
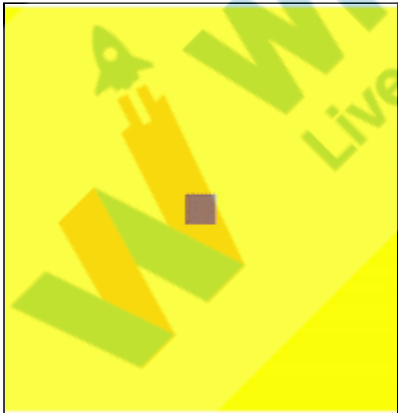







Topic	TREX AND THE INFINITE GAME WORLD	
Class Description	Students design a dinosaur for the Trex game with jump and run animations. Students will also learn to create an infinite game world in the limited screen space available to the players.	
Class	PRO-C10	
Class time	50 mins	
Goal	<ul style="list-style-type: none"> • Build a dinosaur with jump and run animations. • Learn to scale the images in the game. • Learn to print/display messages/outputs from the program into the console for testing purposes. • Create an infinitely scrolling ground for the dinosaur to run on. 	
Resources Required	<ul style="list-style-type: none"> • Teacher Resources <ul style="list-style-type: none"> ○ VS Code Editor ○ Laptop with internet connectivity ○ Earphones with mic ○ Notebook and pen • Student Resources <ul style="list-style-type: none"> ○ VS Code Editor ○ Laptop with internet connectivity ○ Earphones with mic ○ Notebook and pen 	
Class structure	Warm-Up Slides Teacher - led Activity 1 Student - led Activity 1 Teacher - led Activity 2 Student - led Activity 2 Wrap-Up Slides	10 mins 10 mins 5 mins 10 mins 10 mins 5 mins
WARM-UP SESSION - 10mins		
<div>  </div> Teacher starts slideshow from slides 1 to 8		

Refer to speaker notes and follow the instructions on each slide.	
Teacher Action	Student Action
<p>Hi, so good to see you again! How have you been? So this is going to be the first class of our infinite runner game. Are you excited to learn something new?</p> <p>Click on the slide show tab and present the slides.</p> <p>Run the presentation from slide 1 to slide 4.</p> <p>Following are the warm up session deliverables:</p> <ul style="list-style-type: none"> Connecting students to the previous class. 	<p>ESR: Thanks, yes I am excited about it.</p>
QnA Session	
Question	Answer
<p>Select the block of code that changes the background color to red when the right arrow key is pressed.</p>  <pre> A. if (Down(RIGHT_ARROW)) { background("red"); } </pre>	<p>B</p>

<p>B. <pre>if (keyIsDown(RIGHT_ARROW)) { background("red"); }</pre></p> <p>C. <pre>if (key(RIGHT_ARROW)) { background("red"); }</pre></p> <p>D. <pre>if (keyGoesDown(RIGHT_ARROW)) { background("red"); }</pre></p>	
<p>Select the button to be clicked on Visual Studio Code to see the working of the written code.</p> <p>A. </p> <p>B. </p> <p>C. </p> <p>D. </p>	A
Teacher Action	Student Action
<p>Run the presentation from slide 5 to slide 8 to set the problem statement.</p> <p>Following are the warm-up session deliverables:</p> <ul style="list-style-type: none"> ● Explain animation with real-life scenarios. ● Introduce the student to the animation concept. ● Steps to write and run the code. 	<p>Narrate the story by using hand gestures and voice modulation methods to bring in more interest in the student.</p>

<div>  </div> <p>Teacher ends slideshow</p>	
TEACHER-LED ACTIVITY 1 - 10 mins	
Teacher Initiates Screen Share	
<p><u>CHALLENGE</u></p> <ul style="list-style-type: none"> • Creating arrays. • Accessing the array elements. 	
Teacher Action	Student Action
<p>Step 1: Teacher-led Activity 1</p> <p>Before we move on to create the game, let's first learn a very important programming concept. Arrays!</p> <p>Till this point we have used variables to store information such as: a = 10, b = 20.</p> <p>But what if we need to store multiple values in a single variable?</p> <p>We can't just say a = 10,20. That would be programmatically wrong.</p> <p>That's where array's come into the picture. By definition, an array is a collection of the same or different data types. Such as you can create an array named friends and name of your friends in it like</p> <p>friends = ["sam", "john", "ramy"]</p> <p>One more example, if you want to store your marks of all the subjects, storing them in multiple variables would be a very tedious job. We have to define multiple variables.</p> <p>But instead we can create a single array and store all the marks in one place.</p>	<p><i>The teacher opens the VS Code Editor.</i></p>

<p>Here, first we will store all the marks in an array then we will write a function to give us the average of our score, where we will learn how to access the elements from the array.</p> <p>Let's code!</p> <p><i>Open the blank project from the Teacher Activity 1 link in VSCode.</i></p>	
<p>Let's code and define an array at the beginning of the code.</p> <p><i>The teacher writes the code.</i></p> <p>To define the array we need the name of the array as we give to the variables.</p> <p>To add the elements in the array we set the array name equal to the square bracket, and then add the elements in the square bracket separated by a comma(,).</p> <p>Once elements are added in the array, we can print the array in the console by using console.log() function.</p> <p>Let's run this code to see the output.</p> <p>Run the code and open the live server. Press F12 to see the output on the console.</p> <p><i>Ask the student to guess the output of the code.</i></p>	

```
1  var marks = [30,35,45,38,40];
2  console.log(marks);
3
4  function setup() {
5    |   createCanvas(400, 400);
6    | }
7
8  function draw()
9  {
10   |   background(220);
11   | }
```

Output:

```
▶ Array(5) [ 30, 35, 45, 38, 40 ]
```

Here we can see the array printed on the console.

Now, let's move forward and learn how to access the elements of the array.

If you want to go to your friend's house, what is the most important information you need?

Yes! You need the house number of your friend. In a similar way, if you want to access the elements of the array, we need the index of that element.

ESR:

such as the name of the friend's address, transportation etc.

Now what is an index?

Index is just like an address of the element inside the array.

But the important thing is that the index always starts from 0.
Our first element is always at 0 index.

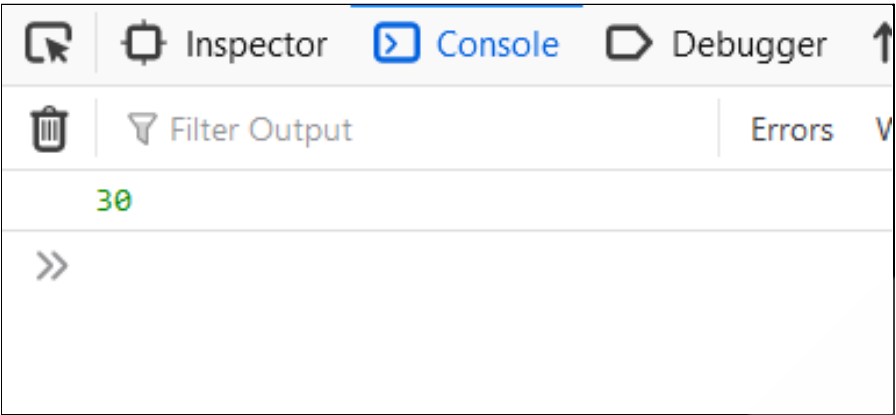
In our example, if we want to see the first element we will use the index as zero.

To get the element we will write the name of our array, then square bracket and inside the brackets we will write the index; here it is 0.

Let's run this code and see the output.

```
1  var marks = [30,35,45,38,40];
2  console.log(marks[0]);
3
4  function setup()
5  {
6    createCanvas(400, 400);
7  }
8
9  function draw()
10 {
11   background(220);
12 }
```

Output:

	
<p>Here we can see our first element printed as 30.</p> <p>Now let's write a function to get the average of our marks.</p> <p>To get the average we need to add all the elements and then divide the sum with the number of elements.</p> <p>Here, the number of elements is the length of the array. We can get the length of the array by writing arrayname.length.</p>	


```

var marks = [30,35,45,38,40];

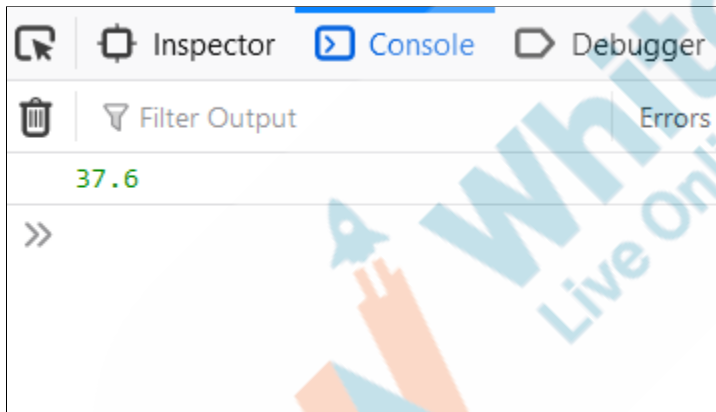
function score_average()
{
  var sum = marks[0] + marks[1] + marks[2] + marks[3] + marks[4];
  var avg = sum/marks.length;
  console.log(avg);
}

function setup()
{
  createCanvas(400, 400);
  score_average();
}

function draw()
{
  background(220);
}

```

Output:



Here we saw arrays can be very useful when we need to store multiple values in a single variable.


Now we have a challenge for you related to arrays.

Are you excited?

ESR:
Varied.

Teacher Stops Screen Share

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

STUDENT-LED ACTIVITY 1 - 5mins	
	Now it's your turn. Please share your screen with me.
<ul style="list-style-type: none"> • 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. 	
<div style="text-align: center;">  <p>Teacher starts slideshow for slide 9 and slide 10 Refer to speaker notes and follow the instructions on each slide.</p> </div>	
<p style="text-align: center;"><u>ACTIVITY</u></p> <ul style="list-style-type: none"> • Create an array. • Access array elements using index. 	
Teacher Action	Student Action
<p>Step 2: Student-led Activity-1</p> <p>In the previous activity we have seen how to create an array and access the elements from it.</p> <p>We created a program to find the average of the marks from the marks array.</p> <p>In this activity we are going to do a similar task.</p> <p>You will create an array which will store the weight of 4 people; then you will calculate the average weight of these 4 people.</p> <p>This can be used in multiple scenarios such as finding the average height or weight of all the students in the class or in a sports team.</p> <p>First create an array such as var weight = [35,38,40,45];</p>	<p><i>The student downloads the code from Student Activity 1.</i></p>

```
var weight = [35,38,40,45];
```

To calculate the average, we need to add all the weight values and then divide that sum with the number of elements in the array which is 4.

We are going to perform this task in the **setup()** function.

Create a variable to store the sum of all the weight values as **var sum** and we will add all the array elements in this variable.

```
var sum = weight[0] + weight[1] + weight[2] + weight[3];
```

We have the sum of all the weight values. We need to divide this sum with the number of elements of this array.

Can you tell me how we can find out the number of elements present in the array?

Great! `array.length` gives us the length of the array or we can say the number of elements present in that array.

Now, create a variable as **var average**; this variable will hold the average value which we are going to get by doing **var average = sum/weight.length**; to display the result use **console.log(average)**;

ESR:
using `array.length`

```
var weight = [35,38,40,45];

function setup() {
  createCanvas(400,400);

  var sum = weight[0] + weight[1] + weight[2] + weight[3];
  var average = sum/weight.length;
  console.log(average);
}
```

Output:

```
39.5
>>
```

Once the student completes the activity, the teacher will start with the main activity of the class.

TEACHER-LED ACTIVITY 2 - 10 mins

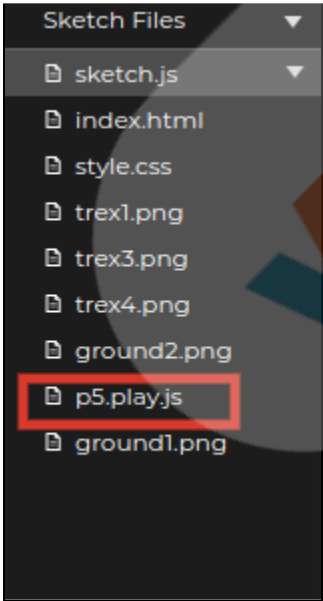
Teacher Initiates Screen Share

CHALLENGE

- Use of console.log() for testing purposes.
- Trick the player to perceive an infinitely scrolling ground.



Teacher can show slideshow from slides 11 to 22
 Refer to speaker notes and follow the instructions on each slide.

Teacher Action	Student Action
<p>Step 3: Teacher-led Activity Great! Let's open and look at the activity for the class (Teacher Activity 2).</p> <p>Check under the files and folders tab; I have uploaded different pictures of the T rex dinosaur. We will use them to create the animation.</p> <p>You will also observe that there is a new file called p5.play.js. This is a game library which code.org was internally using to allow you to create games!</p> <p>We have added the library in p5 now and we can create games similar to how we were creating in code.org.</p>	
	

```

<!DOCTYPE html><html><head>
  <script src="p5.min.js"></script>
  <script src="p5.dom.min.js"></script>
  <script src="p5.sound.min.js"></script>

  <script src="p5.play.js"></script>

  <link rel="stylesheet" type="text/css" href="style.css">
  <meta charset="utf-8">
</head>
<body>
  <script src="sketch.js"></script>
</body></html>

```

Creating animation is similar to adding images.

p5 has something called `loadAnimation` and `addAnimation` to add animations to the sprite.

We use **loadAnimation** to load all the images for the animation and then use **addAnimation** to add the animation to the sprite.

Let's code and create a sprite somewhere near the ground.
 Let's give a name to our sprite and store it in a variable.

Ask the student to give inputs while you are writing the code.

The teacher runs the code to check the output.

ESR:

*The student asks the teacher to use **createSprite()** to create a sprite and give it specific positions, width and height.*

*The student uses **drawSprites()** to draw the sprite on the screen.*

```
var trex

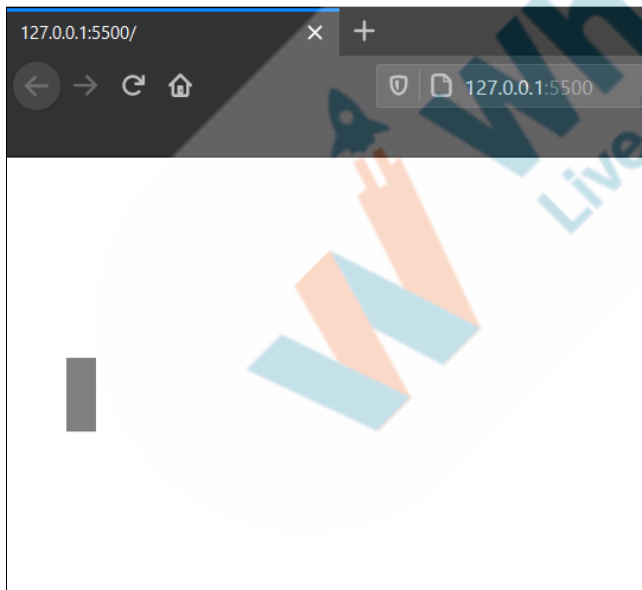
function setup(){
  createCanvas(600,200)

  //create a trex sprite
  trex = createSprite(50,160,20,50);

}

function draw(){
  background("white")
  drawSprites();
}
```

Output:



Let's load our animation for our Trex sprite in the **preload()** function and in the preload function we will define our animation using **loadAnimation()** function.

The student observes how to load and set animation for the Trex.

Note:

Using **loadAnimation()** to load the animations in the preload function, you can load multiple images in a single variable to make it animated.

addAnimation takes two arguments - a label which could be any string and the name of the animation which was loaded.

```

1  var trex ,trex_running;
2
3  function preload()
4  {
5      trex_running = loadAnimation("trex1.png", "trex
6      3.png", "trex4.png");
7  }
8
9  function setup(){
10     createCanvas(600,200)
11
12     //create a trex sprite
13     trex = createSprite(50,160,20,50);
14     trex.addAnimation("running", trex_running);
15 }
16
17 function draw(){
18     background("white")
19     drawSprites();
20
21 }
```

We have our running Trex. Now, let's make it jump when we press "space". The dinosaur should jump when we press space and then drop back to the ground. Any ideas on how to do it?

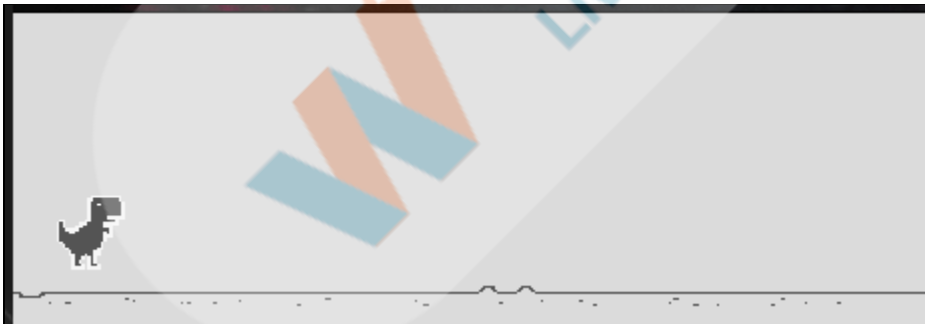
ESR:
Varied.

<p>Let's recollect some physics and gravity here. What happens to a ball when it bounces off the ground? How does its velocity change?</p> <p><i>The teacher demonstrates the physics behind a falling object with a physical example.</i></p>	<p>ESR: When the ball jumps off the ground, it has some velocity but due to gravity the velocity keeps reducing. It becomes 0 at the top but due to gravity it starts gaining velocity again when it starts falling towards the ground.</p>
<p>Exactly! Let's give the ball some velocity when space is pressed. Do you know how to do that?</p> <p>Which direction should we give the velocity?</p> <p>In our game is upwards velocity positive (+) or negative (-)? When we move from the bottom of the canvas to the top, the value of y decreases.</p> <p>Let's give a velocity of -10 to the Trex in vertical direction when space is pressed.</p> <p>This value of -10 is chosen arbitrarily; we can increase or decrease based on our need.</p> <p>If Trex is jumping very high we need to reduce this value; if Trex is jumping low, we have to increase it.</p> <p><i>The teacher writes to give the dinosaur a velocity of -10 when the "space" key is pressed.</i></p> <p><i>The teacher runs the code to see the output.</i></p>	<p>ESR: By using the if condition.</p> <p>ESR: Upwards?</p> <p>ESR: Negative.</p>

```

1  var trex ,trex_running;
2
3  function preload()
4  {
5      trex_running = loadAnimation("trex1.png", "trex3.png",
        "trex4.png");
6
7  }
8
9  function setup(){
10     createCanvas(600,200)
11     trex = createSprite(50,160,20,50);
12     trex.addAnimation("running", trex_running);
13 }
14
15 function draw(){
16     background("white");
17
18     if(keyDown("space"))
19     {
20         trex.velocityY = -10;
21     }
22
23     drawSprites();
24 }

```



What happened?

The Trex jumped off the screen.

This is because there was no gravity! Let's give some gravity.
What does gravity do?

ESR:
It pulls the object towards

What will be the effect of gravity on the velocity of the T-rex which is going up?

Let's add a line in our code which will do that.

trex.velocityY = trex.velocityY + 0.5;

Since velocity is -ve, the +0.5 will reduce the velocity every time in the upward direction and bring it to 0. Then, it will make the T-rex move in the other direction.

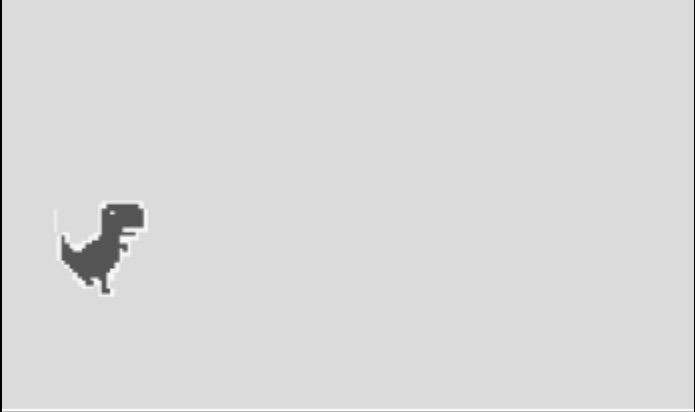
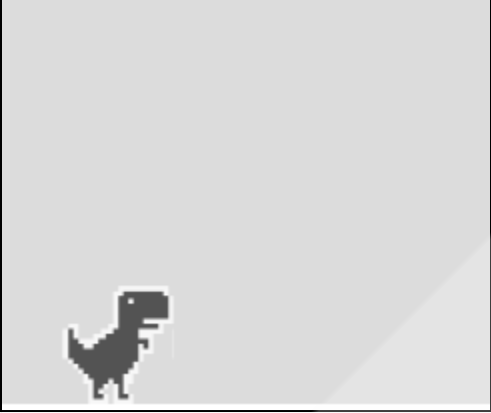
The teacher adds the line of code that will give the effect of gravity for the dinosaur. The teacher runs the code to see the output.

itself.

ESR:

Gravity will slow down the T-rex and reduce its velocity to 0. It will then pull the T-rex down.

```
function draw(){  
  background("white");  
  
  if(keyDown("space"))  
  {  
    trex.velocityY = -10;  
  }  
  trex.velocityY = trex.velocityY + 0.5;  
  drawSprites();  
}
```

 	
<p>Do you see gravity?</p> <p>What is the problem now?</p>	<p>ESR: Yes.</p> <p>ESR: The T-rex falls off the ground.</p>
<p>Any ideas on how to solve this.</p> <p>Let's create the edges sprite using createEdgeSprites() (this function creates the invisible walls around the edges of the canvas) and make the T-rex collide with the bottom edge so that it does not fall off the ground.</p>	<p>ESR: We can write code to create the edges and make the T-rex collide with the bottom edge.</p>

```
function setup(){
  createCanvas(600,200)
  trex = createSprite(50,160,20,50);
  trex.addAnimation("running", trex_running);
  edges = createEdgeSprites();
}

function draw(){
  background("white");

  if(keyDown("space"))
  {
    trex.velocityY = -10;
  }
  trex.velocityY = trex.velocityY + 0.5;
  trex.collide(edges[3]);

  drawSprites();
}
```

Edge is an array of sprites; we have 4 edges on the canvas and createEdgeSprites() will create the 4 edge sprites, But we only want to use the sprites which are below the Trex, so we will do that using array indexing.

Like we did in the starting of the class to access an element from the array we use its index.

Here the ground sprite is stored at the last element and it has a total of 4 elements. The Array index starts from 0 hence we have the index of the last edge sprite as 3, which is why we are detecting the collision with that sprite.

Looks like we have the jumping and running dinosaurs now!
And that was quick!



When we write the code, we understand what is happening in each line.

If you share the program with your friends, they may not understand the meaning of each and every line.

And also if you open your own code after some time you also may not understand each and everything at once.

To overcome this problem we can use comments in the code.

Comments are written in plain English to explain the purpose or working of code.

Comments are not executed when we run the program. In JavaScript to add the comment we first write 2 forward slashes (//); after this you can write the comment.

Our code doesn't have any comments.

Can you quickly help me in writing comments for the code?

The student helps the teacher in writing comments for different blocks of lines in the code.

```

3  function preload()
4  {
5      trex_running = loadAnimation("trex1.png", "trex3.png",
6      "trex4.png");
7  }
8  function setup(){
9      createCanvas(600,200)
10     trex = createSprite(50,160,20,50);
11     trex.addAnimation("running", trex_running);
12     edges = createEdgeSprites();
13
14 }
15
16 function draw(){
17     //set background color to white
18     background("white");
19
20     //jump when space key is pressed
21     if(keyDown("space"))
22     {
23         trex.velocityY = -10;
24     }
25     trex.velocityY = trex.velocityY + 0.5;
26
27     //stop trex from falling down
28     trex.collide(edges[3]);
29     drawSprites();
30 }

```

Good work. Now anyone can read our code and understand it easily.

Don't you think our dinosaur is too huge compared to the original game?

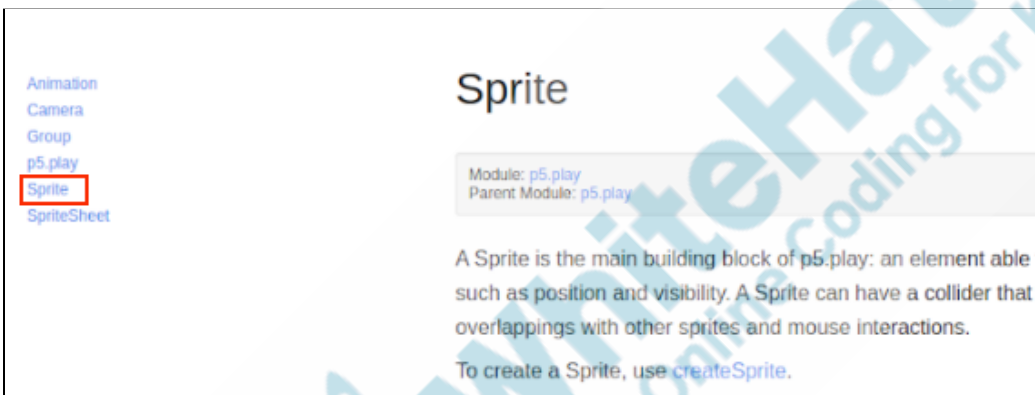
ESR: Yes.



There should be something in our Sprite module to help us scale the dinosaur. Help me look for it.

Let us look for the sprite object in p5.play documentation.

The student helps the teacher in looking for the `sprite.scale` in the sprite documentation.



- Properties
 - _rotation
 - animation
 - collider
 - debug
 - depth
 - friction
 - groups
 - height
 - immovable
 - life
 - mass
 - maxSpeed
 - mouseActive
 - mouseIsOver
 - mouseIsPressed
 - originalHeight
 - originalWidth
 - position
 - previousPosition
 - removed
 - restitution
 - rotateToDirection
 - rotation
 - rotationSpeed
 - **scale**
 - shapeColor
 - touching
 - velocity
 - visible
 - width

Let us use the **sprite.scale** instruction to scale the dinosaur to half its size.

The teacher writes and runs the code.

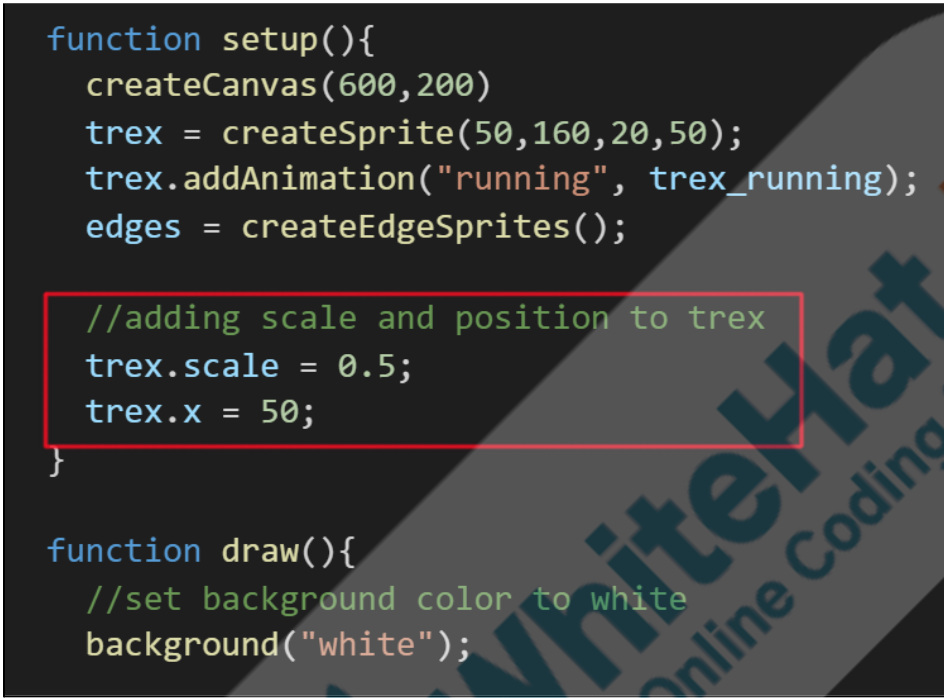
Does the dinosaur look the same size as in the original game?

ESR: Yes!



How can we position the dinosaur to the extreme left?

ESR: By setting the x position of the Trex sprite.

<p>Let us set the x position of the Trex sprite so that it is to the left. What could be the instruction?</p> <p><i>The teacher writes and runs the code.</i></p>	<p>ESR: trex.x = 50;</p>
 <pre> function setup(){ createCanvas(600,200) trex = createSprite(50,160,20,50); trex.addAnimation("running", trex_running); edges = createEdgeSprites(); //adding scale and position to trex trex.scale = 0.5; trex.x = 50; } function draw(){ //set background color to white background("white"); } </pre>	
<p>Awesome. Do you see our dinosaur doesn't jump correctly right now? Can you identify the problem?</p>	<p>ESR: The dinosaur keeps moving up and doesn't fall if we keep the space key pressed.</p>
<p>Right. This is a problem that we will solve later.</p> <p>For now, I will tell you an important tool using which you can find out what's happening inside the program while the program is running. This will help us find out and correct errors in our program.</p>	

Our editor has a console window where we can log any message while the program is running. We do this using **console.log()** instruction.

Let us print/display the name of the game in the console window.

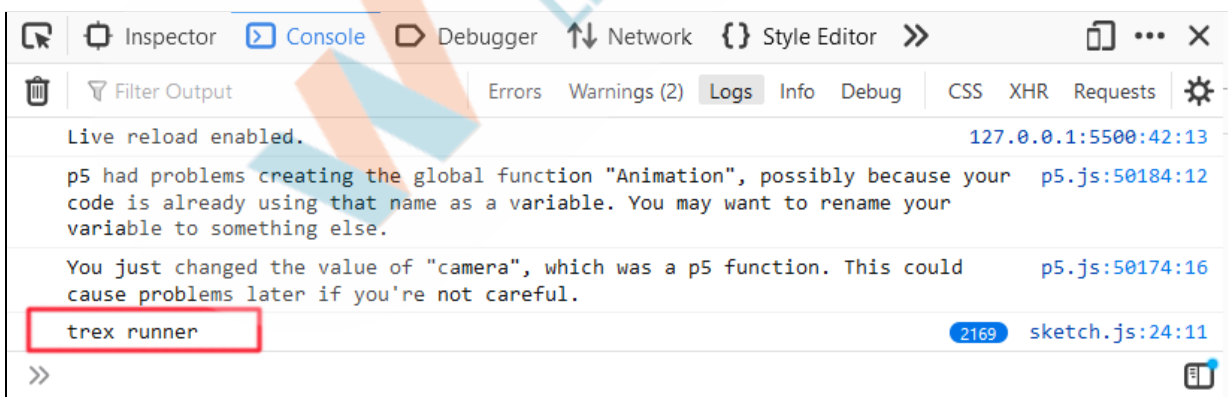
The teacher writes and runs the code.

The student observes how to use console.log() instruction.

```
function draw(){
  //set background color to white
  background("white");

  //using console.log
  console.log("trex runner");

  //jump when space key is pressed
  if(keyDown("space"))
  {
    trex.velocityY = -10;
  }
}
```



Do you know just like movies, games are made up of many frames? Anything we write inside the function draw() runs for

every frame. Anything we write outside the function draw() runs only once.

What would happen if we run console.log() inside the draw() function?

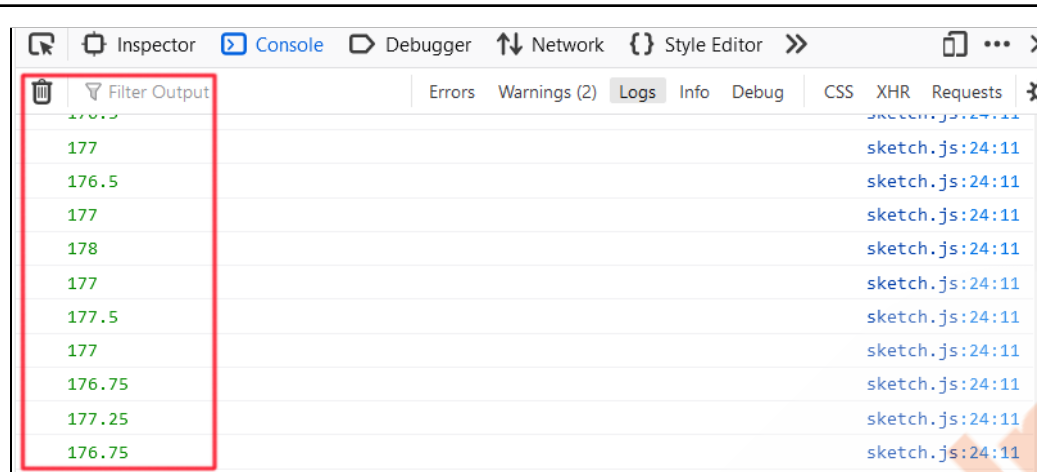
Let us write the console.log() instruction inside the draw() function. Also, instead of displaying the name of the game, let us print the y position of the T rex sprite.

The teacher writes the code and runs.

ESR: We would see the message from console.log() multiple times in the console window for every frame.

The student observes.

```
function draw(){  
  //set background color to white  
  background("white");  
  
  //logging the y postion of the trex  
  console.log(trex.y);  
  
  //jump when space key is pressed  
  if(keyDown("space"))  
  {  
    trex.velocityY = -10;  
  }  
}
```



The teacher runs the code and presses the space key to make the dinosaur jump.

What do you see in the console window? Why do the numbers change?

ESR: There are numbers as output in the console window. These numbers correspond to the y position of the Trex. When the Trex jumps, the y position of the Trex changes and the numbers in the console change.

Isn't it amazing to see how the y position of the Trex is changing?



`console.log()` is a powerful tool to help us understand what our program is doing at each stage.

We will use it repeatedly in our projects.

For now, I want you to get started on moving the dinosaur along an infinite ground space. I will guide you to do that.

ESR: Yes.

Teacher Stops Screen Share

<div style="text-align: center;">  <p>Teacher can show slideshow from slides 23 to 25 Refer to speaker notes and follow the instructions on each slide.</p> </div>	
Run the presentation for slide 18 to set the student activity context.	
Why don't you share your screen and create a moving ground?	<i>The student shares his/her screen, opens the Student Activity, duplicates the code and adds more code to it.</i>
<div style="text-align: center;">  <p>Teacher ends slideshow</p> </div>	
STUDENT-LED ACTIVITY 2 - 10 mins	
<ul style="list-style-type: none"> • 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. 	
<p style="text-align: center;"><u>ACTIVITY</u></p> <ul style="list-style-type: none"> • Create a ground sprite and make it move backward. • Reset the ground position when the dinosaur reaches the end of the ground. • Use a symmetrical ground image for the ground animation. 	
Teacher Action	Student Action
<p>Step 4: Student-led Activity Let us create a rectangular sprite called ground. This is where the T rex dinosaur will run. The ground sprite should ideally cover the entire screen.</p> <p>What will be the height and width of such a sprite? What will be its x and y position?</p>	<p><i>The student opens the Student Activity 2 and duplicates the code.</i></p> <p>ESR: Varied.</p>

Guide the student to create a ground sprite.

The student writes the code to create a ground sprite with the guidance of the teacher.

```
function setup(){
  createCanvas(600,200)
  trex = createSprite(50,160,20,50);
  trex.addAnimation("running", trex_running);

  //adding scale and position to trex
  trex.scale = 0.5;
  trex.x = 50;

  //create ground Sprite
  ground = createSprite(200,180,400,20);
}
```



Right now the ground is on the dinosaur. Don't you think the dinosaur should collide with the ground instead of the edges?

Let's fix this. Why don't we remove the edges and make the Trex collide with the ground sprite?

ESR: Yes.

The student writes the code.


```
//create ground Sprite
ground = createSprite(200,180,400,20);
}

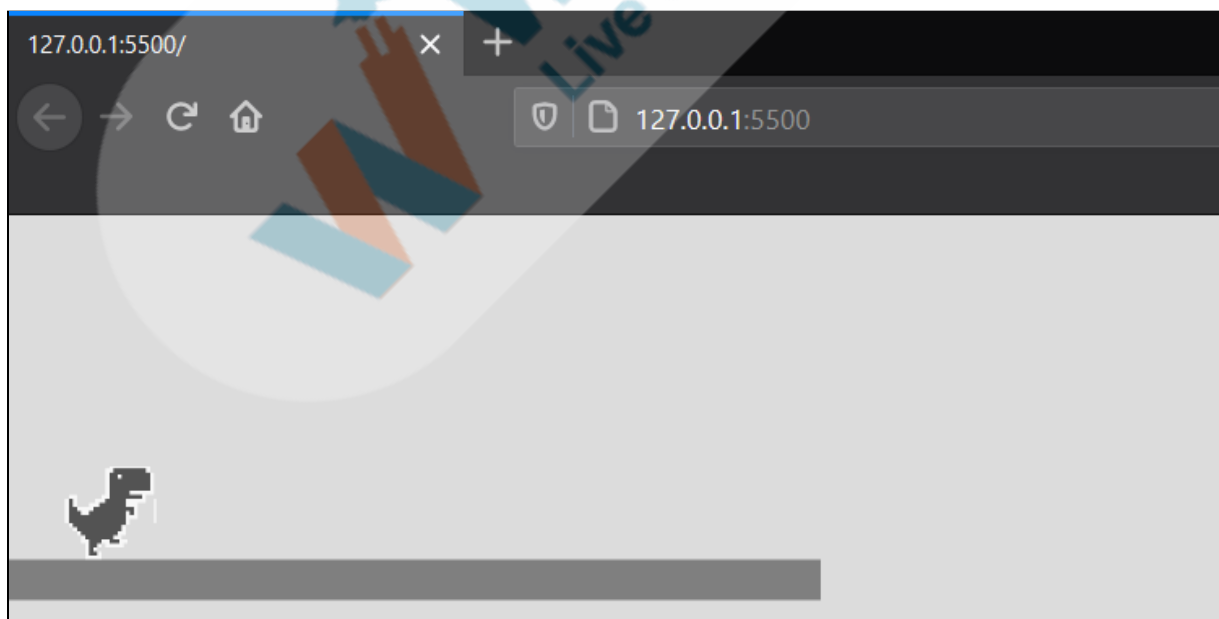
function draw(){
  background(220);

  //jump when space key is pressed
  if(keyDown("space"))
  {
    trex.velocityY = -10;
  }
  trex.velocityY = trex.velocityY + 0.5;

  //stop trex from falling down
  trex.collide(ground);

  drawSprites();
}
```

Output:



Ok, now let's move the dinosaur. There are two ways to make the player feel that the dinosaur is moving.

One would be to give some forward velocity to the Trex and the other would be to give some backward velocity to the ground.

Let us try each of the ways. Let us first give some forward velocity to the ground and see what happens.

To give the velocity in the x direction to the ground we will set **ground.velocityX = -2**, Here the ground will move from left to right direction.

Let's also keep displaying both the trex.x and trex.y in the console window.

Student writes code to give some x velocity to the ground.

The student first displays the trex.x and then trex.y in the console window.

Student runs the code and observes both the output and the console window.

```
function draw(){
  background(220);

  ground.velocityX = -2;
  console.log(ground.x);

  //jump when space key is pressed
  if(keyDown("space"))
  {
    trex.velocityY = -10;
  }
  trex.velocityY = trex.velocityY + 0.8;

  //stop trex from falling down
  trex.collide(ground);

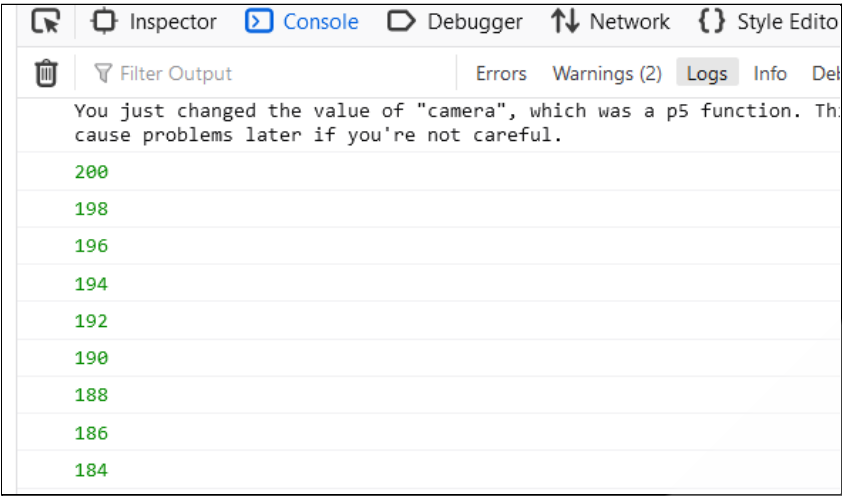
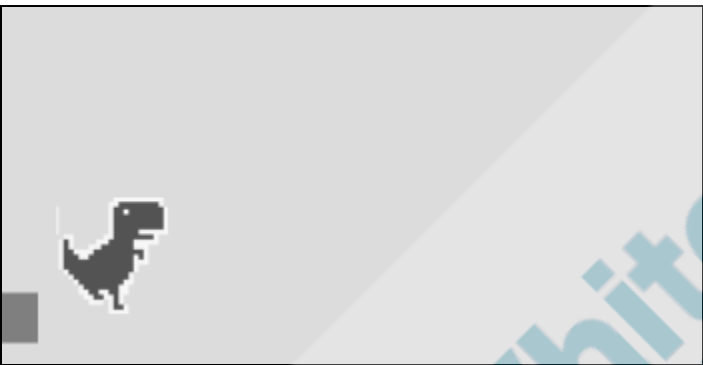
  drawSprites();
}
```

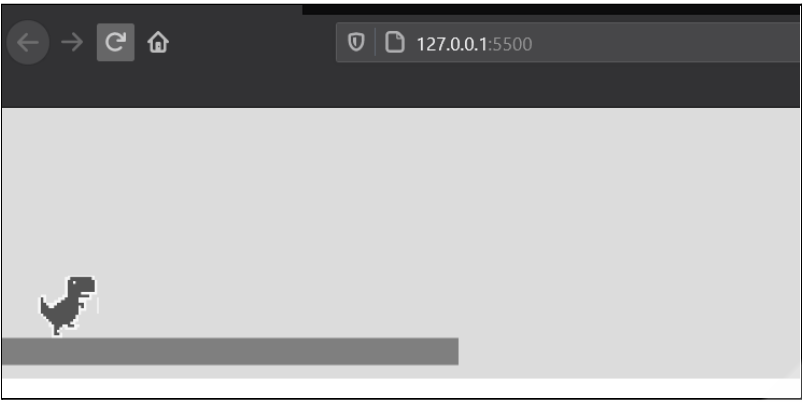
What is happening to the Trex?

Guide the student to look at the messages by pressing the F12 key on the keyboard

ESR:

The Trex is going outside the screen and then falling. trex.x is increasing constantly and trex.y is increasing exponentially.

		<p>Let's try the other way now. Let us try to give some backward velocity to the ground and see what happens. Also, let us log the ground.x now.</p>	<p><i>The student writes code to give some x velocity to the ground in the negative direction. The student prints/displays the ground.x on the console window.</i></p> <p><i>The student runs the code and observes both the output and the console window.</i></p>
---	--	--	---

	
What do you observe?	ESR: The Trex reaches the edge of the ground and then falls off.
Can you think of a way where the Trex never falls and the ground keeps scrolling infinitely?	ESR: Varied.
<p>One simple way would be to reset the ground back to the center of ground.x < 0.</p> <p>Once the ground has crossed the screen to the left, we are bringing it back to its original position.</p> <p>This way the ground will always be there. Let's do that and see what happens. How will we do that?</p> <p>The teacher guides the student to write the code to reset the ground to the center of ground.x < 0.</p>	ESR: Using conditional programming. <i>The student writes and runs the code to reset the ground if ground.x < 0.</i>

```
//create ground Sprite
ground = createSprite(200,180,400,20);
ground.addImage("ground",groundImage);
ground.x = ground.width/2;
}

function draw(){
  background(220);

  ground.velocityX = -2;
  console.log(ground.x);

  if (ground.x<0)
  {
    ground.x = ground.width/2;
  }
}
```



What do you see now?

ESR:

The ground keeps resetting itself.

What is the problem right now?

ESR:

We can see the ground

	moving and resetting itself.
How can we solve this problem?	ESR: Varied.
<p>Let us place the two identical ground images side by side.</p> <p>These images are joined together.</p> <p>When half of the ground image goes off the screen, we will reset the ground.</p> <p>This will build an illusion that there is always ground to cover.</p> <p>The player will never feel that the ground has reset itself and will see an infinitely scrolling ground.</p> <p>Let us use the image "ground1" already uploaded in the files directory. It has two ground images - one is red and the other is green - so that you can see what is happening. Later we will use a ground image as in the game.</p> <p>Where should the <code>ground.x</code> be for this image?</p> <p>We want the ground to be symmetrically placed on the screen. So, let us keep <code>ground.x = ground.width / 2;</code></p> <p>When <code>ground.x < 0</code>, we will reset the ground back to its center.</p>	<p>ESR: At half the width of the ground.</p> <p><i>The student sets the ground's sprite animation to ground2 and makes changes in the code. The student runs the code and sees the output.</i></p>

```

var trex ,trex_running;
var groundImage;

function preload()
{
  trex_running = loadAnimation("trex1.png", "trex3.png", "trex4.png");
  groundImage = loadImage("ground2.png");
}
  
```

You can see the ground resetting itself.

Let us use the actual ground image. There is a ground image uploaded on the animations tab. It contains two ground images joined to each other.

The student sets the new animation for the ground and runs the code.

Teacher Guides Student to Stop Screen Share

WRAP UP SESSION - 5 Mins



Teacher can show slideshow from slides 26 to 35
 Refer to speaker notes and follow the instructions on each slide.

Teacher Action

Student Action

Run the presentation from slide 26 to slide 35.






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
<p>What will be the output of the following code block?</p> <pre> 1 var marks = [30,35,45,38,40]; 2 console.log(marks[0]); 3 4 function setup() 5 { 6 createCanvas(400, 400); 7 } 8 9 function draw() 10 { 11 background(220); 12 }</pre> <p>A. 30 B. 40 C. undefined D. 5</p>	<p>A</p>
<p>What does the highlighted code snippet perform?</p> <pre> 13 } 14 15 function draw(){ 16 background("white"); 17 18 if(keyDown("space")) 19 { 20 trex.velocityY = -10; 21 } 22 23 drawSprites(); 24 }</pre> <p>A. when the user presses the spacebar the Trex jumps B. giving gravity to the Trex C. creating the Trex sprite</p>	<p>A</p>

D. none of the above	
Which of the following is adding gravity to the Trex? A. <code>trex.velocityY = trex.velocityY + 0.5;</code> B. <code>trex.velocityY = trex.velocityY - 0.5;</code> C. <code>trex.velocityX = trex.velocityX + 0.5;</code> D. <code>trex.velocityX = trex.velocityX - 0.5;</code>	A
End the quiz panel	
<div> <div>Teacher ends slideshow</div>  </div>	
<p align="center"><u>FEEDBACK</u></p> <ul style="list-style-type: none"> • 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. 	
<p>You get Hats Off for your excellent work!</p>  <p>Awesome!</p> <p>Remember we have a couple of bugs (problems) in our game.</p>	<p><i>Make sure you have given at least 2 Hats Off during the class for:</i></p> <div> <div> Creatively Solved Activities  +10 </div> <div> Great Question  +10 </div> <div> Strong Concentration  +10 </div> </div>

<p>In computer programs, if we don't get the output we expect, we term it as a bug!</p> <p>You might try to fix these bugs on your own. Let us meet in the next class again. Keep your game energy alive.</p> <p>We will need you to call one of your friends in the class - it is going to be a fun and exciting class where you will brainstorm with your friend about the game and generate new ideas.</p> <p>You can call any of your classmates or anyone who could help you in brainstorming about your game!</p>	
<p>Also, keep writing your notes in the reflection journal.</p>	
<p>* This Project will take only 30 mins to complete. Motivate students to try and finish it immediately after the class.*</p> <p>Project Overview</p> <p>TRAVELLING IN SHIP</p> <p>Goal of the Project: In Class 10, you have learned how to add animations/images and scale them in the game and how to create an infinitely scrolling ground.</p> <p>In this project, you will apply what you have learned in the class to create an infinitely moving background of a sea with a ship in it. The ship will have moving up and down animation.</p> <p>Story: It is vacation time; your younger brother wants to go on a Cruise. Due to the present situation, the parents denied any outing. He is very upset and to cheer him up, you've decided to use your coding ability to create a sailing ship game.</p>	<p>* This Project will take only 30 mins to complete. Motivate students to try and finish it immediately after the class.*</p> <p><i>The students engage with the teacher over the project.</i></p>

I am very excited to see your project solution and I know you will do really well. Bye Bye!	
<div> <div>Teacher Clicks</div> <div>✕ End Class</div> </div>	
Additional Activities <i>Challenge the student to create the running ghost animation using Student Additional Activity Link. Images are uploaded.</i> <i>The teacher can show the output to the student on what the ghost animation should look like.</i>	<i>The student creates the running ghost animation similar to Teacher Additional Activity Reference.</i>
<i>Encourage the student to write reflection notes in their reflection journal using markdown.</i> Use these as guiding questions: <ul style="list-style-type: none"> • What happened today? <ul style="list-style-type: none"> ○ 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? 	<i>The student uses the markdown editor to write her/his reflections in the reflection journal.</i>

Activity	Activity Name	Links
Teacher Activity 1	Template Code	https://github.com/pro-whitehatjr/C10_teacher_activity_1
Teacher Reference	Reference code of Teacher Activity 1	https://github.com/pro-whitehatjr/proc-c10-ta1-reference

Teacher Activity 2	Trex Stage 0	https://github.com/pro-whitehatjr/Trex_S tage_0
Teacher Activity 3	Sprite documentation	https://molleindustria.github.io/p5.play/docs/classes/Sprite.html
Teacher Activity 4	Reference code (Trex Stage 1)	https://github.com/pro-whitehatjr/Trex_S tage_1
Teacher Reference	Reference code of Student Activity 1	https://github.com/pro-whitehatjr/pro-c10-sa1-reference
Student Activity 1	Template Code	https://github.com/pro-whitehatjr/pro-c10-sa1-template
Student Activity 2	Trex Stage 0.5	https://github.com/pro-whitehatjr/Trex_ stage_0.5
Student Activity 3	Trex Game Link	http://www.trex-game.skipser.com/
Student Additional Activity	Running Ghost (Empty Activity)	https://github.com/pro-whitehatjr/Runnin g-ghost_animation
Teacher Additional Activity	Reference	https://github.com/pro-whitehatjr/Ghost_animation_2
Teacher Reference visual aid link	Visual aid link	https://curriculum.whitehatjr.com/Visual+Project+Asset/PRO_VD/BJFC-PRO-V3-C10-withcues.html
Teacher Reference In-class quiz	In-class quiz	https://s3-whjr-curriculum-uploads.whjr.online/a87850d6-6428-40d8-a55a-acfa840bc922.pdf
Project Solution	Travelling In Ship	https://github.com/pro-whitehatjr/Project_C10_Travelling_in_Ship