Quick notes to understand changes in Simplified version:

- Ground and Tower classes have been replaced by respective physics bodies.
- The arc of the cannon has been replaced by a rectangle body and overlaid by an image.
- Unit of measurement for Cannon's angle of rotation has been changed from radians to degrees.
- Trajectory tracking of cannonballs has been moved to additional activity.
- Creating a sprite sheet from scratch has been moved to additional activity and cleaned JSON has been provided as boilerplate for students' ease.
- Lastly, improved explanations at some places in the LP for the teacher's understanding.

Topic	**SOUNDS AND GAME OVER**
Class Description	**Students will learn to make the game more interesting by adding the sounds. Students will also code to add the game over and scoring functionality to the game.**
Class	**C28**
Class time	**45 mins**
Goal	• Add sounds to the game. • Write the game over and scoring functionality,
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 15 min 20 mins 5 min

WARM-UP SESSION - 5 mins

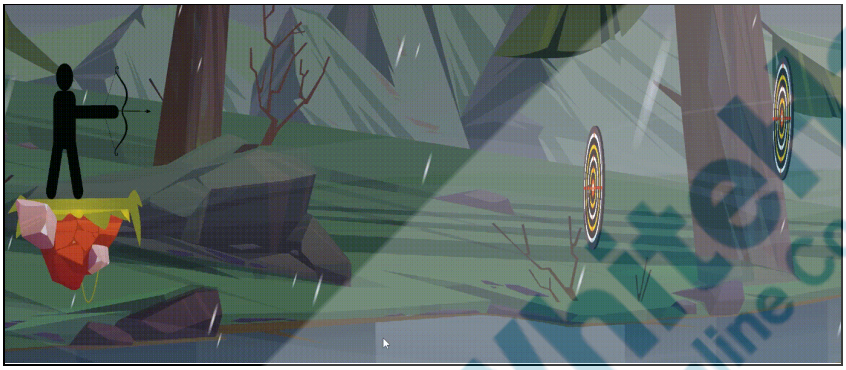
Teacher starts slideshow  **from slides 1 to 8**

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

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 slide 3.*** **The following are the warm-up session deliverables:** • **Greet the student.** • **Revision of previous class activity.** • **Quizzes**	**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 push the position for **trajectory** if the x velocity of the body is greater than 0 and the x position of the body is greater than 400. A. `/*if (this.body.velocity.x > 0 && this.body.position.x > 400) { var position = [this.body.position.x, this.body.position.y]; this.trajectory.push(position); }*/`	**A**

B. ```/*if (this.body.velocity.x > 0 || this.body.position.x > 400) { var position = [this.body.position.x, this.body.position.y]; trajectory.push(position); }*/``` C. ```/*if (this.body.velocity.x < 0 && this.body.position.x < 400) { var position = [this.body.position.x, this.body.position.y]; this.trajectory(position); }*/``` D. ```/*if (this.body.velocity.x > 0 this.body.position.x > 400) { var position = [this.body.position.x, this.body.position.y]; this.trajectory.push(); }*/```	
Select the correct option to create ellipses to display the **trajectory**.  A. `ellipse(this.trajectory[0], this.trajectory[1], 5, 5);` B. `ellipse(trajectory[i][0], trajectory[i][1], 5, 5);` C. `ellipse(this.trajectory[i][0], this.trajectory[i][1], 5, 5);` D. `ellipse(this.trajectory(i)(0), this.trajectory(i)(1), 5, 5);`	**C**

Continue the warm-up session	
Teacher Action	**Student Action**
Run the presentation from slide 4 to slide 8 to set the problem statement. **The following are the warm-up session deliverables:** ● Appreciate the student on his performance in the quizzes.	Narrate the slides by using hand gestures and voice modulation methods to bring in more interest in students.

• Explain how the application of Force will change the scenario.	
Teacher ends slideshow 	
TEACHER-LED ACTIVITY - 15 mins	
CONTEXT • **Review the code from the previous class.** • **Talk about how the sounds have effects on the player.**	
Teacher Initiates Screen Share	
CHALLENGE • **Add the splash animation when the cannonball falls in the water.**	
Teacher Action	**Student Action**
Let me quickly download the code from GitHub. Help me how to do this. *Teacher uses downloads the code from **Teacher Activity 1.***	*Student helps the teacher in cloning the project.*
Well, our game is almost complete now. Can you tell me currently what happens when the boat touches the tower? And what do we want to happen?	**ESR:** When the boat touches the tower nothing happens and the boat. **ESR:**

yes! We want the game to get over when a proper boat touches the tower. And what kind of boat do we want to be touching the tower?	We want the game to get over when the boat touches the tower and show a game over message And add a reload game button. **ESR:** We want a proper boat to touch the tower and not the broken one.
So first let's start by writing the condition to check if the intact boat is touching the tower. We'll be writing this condition in the **showBoats()** function. Inside the first **if** condition, just after where we are pushing the boat in the boats array. First, we'll write a for loop to get all the boats from the boats array and set velocity and animation and using Matter.SAT.collides we'll check the collision between the tower and the boat.	

```
    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();
      boats[i].animate();
      var collision = Matter.SAT.collides(tower, boats[i].body);
    } else {
      boats[i];
    }
  }
} else {
  var boat = new Boat(width, height - 60, 170, 170, -60, boatAnimation);
  boats.push(boat);
}
```

After we have checked for the collision between the boat and the tower we will write another **if condition to check if the boat has touched the tower and is not broken**.

We'll first define a **isGameOver** flag and set it to false.
If the condition is satisfied then we'll create a isGameOver flag and set it to true and call the gameOver() function.

```
var isGameOver = false;
```

```
if (collision.collided && !boats[i].isBroken) {
  isGameOver = true;
  gameOver();
}
```

Here **isBroken** is actually a flag which has the boolean values such as true and false. We'll also need to add this flag in the boat class. In the boat class, we'll create a

this.isBroken and set its value to false as initially the boat is not broken.

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

    this.animation = boatAnimation;
    this.speed = 0.05;
    this.body = Bodies.rectangle(x, y, width, height, options);
    this.width = width;
    this.height = height;

    this.boatPosition = boatPos;
    this.isBroken = false;

    World.add(world, this.body);
  }
  animate() {
    this.speed += 0.05
  }
```

Now let's write the code for the **gameOver()** function.

When the game is over, we want the user to see the game over message and have a reload button which will start the game again for the player to play.

To show this message we'll use a sweet alert which will help us show a popup when the boat touches the tower.

Using a sweet alert is very easy. To use this we first need to import the library into the index.html file.
Sweet alert library is already added to the code.

To use sweet alert we call **swal()** and inside the round brackets we'll create a object with the information that we want to display.

This object will contain:-
Title key with Game Over!!! as its value.
Text key with " Thanks for playing!"
We'll also show an image of the pirate boat as the pirates won; **imageUrl** key with image path as the value
imageSize key with the 150*150 as its value.

Then we'll add a confirm button which will have text as "Play Again".

And when this play button is pressed we'll want to reload the game.

Then using the confirm function we'll call the **location.relaod()** function.

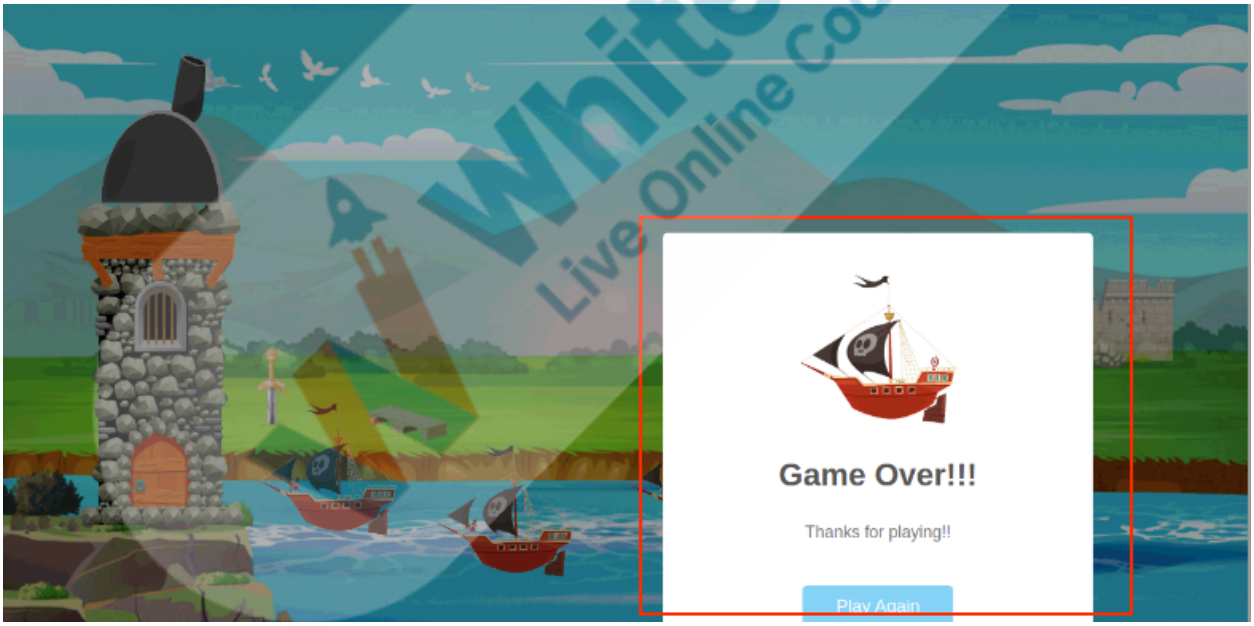
```
<!-- Sweet Alert -->
<script
  src="https://code.jquery.com/jquery-3.5.1.min.js"
  integrity="sha256-9/aliU8dGd2tb6OSsuzixeV4y/faTqgFtohetphbbj0="
  crossorigin="anonymous"
></script>
<script src="./lib/sweetalert.min.js"></script>
<link rel="stylesheet" type="text/css" href="./lib/sweetalert.css" />

<link rel="stylesheet" type="text/css" href="style.css" />
```

```
function gameOver() {
  swal(
    {
      title: `Game Over!!!`,
      text: "Thanks for playing!!",
      imageUrl:
        "https://raw.githubusercontent.com/whitehatjr/PiratesInvasion/main/assets/boat.png",
      imageSize: "150x150",
      confirmButtonText: "Play Again"
    },
    function(isConfirm) {
      if (isConfirm) {
        location.reload();
      }
    }
  );
}
```

Let's test if the code is working properly.

Teacher runs the code and lets the boat touch the tower.



Awesome, now we can reload the game. But our game still doesn't look that fun. How can we make it fun.?

ESR:
We can add the sounds and score to the game

Can you try doing that?	ESR: Yes
alright let's get you started.	

Teacher Stops Screen Share

STUDENT-LED ACTIVITY - 20 mins

	Now it's your turn. Please share your screen with me.	

- **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.**

ACTIVITY

- **Add sounds to the game.**
- **Add the scoring functionality to the game.**

Teacher starts slideshow  **:Slide 9 to Slide 12**

Run the presentation slide to set the student activity context.	

Teacher ends slideshow 

Teacher Action	Student Action
Guide the student to open the previous class code. *Sound files have already been added to the assets.* What are the different sounds we can add to the game?	*Student downloads Student Activity 1.* **ESR:**

<table>
<tr><td>Awesome!</td><td>

- we can add sounds like a background sound.
- We can have a cannon firing sound.
- Water splash sounds when the cannonball touches the water.
- A pirate laugh sound when the boat touches the tower,

</td></tr>
<tr><td>First let’s load the sound files in the preload function.</td><td><The student codes to load the sound files in the preload function.></td></tr>
<tr><td colspan="2">in sketch.js file preload function

```
function preload() {
  backgroundImg = loadImage("./assets/background.gif");
  backgroundMusic = loadSound("./assets/background_music.mp3");
  waterSound = loadSound("./assets/cannon_water.mp3");
  pirateLaughSound = loadSound("./assets/pirate_laugh.mp3");
  cannonExplosion = loadSound("./assets/cannon_explosion.mp3");
```

</td></tr>
<tr><td>We’ll start by playing the background sound.

We’ll play this sound in the draw function.</td><td>Student codes to run the background image.</td></tr>
</table>

in sketch.js file draw function

```
function draw() {
  background(189);
  image(backgroundImg, 0, 0, width, height);

  if (!backgroundMusic.isPlaying()) {
    backgroundMusic.play();
    backgroundMusic.setVolume(0.1);
  }
```

Next is the cannon explosion sound. We'll play this sound when we shoot the cannonball.	*The Student codes to play the cannon explosion sound on the shooting of the cannon.*

In sketch.js file

```
function keyReleased() {
  if (keyCode === DOWN_ARROW ) {
    cannonExplosion.play();
    balls[balls.length - 1].shoot();
  }
}
```

Then we want to have the sound when the cannonball hits the water.	*The student codes to add the sound for water splash*

in sketch,js file

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

Now we want to add the sound when the pirate boat reaches the tower. This will act as an indication that the pirates have captured the village. Which sound should we add here?	ESR: We should add the pirate laughing sound.
So we want the pirate to laugh when the boat reaches the tower. Let us see how it works in the game. *Teacher shares link<Teacher Activity 3> with Student to see the output. Ask student to allow Boat to touch the Tower for Game Over.* Where should we write the code for this?	ESR: We should write this code where we are checking if the boat has collided with the tower.
We want the pirate laugh sound to play just once as the collision condition will become true and the sound will keep playing in the loop which might become a little annoying. To avoid this, what can we do?	ESR: Varied.

We can create a **isLaughing** flag, a simple variable to hold true and false values. This will have a false value by default. If the **isLaughing** flag is false, we'll check if the pirate laughing sound is playing. We'll check if these conditions are false then we'll play the pirate laughing sound. Let's start by creating a **isLaughing** flag.	*<Student codes to create the **isLauging** flag and sets it's value to false.>*
In sketch.js file declaring variable `var isGameOver = false;` `var isLaughing = false;`	
Now, let's write the condition to check if the **isLaughing** flag is false and if the pirate laugh sound is playing.	*Student code to write the condition to check if the **isLaughing** flag is false and the pirate laugh sound is playing.* *If so, change the condition to play the **pirateLaughSound** and set the **isLaughing** flag to true.*
In the showBoats function.	

```
  var collision = Matter.SAT.collides(tower.body, boats[i].body);
  if (collision.collided && !boats[i].isBroken) {
    //Added isLaughing flag and setting isLaughing to true
    if(!isLaughing && !pirateLaughSound.isPlaying()){
     pirateLaughSound.play();
     isLaughing = true
    }
    isGameOver = true;
    gameOver();
  }
}
```

Let's test if the sounds are playing or not . *Teacher asks the student to run the code to check for the sounds.*	*Student runs the code to check for the sounds*
The game won't be interesting unless we have made some score in it. So, let's add the score functionality to the game. First, we'll declare a score variable. We want to increase the score when the cannonball hits the boat and finally display the score.	*The student codes to show the score.*

in the sketch.js file

```
for (var i = 0; i < balls.length; i++) {
  showCannonBalls(balls[i], i);
  for (var j = 0; j < boats.length; j++) {
    if (balls[i] !== undefined && boats[j] !== undefined) {
      var collision = Matter.SAT.collides(balls[i].body, boats[j].body);
      if (collision.collided) {
        if (!boats[j].isBroken && !balls[i].isSink) {
          score += 5;
          boats[j].remove(j);
          j--;
        }
```

in sketch.js file displaying score.

```
fill("#6d4c41");
textSize(40);
text(`Score:${score}`, width - 200, 50);
textAlign(CENTER, CENTER);
```

Awesome job! With this we have completed Pirate Invasion. Did you enjoy working with Physics Engine? Great! We will stop her for now; and will explore more functionalities of the Physics Engine in next class.	ESR: Varied
Teacher Guides Student to Stop Screen Share	
WRAP-UP SESSION - 5 Mins	
Teacher starts slideshow  **from slide 13 to slide 22**	
Activity details	**Solution/Guidelines**
Run the presentation from slide 13 to slide 16. **The following are the warm-up session deliverables:** ● **Appreciate the student.** ● **Revise the current class activities.** ● **Discuss the quizzes.**	Discuss with the student the current class activities and Student will ask doubts related to the activities.
Quiz time - Click on in-class quiz	
Question	**Answer**
Which function is used to add speed to the ball?	**A**

A. speed() B. display() C. animate() D. velocity()	
What does the floor function do? A. It rounds up a number to the nearest greater integer. B. It rounds up a number to the nearest smaller integer. C. It squares the number D. It converts the number to a floating constant.	**B**
Which function is used to load the sound in the game? A. loadsound() B. loadSound() C. play() D. LoadSound()	**B**
End the quiz panel	
Activity details	**Solution/Guidelines**
Run the presentation from slide 17 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
<u>FEEDBACK</u> ● **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 min)	Amazing work! We finished the Pirates Invasion game today. Wasn't that fun?	ESR: Yes, that was so much fun.
	Next class, we'll be working to create a new game "Bunny Catch Fruits" using Physics Engine. How are you feeling?	ESR: Varied.
	You get a hats off. See you in the next class then.	Make sure you have given at least 2 Hats Off during the class for:   
*** 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 7**		**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.**

Goal of the Project:

In Class 28, you added the sounds, game over and scoring functionality to the game. In this project, using similar concepts you need to add score & game over features to this game.

* This is a continuation of Project 22, 23, 24, 25, 26 & 27. Make sure to complete that work before attempting this one.

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.

You have done an excellent job so far. Now, you need to add game over & score to this game to make it more fun and attractive.

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

Bye Bye!

Students engage with the teacher over the project.

Teacher ends slideshow 

Teacher Clicks ✖ End Class

ADDITIONAL ACTIVITIES

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? ● What aspects of the class helped me? What did I find difficult?	*The student uses the Markdown editor to write their reflections in a reflection journal.*

Activity	Activity Name	Links
Teacher Activity 1	Previous class code.	https://github.com/whitehatjr/PiratesInvasionStage-5
Teacher Activity 2	Teacher Reference	https://github.com/whitehatjr/PiratesInvasionStage-6
Student Activity 1	Boilerplate code	https://github.com/whitehatjr/PiratesInvasionSatgae-5.5
Teacher Activity 3	Game Output	https://whitehatjr.github.io/PiratesInvasionStage-6/
Teacher Reference visual aid link	Visual aid link (with cues)	https://s3-whjr-curriculum-uploads.whjr.online/ef26640c-c31e-4d21-a93a-921a19dd9bb9.html
Teacher Reference In-class quiz	In-class quiz	https://s3-whjr-curriculum-uploads.whjr.online/72c533d9-8e30-4f8f-9850-c86c4d845847.pdf

Project Solution	Epic Archery Stage-7	https://github.com/pro-whitehatjr/V3_Project_Solution_C28