y**.
- Call the rotate function and pass the angle to it.
- Set the image mode as the center.

Draw the image using **image()** function and pass the **image, x ,y position, width and height**.

(In the image, **this.boatPosition** would be the y position of the boats.)

**ESR:**

*Student observes and asks questions.*

**this.boatPosition** would have the random **y** position for the boat.

Finally, we'll call **pop()** to reset the drawing styles.

*The student codes to create the **display()** function*

**In the display() function using the push() and pop() in Boat.js**

```
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();  
}
```

Now we have the class ready, let's create a new boat object and display it.

Here we are giving the **boatPos** variable the value of -100 as it's the initial boat.

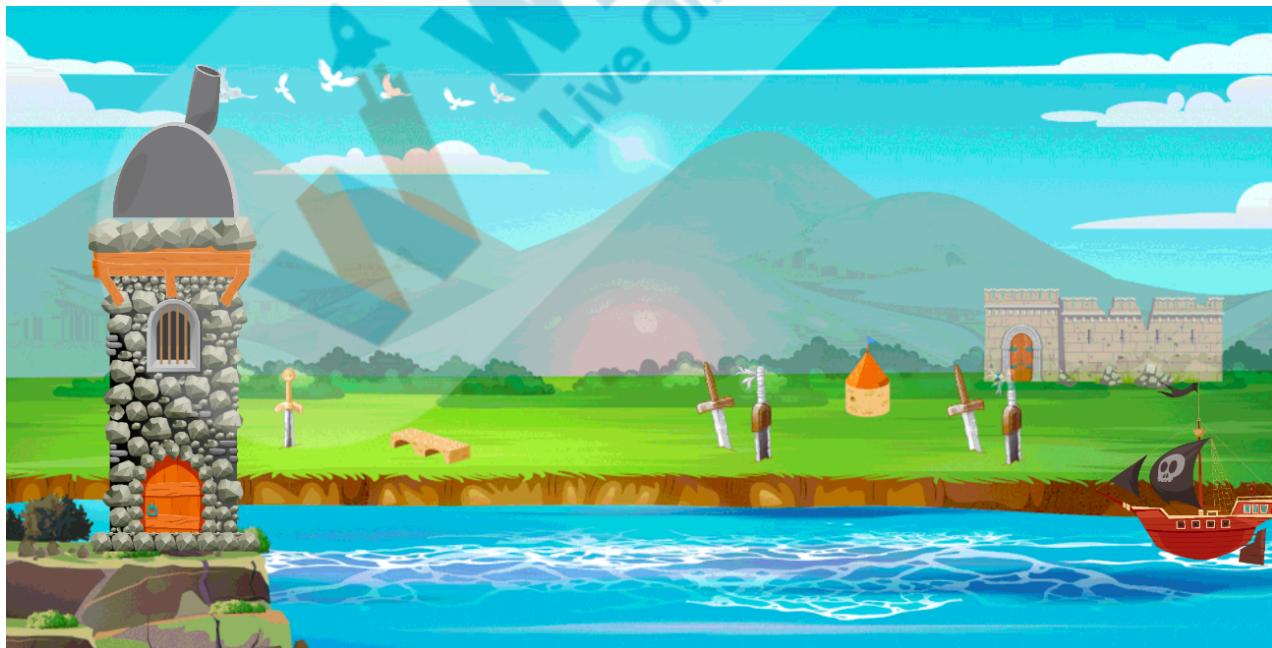
**Create a boat object using Boat class in setup() function of sketch.js**

```
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);  
}  
}
```

Display that object in the draw() function

```
boat.display()
```

Output:



What do we see here?

**ESR:**

The boat is being created at the corner.

What action do we want the boat to perform?

**ESR:**

We want the boat to be moving towards the tower.

Can you tell me how we can do that?

**ESR:** Varied.

**Matter.min.js** library has the function called **Matter.Body.setVelocity()**.

**Matter.Body.velocity()** is a vector that measures the current velocity of the body after the last body is updated. It is read-only.

And **setVelocity()** function allows you to set the velocity on a particular object. This function takes 2 parameters such as the object body and the velocity.

Let's give some velocity to the boat.

*<Teacher codes to set velocity to the boat.>*

**Set velocity for the boat inside function draw() of sketch.js**

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

**Output:**





Now, what do we see?

But just one enemy won't be enough. Won't it be interesting if we can create many such enemies?  
Can you try doing that?

Alright! Let's get you started.

**ESR:**

The boat is traveling towards the tower.

**ESR:** Yes.



Teacher starts slideshow :Slide 9 to Slide 10

Run the presentation slide to set the student activity context.

<p style="text-align: center;"></p> <p><b>Teacher ends slideshow</b></p> <p><b>Teacher Stops Screen Share</b></p> <p><b>STUDENT-LED ACTIVITY - 25 mins</b></p> <ul style="list-style-type: none"> <li>Ask the student to press the ESC key to come back to the panel.</li> <li>Guide the student to start Screen Share.</li> <li>The teacher gets into Fullscreen.</li> </ul> <p style="text-align: center;"><b><u>ACTIVITY</u></b></p> <ul style="list-style-type: none"> <li>Write a function to create multiple boats.</li> </ul>	
Teacher Action	Student Action
<p><i>&lt;The teacher helps the student download the boilerplate code from <a href="#">Student Activity 1</a>.&gt;</i></p> <p>To create multiple boats we'll write a function called <b>showBoats()</b>.</p> <p>This function will help us create multiple boats.</p> <p>At a time how many boats do we want to show on the screen?</p>	<p><i>Student takes code from the <a href="#">Student Activity 1</a>.</i></p> <p><b>ESR:</b> We would want the boats to keep coming on the screen.</p>

We are going to be doing the same thing that we did to create multiple cannonballs.

Let's start with creating an empty boat array where we'll be storing our boat objects so that we can access them later.

*The teacher helps students with the code.*

*The student codes to create the empty boat array.*

Write code in **Sketch.js**

**Creating empty balls and boats array.**

```
6  var canvas, angle, tower, ground, cannon,
7  var balls = [];
8  var boats = [];
9
```

Now let's write the **showBoats()** function.

So we want to create a new boat when there is no boat in the array.

And we only want to create 4 boats at a time on the screen.

For this we'll write an **if-else** condition.

If the boat array has less than 4 or the position of any boat is less than that of the width, then we want to create a new boat to keep playing continuously.

Else we'll create an initial boat to make the if condition true and to keep playing.

*The teacher helps the student with code.*

*Student codes to write the if else condition.*

## Writing if-else condition to create multiple boats in the sketch.js

Here we first have an outer **if-else** condition.

In this condition, we check if there are boats more than 0 in the boats array. Else we create a new boat in the else condition and add it in the boats array.

```
function showBoats() {  
    if (boats.length > 0) {  
  
    } else {  
        var boat = new Boat(width, height - 60, 170, 170, -60);  
        boats.push(boat);  
    }  
}
```

At the start the length of the boats array will always be 0 so the **if** condition won't be satisfied and the code will move to **else** condition creating a new boat.

When this new boat is created the **if** condition will become true and the code inside the brackets will be executed.

Now that we have a boat we want to add velocity to that boat.

So to get the boats from the boats array we'll use a for loop as later on we'll be adding many boats to the boats array.

Inside this loop, we'll use another **if** condition to check whether there is a boat at that index. If there is a boat then using the **Matter.Body.setVelocity()** function set the velocity to that particular boat and display the 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();  
    }  
}
```

Inside the first **if** condition we have another **if** condition which checks if the last element inside the **boats** array is a boat body and not any undefined body OR the position of the boat is lesser than the width-300 , we are using width-300 as we want the next boat to appear only when the previous boat has covered a certain distance.

As there is only one boat in the array, we'll see only one boat on the screen and we want to create the second boat when the first boat covers some distance on the screen so that it looks like the second boat is following the first one.

To do this in the above loop we'll use another **if** condition which will check if the boat on the screen has crossed 300 distance from the total width of the screen. If the boat has crossed the distance then inside that **if** condition we:-

- Declare the **positions** array and in the array have some values which will be the position for the boat to enter the screen.
- Declare a **position** variable and use the **random()** function on the positions array to get a random value from it.
- Declare a **boat** variable and using the boat class create a new boat and pass the position that we got earlier
- Then finally push the boat in the **boats** array.

After this our boat will be created continuously when a boat crosses a width-300 distance.

```
if (boats.length > 0) {
    if (
        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);
    }
}
```

To avoid any unexpected errors or exceptions we'll add another condition to check if the boat is undefined then also create the boat and push in the boats array so that we have a new boat to continue with the game.

```
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);
}
```

Final code will look like this.

```

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);
    }
}
    
```

Now let's call this function in the **draw()** function.

*The teacher helps the student with the code.*

*The Student codes to call  
the function in the draw  
function.*

**Call the `showBoats()` function in the `draw()` function**

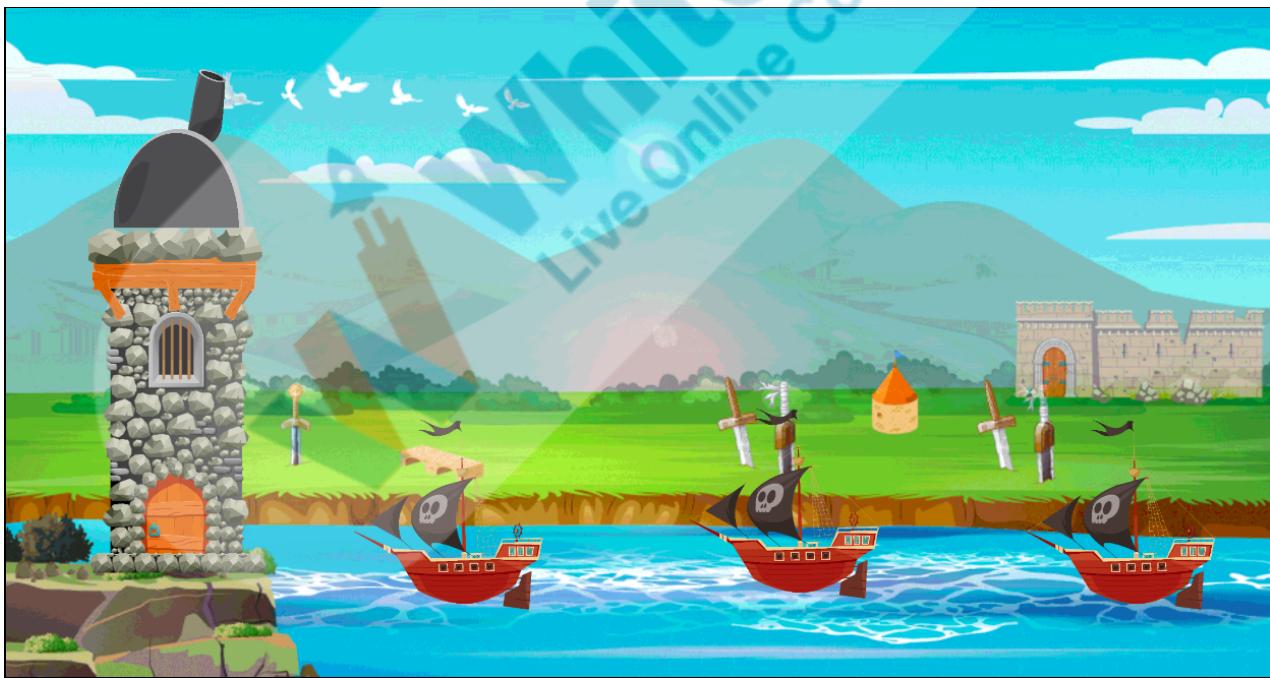
```
push();
imageMode(CENTER);
image(towerImage,tower.position.x, tower.position.y, 160, 310);
pop();

showBoats();

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

cannon.display();
```

OUTPUT:



Now we see that boats are being created at random on the screen.

<p>What happens when we shoot the cannonball at the boat?</p> <p>Yes, so we want to make both the boat and the ball disappear. so we'll be doing this in the next class.</p>	<p><b>ESR:</b></p> <p>When we shoot the cannonball at the boat the cannonball just bounces off the boat and the ball doesn't disappear.</p>
--	---

Teacher Guides Student to Stop Screen Share	
WRAP-UP SESSION - 5 Mins	
 Teacher starts slideshow from slide 11 to 19	
Activity details	Solution/Guidelines
<p><b>Run the presentation from slide 11 to slide 19</b></p> <p><b>Following are the WARM-UP session deliverables:</b></p> <ul style="list-style-type: none"> <li>• Explain the facts and trivia</li> <li>• Next class challenge</li> <li>• Project for the day</li> <li>• Additional Activity</li> </ul>	<p>Guide the student to develop the project and share with us</p>
Quiz time - Click on in-class quiz	
Question	Answer
How do we make the boat move towards the tower?	A

<p>A. By adding velocity in the negative direction.      B. By adding velocity in the positive direction.      C. By adding velocity in an upward direction.      D. By adding velocity in the downward direction.</p>		
<p>Which function is used to set velocity to boat?</p> <p>A.The <b>velocity()</b> function is used to set velocity to the boat.      B.The <b>addVelocity()</b> function is used to set velocity to the boat.      C.The <b>setVelocity()</b> function is used to set velocity to the boat.      D. None of the above functions can be used to set velocity to the boat.</p>	C	
<p>Which is the use of the <b>push()</b></p> <p>A. The use of the <b>push()</b> is to create a new array.      B. The use of the <b>push()</b> is to add elements at the end of the array.      C. The use of the <b>push()</b> is to add elements at the start of the array.      D. The use of the <b>push()</b> is to add elements at the center of the array.</p>	B	
End the quiz panel		
 <b>Teacher ends slideshow</b>		
<b><u>FEEDBACK</u></b> <ul style="list-style-type: none"> <li>• Appreciate the student for their efforts in the class.</li> <li>• Ask the student to make notes for the reflection journal along with the code they wrote in today's class.</li> </ul>		
<b>Step 4: Wrap-Up (5 mins)</b>	You get Hats Off for your excellent work!	<i>Make sure you have given at least 2 Hats Off during the class for:</i>

	<p>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.</p> <p>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.</p>	<div style="background-color: #0072BD; color: white; padding: 5px; border-radius: 10px; text-align: center;">  +10 Creatively Solved Activities         </div> <div style="background-color: #0072BD; color: white; padding: 5px; border-radius: 10px; text-align: center;">  +10 Great Question         </div> <div style="background-color: #0072BD; color: white; padding: 5px; border-radius: 10px; text-align: center;">  +10 Strong Concentration         </div>
<p>* This Project will take only 30 mins to complete. Motivate students to try and finish it immediately after the class.*</p> <p><b>PROJECT OVERVIEW</b></p> <p><b>EPIC ARCHERY STAGE 4</b></p> <p><b>Goal of the Project:</b></p> <p>In Class 25, you learned to create the <b>Boat</b> class and using this <b>Boat</b> class you've created multiple boats. In this project, you'll be using what you learned in the class to create the target board for the archer to shoot.</p> <p>* This is a continuation of Projects 22, 23, &amp; 24. Make sure to complete that work before attempting this one.</p>	<p><b>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.</b></p> <p><i>Students engage with the teacher over the project.</i></p>	

### 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.

Create a target board for the player to shoot the arrow.

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.
- How did I feel after the class?
- What have I learned about programming and developing games?

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

- What aspects of the class helped me? What did I find difficult?

Activity	Activity Name	Links
Teacher Activity 1	Previous Class Code	<a href="https://github.com/pro-whitehatjr/PRO-C24-Reference_code">https://github.com/pro-whitehatjr/PRO-C24-Reference_code</a>
Teacher Activity 2	Teacher Reference	<a href="https://github.com/pro-whitehatjr/PRO-C25-Reference">https://github.com/pro-whitehatjr/PRO-C25-Reference</a>
Student Activity 1	Boilerplate code.	<a href="https://github.com/pro-whitehatjr/PRO-C25-SA">https://github.com/pro-whitehatjr/PRO-C25-SA</a>
Student Activity 2	Matter.SAT.Collide reference	<a href="https://brm.io/matter-js/docs/classes/SAT.html">https://brm.io/matter-js/docs/classes/SAT.html</a>
Teacher Reference Visual Aid Link	Visual Aid Link	<a href="https://s3-whjr-curriculum-uploads.whjr.online/753010bd-697c-4e1f-bebe-f3244351f30c.html">https://s3-whjr-curriculum-uploads.whjr.online/753010bd-697c-4e1f-bebe-f3244351f30c.html</a>
Teacher Reference In-class quiz	In-class quiz	<a href="https://s3-whjr-curriculum-uploads.whjr.online/bd04b71b-3d7b-4aa6-a759-d5366e8b64b4.pdf">https://s3-whjr-curriculum-uploads.whjr.online/bd04b71b-3d7b-4aa6-a759-d5366e8b64b4.pdf</a>
Project Solution	Epic Archery Stage 4	<a href="https://github.com/pro-whitehatjr/PRO-C25-Project-solution">https://github.com/pro-whitehatjr/PRO-C25-Project-solution</a>