Topic	CREATING BLUEPRINTS	
Class Description	**Students will create the layout of the game using the physics engine. Students will learn to create the tower and the cannon class. Students will also learn to add images to the tower and cannon.**	
Class	**C22**	
Class time	**45 mins**	
Goal	• Create classes and objects for ground, cannon and tower. • Create objects for ground, tower & cannon. • Add images for background & Tower.	

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 min** **5 mins**

WARM-UP SESSION - 5 mins

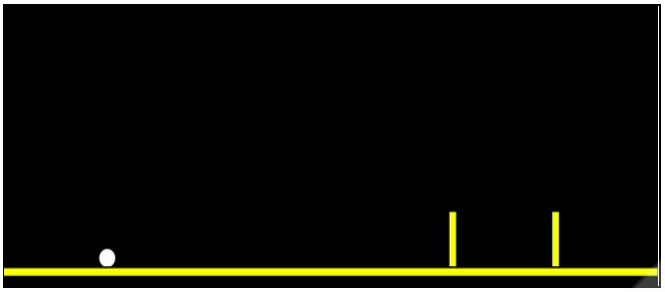
<table>
<tr><td colspan="2"><u>CONTEXT</u>
• Revisit the concept of class and objects and how we can create new objects based on the class blueprint.
• Talk about the different elements in a game.</td></tr>
<tr><td colspan="2">Teacher starts slideshow  from slides 1 to 13
Refer to speaker notes and follow the instructions on each slide.</td></tr>
<tr><td>Activity details</td><td>Solution/Guidelines</td></tr>
<tr><td>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 4

Following are the WARM-UP session deliverables:

• Connect the student
• Previous class activity revision.
• WARM-UP Quiz Session</td><td>ESR: Hi, thanks, Yes I am excited about it!

Click on the slide show tab and present the slides</td></tr>
<tr><td colspan="2">QnA Session</td></tr>
<tr><td>Question</td><td>Answer</td></tr>
<tr><td>Which of the following is the correct line of code to create the right wall of the basket as shown in the image below?
 
A. <code>rightSide = new ground(1350,600,20,120)</code></td><td>A</td></tr>
</table>

B. `rightSide = new ground(150,600,20,120)` C. `rightSide = new ground(1350,100,20,120)` D. `rightSide = new ground(1350,600,20,20)`	
Which of the following option is the correct block to apply force on the ball when the up arrow key is pressed?  A. `ball.applyForce(ball,ball.position,{x:85,y:-85})` B. `Matter.Body.applyForce(ball,ball.position,{x:85,y:-85})` C. `Matter.Body.applyForce(ball.position,{x:85,y:-85})` D. `Matter.Body.applyForce(ball,position,{x:85,y:-85})`	B
Continue the WARM-UP session	
Activity details	**Solution/Guidelines**
Run the presentation from slide 5 to slide 13 to set the problem statement. **Following are the WARM-UP session deliverables:** ● Appreciate the student. ● Explain class and object concepts briefly. ● Set the agenda for the class.	Narrate the story by using hand gestures and voice modulation methods to bring in more interest in students.
Teacher ends slideshow 	
TEACHER-LED ACTIVITY - 15 mins	
Teacher Initiates Screen Share	

CHALLENGE • **Explore code for ground body.** • **Create a tower body**	
Teacher Action	**Student Action**
Teacher downloads the code from **(Teacher Activity 1)** *and opens it on Visual Code Studio.* *Teacher fires up the webserver.*	*Student observes*.
Let's look at the Pirates Invasion game which we will be making. **Teacher Activity 2** What are all the elements in the game that you see?	*The student listens and learns* **ESR:** There is a tower on which the cannon is mounted. Pirate ships are travelling towards the tower. The cannon is shooting the ships. The game ends when the pirate ship reaches the tower.
Yes! Very true. so we'll start by creating the basic elements which are the ground and a tower. Later on, to create multiple boats we'll be using classes. We'll see how to use classes to create those boats and canon balls later on. A blueprint for an object is called a Class. Today, we will be creating the tower and the cannon.	

Let's see how.	
Note:- Code to create ground is given as boilerplate in the sketch.js file. *Teacher needs to explain the code to the student.* The code for the ground body is already present, let's see how it works. Can you tell me how the ground is created? Yes! **Matter.Bodies** is already imported in the **Bodies** variable. **Bodies.rectangle** is used to create the rectangle.	*The student observes and learns.* **ESR:** We can see that the **Matter.Bodies** is used to create the body.
Inside the **sketch.js** file, the **ground** variable is already declared**.** In the setup function using **Bodies.rectangle,** a rectangle has been created and set to the **ground** variable. Can you tell me what the position of the ground is? Yes, the X position of the ground is 0 and the Y position is the “height of the screen -1” so that it comes at the exact bottom. And the width of the ground would be doubled so that the objects remain on the ground and not fall off. As for the height, the height is given as 1 as we don’t want tall ground. As the ground body needs to be static, the static property is passed to the ground body. To add this property a variable called **options** are created.	*The student observes and learns.* **ESR:** The position of the ground is at the bottom of the screen.

In this variable, there is an object with the key **isStatic and true** as its value.
And then this option has been passed to the ground

```
function setup() {

  canvas = createCanvas(1200, 600);
  engine = Engine.create();
  world = engine.world;

  var options = {
    isStatic: true
  }

  ground = Bodies.rectangle(0, height - 1, width * 2, 1, options);
  World.add(world, ground);
}
```

To view this ground, In the **display()** function we'll use the **rect() function** and pass the position x, position y, height and width of the ground.
Here we'll have the width of the ground as 1200 so width*2 has been written and height as 1 as we don't need heightened ground.

```
function draw() {
  backgroundImg(189)
  Engine.update(engine);

  rect(ground.position.x, ground.position.y, width * 2, 1);

}
```

OUTPUT:
The black line at the bottom is ground.

Now we can see the ground at the correct place.

Let's add a background image to the game which will help us to set the theme for the game.

We already have a gif image for the background, let's load the image in the preload function.

And then in the draw function, we'll display it using the **image()** function.

*The teacher declares a variable called **backgroundImg** and loads the image in the **preload** function.*
And then display it using the **image()** function.

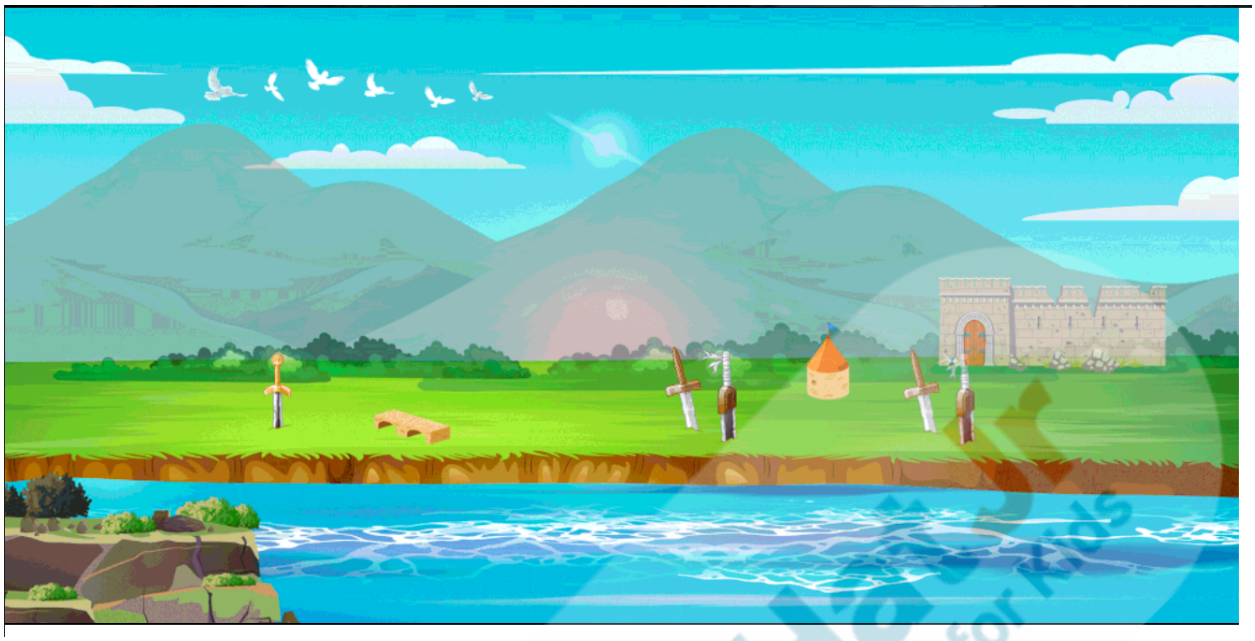
function to load an image.

```
function preload() {
  backgroundImg = loadImage("./assets/background.gif");
}
```

inside the **draw()** displaying the background image using **image().**

```
function draw() {
  image(backgroundImg,0,0,1200,600)
  Engine.update(engine);
```

OUTPUT:



Awesome now our theme is set. Now we have the ground ready. Now let's set up the tower where the cannon will be placed. Can you tell me how can we create the tower? Yes, we'll declare a tower variable and in the **setup() function** using **Bodies.rectangle()** create a horizontal rectangle. While creating the rectangle just add the x and y position as we'll be adding the height and width later on in the **draw()** function. We'll also add the options to the tower body as we want to create a static tower body so that it doesn't move on the screen.	**ESR:** We can create a rectangle using the **Bodies.rectangle** for the tower and place it above the ground.

```
function setup() {

  canvas = createCanvas(1200, 600);
  engine = Engine.create();
  world = engine.world;

  var options = {
    isStatic: true
  }

  ground = Bodies.rectangle(0, height - 1, width * 2, 1, options);
  World.add(world, ground);

  tower = Bodies.rectangle(160, 350, 160, 310, options);
  World.add(world, tower);

}
```

In the **draw()** function, using the **rect()** function pass the position X and position Y of the tower and the height and width of the tower.

```
function draw() {
  image(backgroundImg,0,0,1200,600)
  Engine.update(engine);


  rect(ground.position.x, ground.position.y, width * 2, 1);

  rect(tower.position.x, tower.position.y, 160, 310);

}
```

output

What do we see? Awesome, now let's add the image to the tower. Can you tell me how to add the image to the tower? Can you tell me how can we fix this?	**ESR:** We see that the tower is being created at the top left corner and not at the place where we want it to be. It's the same as the ground. **ESR:** Varied
To draw the tower at our given position we'll use **push() and pop()** functions. These functions are always used together. The **push()** function saves the current drawing style settings and transformations, while **pop()** restores these	

settings.

Between these **push() and pop()** functions we'll have the positions where we want to draw the tower.

We'll also add **rectMode(center)** to create the rectangle from the center.

to summarize
push() -> captures the new setting.
pop() -> reverts to the old setting.

```
function draw() {
  image(backgroundImg,0,0,1200,600)
  Engine.update(engine);


  rect(ground.position.x, ground.position.y, width * 2, 1);


  push();
  rectMode(CENTER);
  rect(tower.position.x, tower.position.y, 160, 310);
  pop();
}
```

OUTPUT:

Now the rectangle is placed at the correct position. We are placing the rectangle there.
So that it looks as if the tower is placed on the stone.

Now let's add the image to the tower.
To add the image to the tower we'll need to load the image in the preload function so that it gets loaded before the code is executed.

The teacher codes to load the image in the preload function.
Note:- The images are provided in the assets folder.

```
function preload() {
  backgroundImg = loadImage("./assets/background.gif");
  towerImage = loadImage("./assets/tower.png");
}
```

Now we need to add this image to the tower to do so we'll use the **image()** function in place of the **rect()** function that we used earlier.
and turn the **rectMode()** to **imageMode()** as now the image will be there in place of the rectangle.

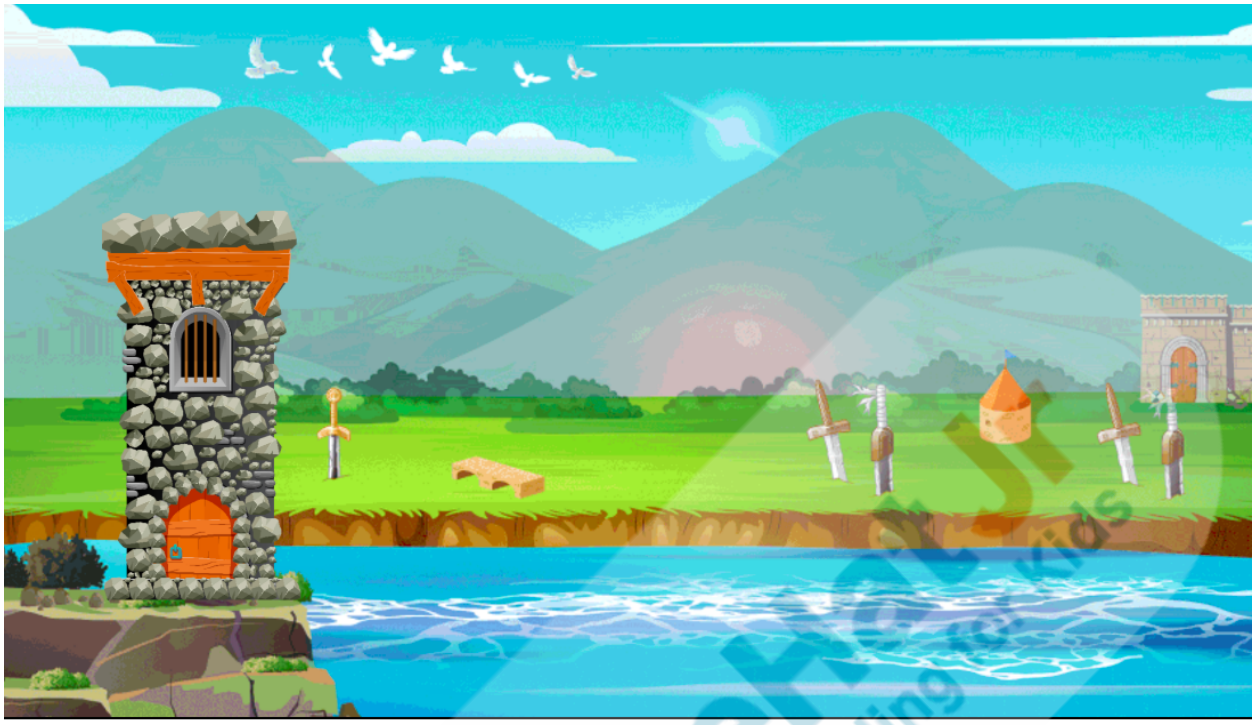
*The teacher codes to replace the **rect()** function with the **image()** function.*

```
function draw() {
  image(backgroundImg,0,0,1200,600)
  Engine.update(engine);


  rect(ground.position.x, ground.position.y, width * 2, 1);


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

Output:



Awesome now we have the tower on the ground. Now we just have to create a cannon and place it on the top of the tower.	
So to create a cannon we'll create a cannon class. We are creating the class for the cannon as we want the code for the cannon to be in a separate file that can be updated as we want without having to change other codes.	
Now we need to create a cannon to shoot the enemies. What are the elements that a cannon will have? Awesome. So let's create a **cannon class** that will help us to create the cannon.	**ESR:** The cannon component has a cannon from which the **cannonball** is shot and a base on which it rests.

First, we'll need to create the **cannon.js** file and add this file to the **index.html** file. *Teacher codes to create an empty cannon class and add the file in the index.html file.*	*Student observes.*

```
js > JS Cannon.js > Cannon
1   class Cannon {
2
3
4   }
```

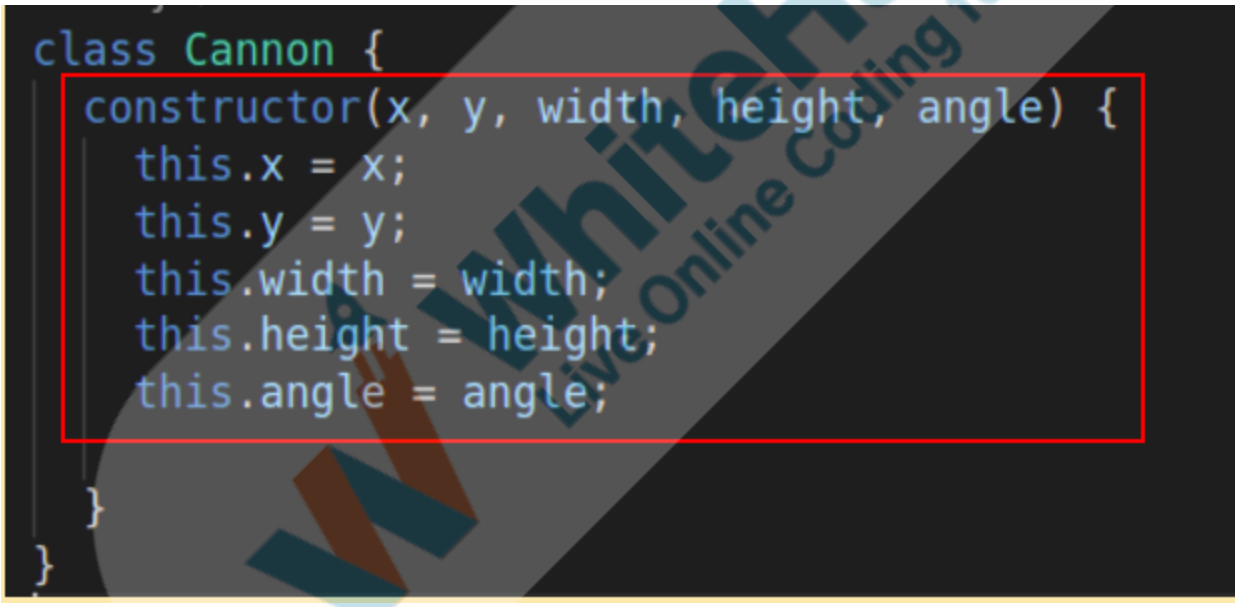
Add in **index.html**

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>Pirates Invasion</title>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/p5.js/0.10.2/p5.js"></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/p5.js/0.10.2/addons/p5.sound.min.js"></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/p5.js/0.8.0/addons/p5.dom.min.js"></script>
    <script src="./lib/p5.gif.min.js"></script>
    <script src="./lib/matter.js"></script>
    <!-- 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" />

    <!-- code to add scirpt will go here-->
    <script src="./js/Cannon.js"></script>


  </head>
  <body>
    <script src="sketch.js"></script>
  </body>
</html>
```

So what are the different properties that the cannon will have? Yes. Let's write these in the **constructor()**. We'll also add the angle property as our cannon will be changing the angles to shoot the cannonballs. *<The teacher codes to add the properties inside the **constructor()**.>*	ESR: The cannon will have the height, width, x and y positions. *The student suggests the properties to the teacher.*
	
Now we have the basic code for the cannon class ready in the constructor. Can you try to write the code for **display()** to display the code? let's get you started then.	**ESR:** yes
Teacher Stops Screen Share	

<table>
<tr><th colspan="2">STUDENT-LED ACTIVITY - 20mins</th></tr>
<tr><td colspan="2">

- **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.**

</td></tr>
<tr><td colspan="2">

ACTIVITY

- **Design the remaining cannon class.**
- **Create a cannon object using the class.**

</td></tr>
<tr><td colspan="2">Teacher starts slideshow  :Slide 14 to Slide 16</td></tr>
<tr><td>Run the presentation slide to set the student activity context.

• Create two boxes.
• Give them a realistic effect using Physics Engine.</td><td></td></tr>
<tr><td colspan="2">Teacher ends slideshow  </td></tr>
<tr><th>Teacher Action</th><th>Student Action</th></tr>
<tr><td>Guide the student to download the code from Student Activity 1 and fire up the web server.

We have started writing code to create the cannon class. We have written the basic code for the constructor now let’s write code to display the cannon body.</td><td>Student download the code from Student Activity 1.

Student fires up the chrome web server.</td></tr>
</table>

So at what position should we display the cannon? There will be two rectangular bodies, one on the top of another to make it look like a cannon. What can we use to create the rectangular cannon body? In the cannon class, create a **display()** function which will help us to display the cannon. Inside this display function we'll call the **push()** and **pop()** function. Then we'll set the **rectMode()** to center then create a rectangle using **rect()** with the width and height. We are using rectMode() to specify from where we want our rectangle to be drawn. When we specify the center then the rectangle is drawn from the center. Below the **pop()** create another rectangle that will serve as the cannon base and call the **noFill()** as we don't want any colors for the base. In the **sketch.js** file, In the **setup** function create a cannon, and in the **draw()** function display it. *The teacher helps the student with the code.*	**ESR:** We want to display it as the cannon mounted on the base. **ESR:** To make a cannon we can use the **rect()** function. Using this function we create two rectangle bodies. *The student codes to create the display function.* *Inside the function:-* • *call the push() and pop() functions.* • *call the **rectMode()** function and set it to **center.***

In the **sketch.js** file declare a variable called as angle and in the **setup()** function set its value as **20**. This would be the initial angle of the cannon. Currently, this doesn't have any use for the cannon as it is static; this will be used later when we add the movement to the cannon.	• *create the rectangle using **rect()** with the width and height from the constructor.* • *below **pop()** using **rect()** function create the rectangle with x= 70, y=20, width= 200, height= 200. Then call the **noFill()**.*

```
display() {
  //code to create cannon top
  push();
  rectMode(CENTER);
  rect(this.x, this.y, this.width, this.height);
  pop();

  //code to create cannon bottom
  rect(70, 20, 200, 200);
  noFill();
}
```

in **sketch.js** file in the setup function.

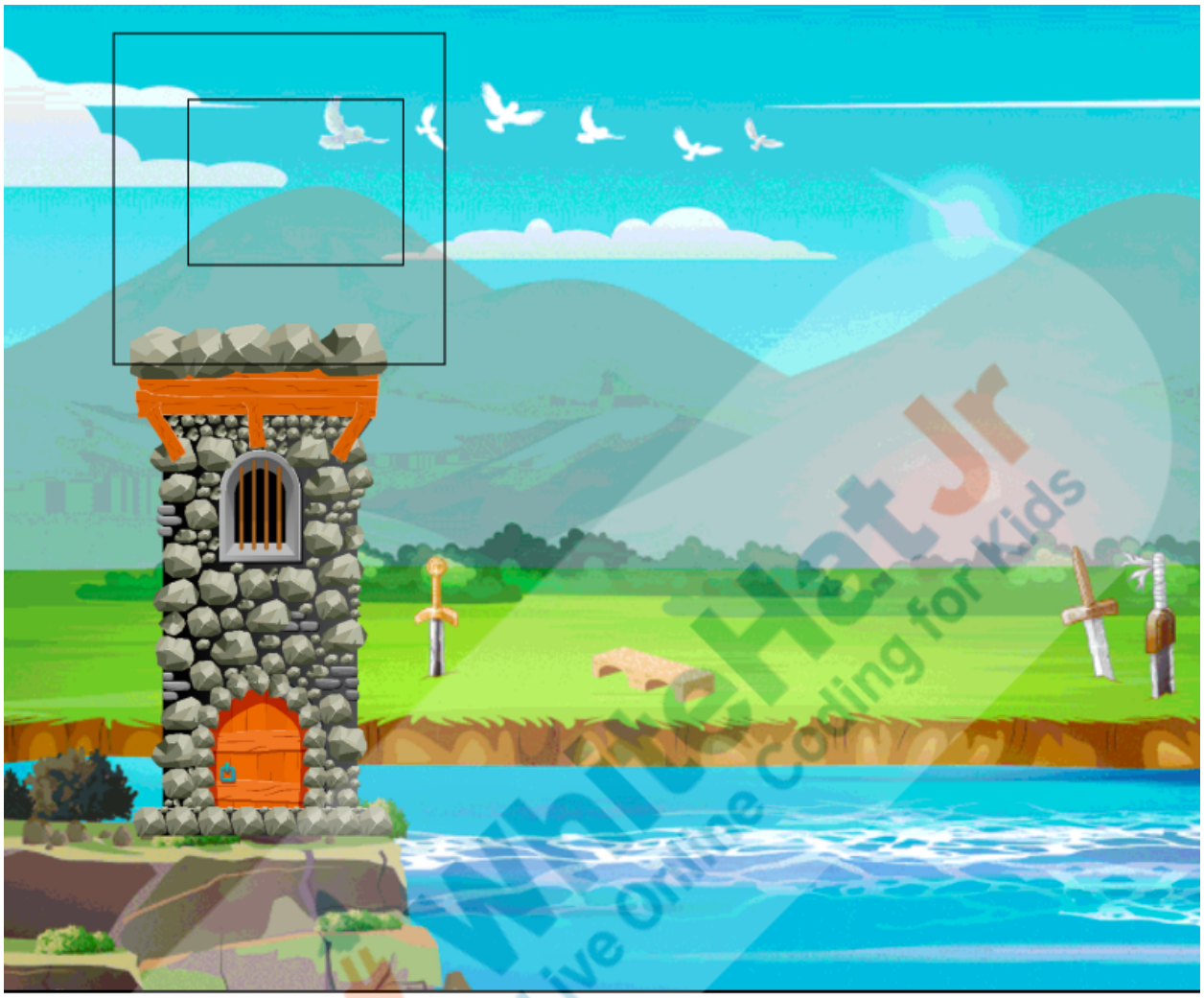
```
angle = 20
```

```
cannon = new Cannon(180, 110, 130, 100, angle);
```

in the draw function

```
cannon.display();
```

OUTPUT:



In our output, we can see that there are just 2 rectangles on the top of the tower. But we wanted the rectangles shaped like a cannon right? This will be fixed when we add the images to the rectangles.	**ESR:** Yes.

Can you tell me how to add the images to the rectangles? Perfect! *Teacher helps the student to add the images.*	**ESR:** We will first load the images in the constructor. Then using the **image()** add the images in the palace of rectangles and change the **rectMode()** to **imageMode().** **Student codes to add the images to the rectangles.**

```
display() {
  //code to create cannon top
  push();
  imageMode(CENTER);
  image(this.cannon_image, this.x, this.y, this.width, this.height);
  pop();

  //code to create cannon bottom
  image(this.cannon_base,70, 20, 200, 200);
  noFill();
}
```

OUTPUT:

Doesn't our game look awesome?	ESR: Yes.

Teacher Guides Student to Stop Screen Share

WRAP-UP SESSION - 5 Mins

FEEDBACK

- **Encourage the student to make reflection notes in Markdown format.**
- **Complement the student for her/his effort in the class.**
- **Review the content of the lesson.**

Teacher starts slideshow **from slide 17 to slide 27**

Activity details	Solution/Guidelines

Run the presentation from slide 17 to slide 21 **Following are the WARM-UP session deliverables:** ● **Appreciate the student.** ● **Revise the current class activities.** ● **Discuss the quizzes.**	Student will ask doubts related to the activities.
Quiz time - Click on in-class quiz	
Question	**Answer**
What is a class? A. it is a blueprint that contains the properties and functions of the object B. it is a library C. it contains a set of predefined functions D. it contains a collection of sprites	**A**
Which function is used to revert back to the old positions/settings? A. push() B. pop() C. translate() D. rotate()	**B**
Which is the correct usage to load the image in the game? A. addImage() B. image() C. loadImage() D. image.load()	**C**
End the quiz panel	
FEEDBACK ● **Encourage the student to make reflection notes in markdown format.** ● **Complement the student for her/his effort in the class.** ● **Review the content of the lesson.**	

<table>
<tr>
<td>Step 4:
Warp-Up
(5 mins)</td>
<td>You get Hats off for your excellent work!

In the next class, we will create the moving cannon, and cannonball to shoot from the cannon.</td>
<td>Make sure you have given at least 2 Hats Off during the class for:

 
</td>
</tr>
<tr>
<td colspan="2">* 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 1

Goal of the Project:

In Class 22, you have learned the concept of object-oriented programming and physics engines. In this project, you will practice the concepts learned in the class to create a physics body for the player and player base. You will also create a player archer class and its object.

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.

Can you create a Player against the Computer in the game of archery “EPIC ARCHERY”?</td>
<td>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.

Student engages with the teacher over the project.</td>
</tr>
</table>

I am very excited to see your project solution and I know you will do really well. Bye Bye!		
Teacher ends slideshow 		
Teacher Clicks 		
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	Wireframe Code	https://github.com/pro-whitehatjr/PRO-C22-wireframe

Teacher Activity 2	Pirates games	https://whitehatjr.github.io/PiratesInvasion/
Teacher Activity 3	Reference Link	https://github.com/pro-whitehatjr/PRO-C22-reference-link
Student Activity 1	Boilerplate code	https://github.com/pro-whitehatjr/PRO-C22-SA-Boilerplate_code
Teacher Reference visual aid link	Visual aid link	https://s3-whjr-curriculum-uploads.whjr.online/553693e6-a127-417f-b9ad-f1b9d74d1914.html
Teacher Reference In-class quiz	In-class quiz	https://s3-whjr-curriculum-uploads.whjr.online/7dcef7ca-7924-4ccd-9907-4190557dd81e.pdf
Project Solution	Epic Archery Stage 1	https://github.com/pro-whitehatjr/pro-c22-project-solution