

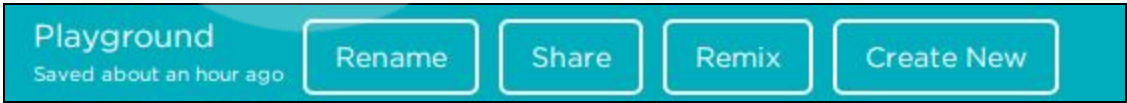
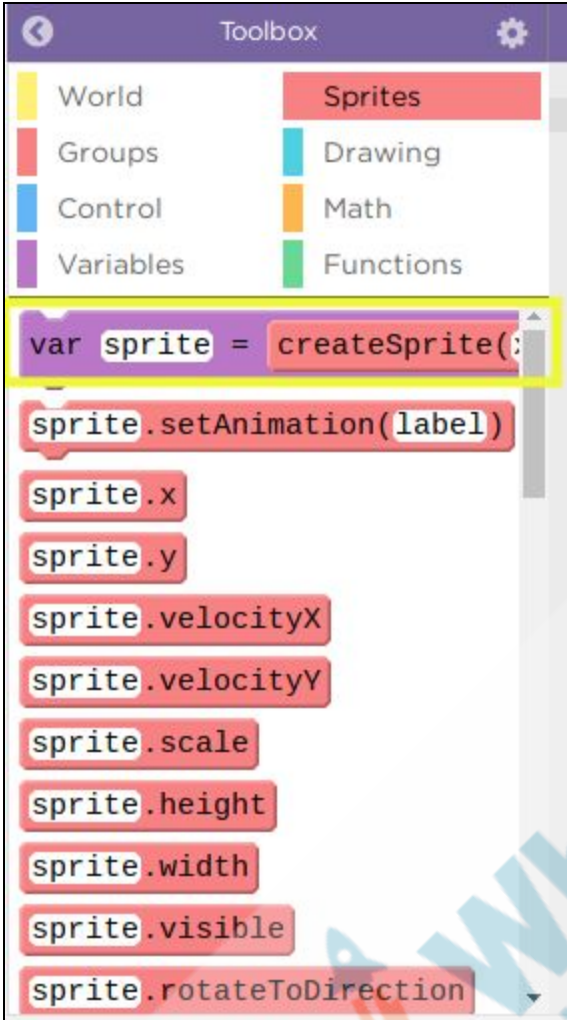


<b>Topic</b>	<b>SPRITE OBJECTS</b>	
<b>Class Description</b>	Students create a sprite object in their game using the existing sprite class and access the predefined sprite properties and functions. They create an animated ball in a playground project where they experiment with different sprite properties.	
<b>Class</b>	<b>PRO-C2</b>	
<b>Class time</b>	<b>45 mins</b>	
<b>Goal</b>	<ul style="list-style-type: none"> <li>• Create a sprite object.</li> <li>• Access predefined sprite properties and functions.</li> </ul>	
<b>Resources Required</b>	<ul style="list-style-type: none"> <li>• Teacher Resources               <ul style="list-style-type: none"> <li>○ Code.org login</li> <li>○ Laptop with internet connectivity</li> <li>○ Earphones with mic</li> <li>○ Notebook and pen</li> </ul> </li> <li>• Student Resources               <ul style="list-style-type: none"> <li>○ Code.org login</li> <li>○ Laptop with internet connectivity</li> <li>○ Earphones with mic</li> <li>○ Notebook and pen</li> </ul> </li> </ul>	
<b>Class structure</b>	<b>Warm Up - Slide show option</b> <b>Teacher-Led Activity</b> <b>Student-Led Activity</b> <b>Wrap Up - Slide show option</b>	<b>15 Mins</b> <b>8 Mins</b> <b>30 Mins</b> <b>5 Mins</b>
<b>WARM UP SESSION - 15 mins</b>		
<div>  </div> <p><b>Teacher starts slideshow from slides 1 to 17</b>            Refer to speaker notes and follow the instructions on each slide.</p>		
<b>Activity details</b>		<b>Solution/Guidelines</b>

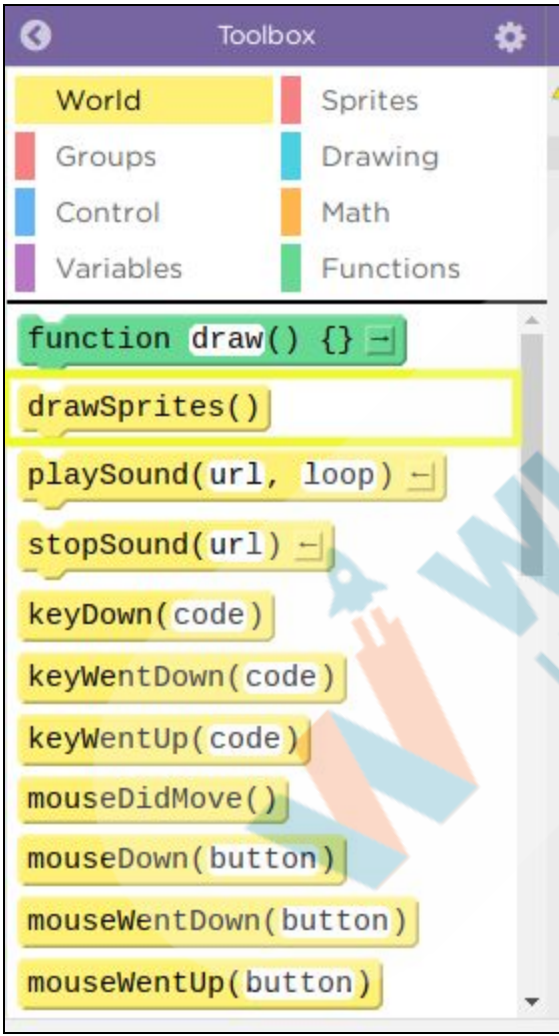
<p><i>Hey &lt;student name&gt;. How are you? It's great to see you! Are you excited to learn something new today?</i></p> <p><b>Run the presentation from slide 1 to slide 10.</b></p> <p><b>Following are the warm up session deliverables:</b></p> <ul style="list-style-type: none"> <li>• Connecting students to the previous class.</li> <li>• Explaining sprites through real life objects.</li> <li>• Explain properties and functions of objects.</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
<p>Consider the following Student class-</p> <pre>class Student {     constructor(){         this.firstname = 'Shalini';         this.lastname = 'Sharma';         this.comp_marks = '75';     } }</pre> <p>Which of these could be a correct way to create an object of Student class?</p> <p>A. var Student s = new Student();          B. var s = new Student();          C. new s = Student();          D. Student s = new Student();</p>	<p><b>B</b></p>
<b>Continue the warm up session</b>	
Activity details	Solution/Guidelines

<p><b>Run the presentation from slide 11 to slide 17 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>• Introduce students to the coding environment (on code.org) - Workspace, blocks and output.</li> <li>• Steps to write and run the code.</li> </ul>		<p>Narrate the story by using hand gestures and voice modulation methods to bring in more interest in students.</p>
<p>Teacher ends slideshow</p> 		
<p><b>TEACHER-LED ACTIVITY - 8mins</b></p>		
<p><b>Teacher Initiates Screen Share</b></p>		
<p><b><u>CHALLENGE</u></b></p> <ul style="list-style-type: none"> <li>• Introduce code org - show block and text mode of coding.</li> <li>• Show how to design objects as sprites.</li> <li>• Challenge them to create their sprite objects</li> <li>• Assign properties and functions to the sprites.</li> </ul>		
<p><b>Activity details</b></p>		<p><b>Solution/Guidelines</b></p>
<p><b>Step 2: Teacher-led Activity (15 min)</b></p>	<p>Let us create a project 'Playground' where we will experiment with sprite objects.</p> <p><i>Logs in code.org and creates a Playground project.</i></p> <p><a href="#">Empty Teacher Activity 1 (Press Remix)</a></p>	<p><i>Student watches.</i></p>
		
	<p>Look into the 'Sprites' section under our Toolbox. It should tell us how to</p>	<p><i>The student looks into the Sprite section under Toolbox.</i></p>

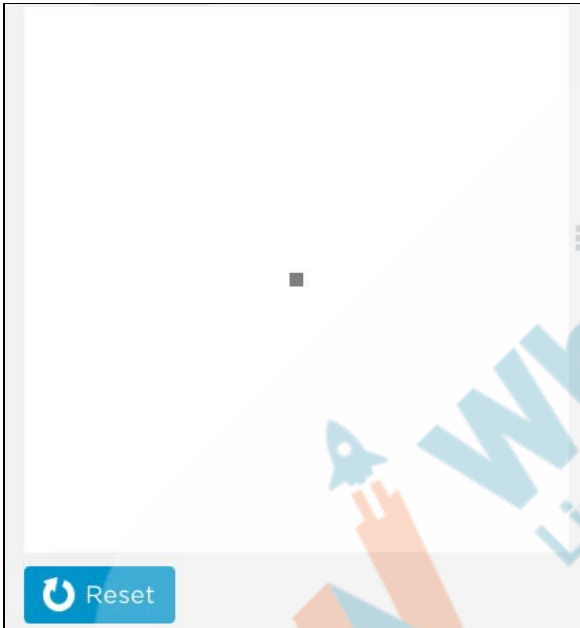
	create a Sprite object and what properties and functions it has.	
 <p>The screenshot shows the WhiteHat Jr interface. On the left is the 'Toolbox' with categories: World (yellow), Groups (red), Control (blue), Variables (purple), Sprites (red), Drawing (cyan), Math (orange), and Functions (green). On the right, a list of properties and functions for a 'sprite' object is displayed: <code>sprite = createSprite()</code>, <code>sprite.setAnimation(label)</code>, <code>sprite.x</code>, <code>sprite.y</code>, <code>sprite.velocityX</code>, <code>sprite.velocityY</code>, <code>sprite.scale</code>, <code>sprite.height</code>, <code>sprite.width</code>, <code>sprite.visible</code>, and <code>sprite.rotateToDirection</code>. At the bottom of the list is a button labeled 'Show Debug Commands'.</p>		
	Do you see something which will help us create a Sprite object?	<b>ESR:</b> Yes! <b>var sprite = createSprite();</b>

	
	<p>Awesome! We had to create the ball in our Pong Game</p> <p>Let's use this to create a sprite called 'ball' - which we will animate in this game! Sounds fun?</p> <p><i>Teacher writes code for creating the ball sprite and runs the code. Teacher asks the student for x,y, width and height of the ball.</i></p>
	<p><b>ESR:</b> Yes!</p> <p><i>The student watches how to create a ball sprite.</i></p>

	Sprites are always created rectangular. We will see how to change their shape later.	
<div> <b>Code:</b> <pre> 1 var ball = createSprite(200,200,10,10); 2 function draw() { 3   } 4 </pre> </div> <div> <b>Output:</b>  </div>		
	<p>The screen doesn't show anything. What do you think happened?</p> <p><i>Remind the student that they have to be very specific in giving instructions to the computer (water bottle example).</i></p>	<p><b>ESR:</b></p> <p><i>Student takes a guess at why the sprite hasn't appeared on the screen.</i></p>

	<p>I think the computer doesn't know that it has to draw the sprite on the screen. Let us see if there is an instruction to draw the sprite on the screen. Do you find one?</p> <p><i>Teacher scrolls the toolbox under the World section to locate drawSprites().</i></p>	<b>ESR:</b> drawSprites()
		
	<p>Let us write this code inside draw() and see what happens.</p>	



	Teacher writes the code for drawing sprites and runs the code.	The Student observes and learns.
<b>Code:</b> <pre> 1 var ball = createSprite(200,200,10,10); 2 function draw() { 3   drawSprites(); 4 } 5 </pre>		
<b>Output:</b> 		
	<p>We seem to have a ball exactly at the centre.</p> <p>Now let's look at what are the properties and functions available for this sprite. Let's see what we can do with the ball. What are some of the properties and functions you can see? Let's go back to Sprites and see"</p>	<b>ESR:</b> sprite.x, sprite.y, sprite.velocityX, sprite.velocityY, sprite.play(), sprite.bounce(), sprite.bounceOff() etc.



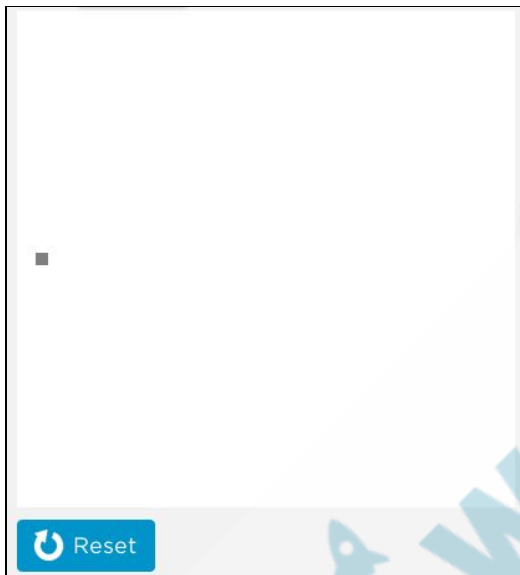
	Ok. Let's try the simplest one. What do you think sprite.x will do?"	<b>ESR:</b> It will set the x position of our sprite.
 <p>The image shows the Scratch interface. On the left is the 'Toolbox' with categories: World, Groups, Control, Variables, Sprites, Drawing, Math, and Functions. The 'Sprites' category is selected. Below it is the 'Sprite Properties' panel, which is highlighted with a yellow box. It contains the following code blocks: <code>var sprite = createSprite(100, 100, 100, 100);</code>, <code>sprite.setAnimation(label)</code>, <code>sprite.x</code>, <code>sprite.y</code>, <code>sprite.velocityX</code>, <code>sprite.velocityY</code>, <code>sprite.scale</code>, <code>sprite.height</code>, <code>sprite.width</code>, <code>sprite.visible</code>, and <code>sprite.rotateToDirection</code>.</p>		
	<p>Let's try and see. Our sprite's name is ball - so we are going to write ball.x.</p> <p><i>Teacher writes code and runs.</i></p>	<i>The student observes and learns.</i>

**Code:**

```

1 var ball = createSprite(200,200,10,10);
2 function draw() {
3   ball.x = 20;
4   drawSprites();
5 }
6

```

**Output:**


The ball seems to have shifted in its horizontal position. Let's try changing both x and y position so that the ball is at the top left corner. What do you think would be ball.x and ball.y?

Remember the sprite's center would be located at the x and y position we mention.

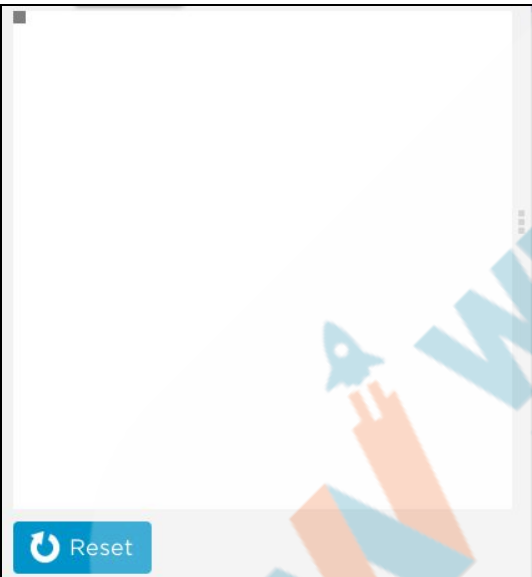
***\*Hint:** Students can use the grid option at the bottom of the preview window to find the position of x and y.*

**ESR:**

```

ball.x = 5;
ball.y = 5;

```

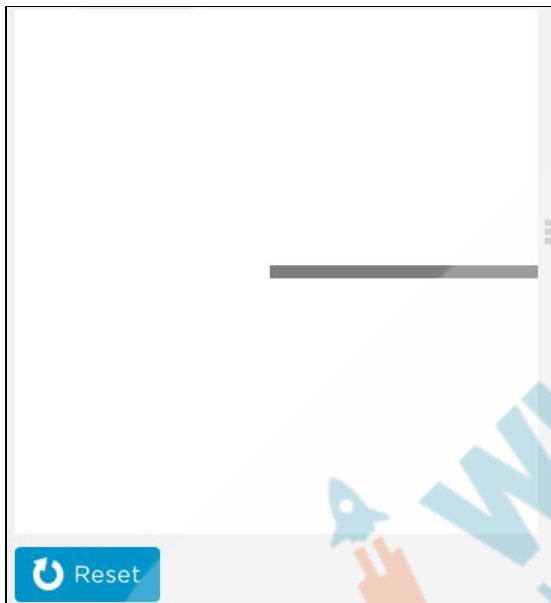
	<p>Let's try that and see what happens.</p> <p><i>Teacher writes code and runs.</i></p>	
<p><b>Code:</b></p> <pre> 1 var ball = createSprite(200,200,10,10); 2 function draw() { 3   ball.x = 5; 4   ball.y = 5; 5   drawSprites(); 6 } 7 </pre> <p><b>Output:</b></p> 		
	<p>So we can now place the ball anywhere on our screen.</p> <p>What do you think will happen if we assign our ball sprite some velocityX?</p> <p>Let us try and see.</p> <p><i>Teacher writes code and runs.</i></p>	<p><b>ESR:</b></p> <p>The ball will start moving horizontally.</p>

**Code:**

```

1 var ball = createSprite(200,200,10,10);
2
3 function draw() {
4   ball.velocityX = 2;
5   drawSprites();
6 }
7

```

**Output:**


What do you think happened?

Actually, the ball is moving but drawing itself continuously at different positions, similar to what happened with our moving paddle - remember? What should we do to allow only one ball to be drawn? Let's try that.

*Teacher writes the code and runs.*

**ESR:**

The ball is growing in size.

**ESR:**

Clear the screen before drawing the ball using `background("white")`.

**Code:**

```

1 var ball = createSprite(200,200,10,10);
2
3 function draw() {
4   background("white");
5   ball.velocityX = 2;
6   drawSprites();
7 }
8

```

**Output:**

The ball moves horizontally to the right.

We have got our ball to move now!  
 What should we do to move the ball horizontally in the opposite direction?

Let's try that.

**ESR:**

ball.velocityX = -2

**Code:**

```

1 var ball = createSprite(200,200,10,10);
2
3 function draw() {
4   background("white");
5   ball.velocityX = -2;
6   drawSprites();
7 }
8

```

**Output:**

The ball moves horizontally to the left.

Let's try assigning velocityY instead of velocityX. What do you think will happen?

Let's try.

*Teacher writes and runs the code.*

Let's try to move the ball in the

**ESR:**

The ball will move vertically.

	upward direction. What do you think we should do?  <i>Teacher writes and runs the code.</i>	<b>ESR:</b> ball.y = -2
<b>Code:</b>  <pre> 1 var ball = createSprite(200,200,10,10); 2 3 function draw() { 4   background("white"); 5   ball.velocityY = 2; 6   drawSprites(); 7 } 8 </pre>		
<b>Output:</b> The ball moves downward vertically.		
<b>Code:</b>  <pre> 1 var ball = createSprite(200,200,10,10); 2 3 function draw() { 4   background("white"); 5   ball.velocityY = -2; 6   drawSprites(); 7 } 8 </pre>		
<b>Output:</b> The ball moves upward vertically.		
	What do you think will happen if we give both velocityX and velocityY?  Yes! The ball will move both in horizontal and vertical direction. It will look as if it is moving diagonally. Let's try it.  <i>Teacher writes and runs the code.</i>	<b>ESR:</b> The ball will start moving diagonally?

### Code:

```

1 var ball = createSprite(200,200,10,10);
2
3 function draw() {
4   background("white");
5   ball.velocityX = 2;
6   ball.velocityY = 3;
7   drawSprites();
8 }
9

```

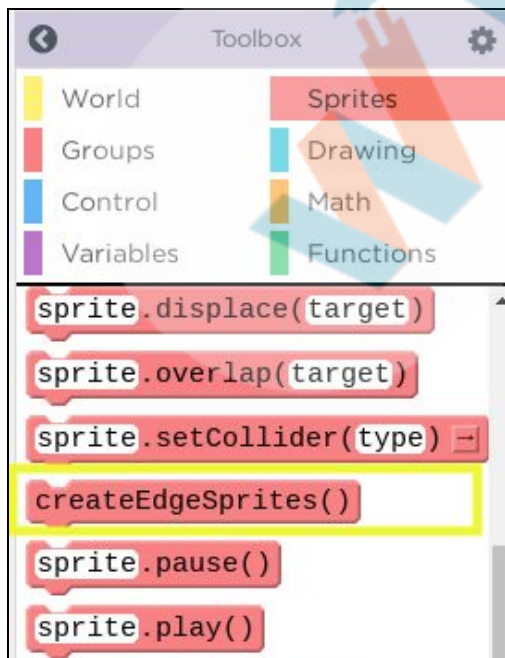
### Output:

The ball moves diagonally.

Wow! It seems we can make our ball move in any direction now!

However, after reaching the end, our ball disappears. We want the ball to bounce off the edges. There is a function in sprites that creates boundaries/edges on our screen. Can you find it?"



**ESR:** createEdgeSprites()





	<p>Let us use this to create edges and make our ball bounce off those edges.</p> <p>We have a bounceOff instruction for each sprite which takes the target from which you want the balls to bounce!</p> <p><i>Teacher writes code to create edges and bounce the ball off the edges.</i></p>	<p><i>Student observes the ball flickering near the edges</i></p>
<p><b>Code:</b></p> <pre> 1  var ball = createSprite(200,200,10,10) 2 3  function draw() { 4    background("white"); 5 6    ball.velocityX = 2; 7    ball.velocityY = 3; 8 9    createEdgeSprites(); 10   ball.bounceOff(edges); 11 12   drawSprites(); 13 } 14 </pre>		
<p><b>Output:</b> The ball keeps moving to and fro near the boundary.</p>		
	<p>What happened?</p> <p>Actually, the draw function is called in every frame! All the instructions inside 'draw' are executed every frame.</p>	<p><i>Student takes a guess at what happened.</i></p>

	<p>The ball first bounces off the edges but then 'draw' gets called in the next frame and balls get a velocity in lines 6 and 7 towards the edges.</p> <p>So it again bounces off the edges and this keeps happening.</p> <p>Let's assign the velocity to the ball outside 'draw' at the top. This will give velocity to the ball only once. Let 's see what happens then.</p> <p><i>Teacher writes the code and runs.</i></p>	
<p><b>Code:</b></p> <pre> 1  var ball = createSprite(200,200,10,10); 2  ball.velocityX = 2; 3  ball.velocityY = 3; 4 5  function draw() { 6    background("white"); 7    createEdgeSprites(); 8    ball.bounceOff(edges); 9    drawSprites(); 10 } 11 </pre> <p><b>Output:</b> The ball bounces off all the edges.</p>		
	<p>Can you recall again what's happening with the ball?</p> <p>Aren't sprites fun? We will also experiment with other properties and functions of sprite and hope it is going to be more fun!</p>	

	For now, there is a challenge for you. Why don't you create 4 balls, all of which bounce off the walls and also bounce off AGAINST EACH OTHER.	<i>Student shares the screen and starts the project.</i>
 <b>Teacher starts slideshow :Slide 18-19 (Only 1 slide for this Activity)</b>		
<b>Run the presentation for slide 18 to set the student activity context.</b>		
<p>Here's a challenge for you. You need to create 4 balls that bounce off when they touch the edges of the canvas as well as when they touch each other.</p> <p>Can you tell me what are the various steps involved?</p>		<b>ESR:</b> - create 4 balls using createSprite() - assign velocityX & velocityY to each ball - use bounceOff function to bounce the ball against edges - use bounceOff function to bounce the ball against each other
<p>Great!</p> <p>Why don't you share your screen and show me how are you going to code this challenge</p>		<i>The student shares his/her screen, opens the Student Activity, remixes the code and adds more code to it.</i>
 <b>Teacher ends slideshow</b>		
<b>Teacher Stops Screen Share</b>		
<b>STUDENT-LED ACTIVITY - 8mins</b>		
<ul style="list-style-type: none"> <li>● <b>Ask Student to press ESC key to come back to panel</b></li> <li>● <b>Guide Student to start Screen Share</b></li> <li>● <b>Teacher gets into Fullscreen</b></li> </ul>		

### ACTIVITY

1. Create 4 sprite balls that bounce off the walls and bounce against each other.
2. Experiment with other sprite properties and functions.

#### Step 3: Student-Led Activity (20 min)

Guide the student to create 4 sprite balls at four corners of the screen and assign them different names.

Observe the student, check for any typos she/he makes. Guide the student to write the code

Student opens [Student Activity 1](#)

He/She writes code to create 4 sprite balls at four different corners and gives them different names.

#### Code:

```

1 var ball1 = createSprite(5,5,10,10);
2 var ball2 = createSprite(395,395,10,10);
3 var ball3 = createSprite(395,5,10,10);
4 var ball4 = createSprite(5,395,10,10);
5
6
7 function draw() {
8
9   background("white");
10  drawSprites();
11 }
12
  
```

Guide the student to create edge sprites, assign velocity to each ball and make the balls bounce off the edges.

Observe the student, check for any typos she/he makes. Guide the student to write the code.

The student writes code to create edge sprites, assign velocity to each ball and make the balls bounce off the edges.

He/She runs the code to see the output.

**Code:**

```

1
2  var ball1 = createSprite(5,5,10,10);
3  var ball2 = createSprite(395,395,10,10);
4  var ball3 = createSprite(395,5,10,10);
5  var ball4 = createSprite(5,395,10,10);
6
7  ball1.velocityX = 2;
8  ball1.velocityY = 2;
9
10 ball2.velocityX = 3;
11 ball2.velocityY = 3;
12
13 ball3.velocityX = -2;
14 ball3.velocityY = -2;
15
16 ball4.velocityX = 2;
17 ball4.velocityY = -3;
18
19 function draw() {
20   background("white");
21
22   createEdgeSprites();
23
24   ball1.bounceOff(edges);
25   ball2.bounceOff(edges);
26   ball3.bounceOff(edges);
27   ball4.bounceOff(edges);
28
29   drawSprites();
30 }
31

```

*Guide the student to make ball1 bounce from other balls.*

*Encourage the student experiment with both bounce() and bounceOff().*

*Observe the student, check for any typos she/he makes. Guide the student to write the code.*

*The student writes code to make the ball1 bounce from other balls.*

*He/She runs the code and sees the output.*

**Code:**

```

4  var ball3 = createSprite(395,5,10,10);
5  var ball4 = createSprite(5,395,10,10);
6
7  ball1.velocityX = 2;
8  ball1.velocityY = 2;
9
10 ball2.velocityX = 3;
11 ball2.velocityY = 3;
12
13 ball3.velocityX = -2;
14 ball3.velocityY = -2;
15
16 ball4.velocityX = 2;
17 ball4.velocityY = -3;
18
19 function draw() {
20   background("white");
21
22   createEdgeSprites();
23
24   ball1.bounceOff(edges);
25   ball2.bounceOff(edges);
26   ball3.bounceOff(edges);
27   ball4.bounceOff(edges);
28
29
30
31   drawSprites();
32 }
33

```

*Guide the student to make the remaining balls bounce from each other.*

*Observe the student, check for any typos.*

*The student writes code to make all the balls bounce from each other.*

*He/She runs the code and sees the output.*

```
var ball1 = createSprite(5,5,10,10);  
var ball2 = createSprite(395,395,10,10);  
var ball3 = createSprite(395,5,10,10);  
var ball4 = createSprite(5,395,10,10);
```

```
ball1.velocityX = 2;  
ball1.velocityY = 2;
```

```
ball2.velocityX = 3;  
ball2.velocityY = 3;
```

```
ball3.velocityX = -2;  
ball3.velocityY = -2;
```

```
ball4.velocityX = 2;  
ball4.velocityY = -3;
```

```
function draw() {  
  
    background("white");  
  
    createEdgeSprites();  
    ball1.bounceOff(edges);  
    ball2.bounceOff(edges);  
    ball3.bounceOff(edges);  
    ball4.bounceOff(edges);  
  
    ball1.bounce(ball2);  
    ball1.bounce(ball3);  
    ball1.bounce(ball4);  
  
    //ball 1 is already bouncing off ball 2  
    ball2.bounce(ball3);  
    ball2.bounce(ball4);
```



```
//ball 3 is already bouncing off ball 1 and ball 2
ball3.bounce(ball4);

  drawSprites();
}
```

This looks amazing! You seem to have done pretty well in this challenge.

Let us quickly add comments in our code. Do you remember what comments are and why they are needed in our code?

*Complement the student for her/his efforts in the class.*

*The student explains how comments make the code readable.*

*He/She comments the code.*

```
//create 4 ball sprites
var ball1 = createSprite(5,5,10,10);
var ball2 = createSprite(395,395,10,10);
var ball3 = createSprite(395,5,10,10);
var ball4 = createSprite(5,395,10,10);
```

```
//assign velocity to our balls
ball1.velocityX = 2;
ball1.velocityY = 2;
```

```
ball2.velocityX = 3;
ball2.velocityY = 3;
```

```
ball3.velocityX = -2;
ball3.velocityY = -2;
```

```
ball4.velocityX = 2;
ball4.velocityY = -3;
```

```
function draw() {
```

```
//make the background white
background("white");

//create boundaries on the edges
createEdgeSprites();

//make the balls bounce off the edges
ball1.bounceOff(edges);
ball2.bounceOff(edges);
ball3.bounceOff(edges);
ball4.bounceOff(edges);

//make the balls bounce off each other
ball1.bounce(ball2);
ball1.bounce(ball3);
ball1.bounce(ball4);

ball2.bounce(ball3);
ball2.bounce(ball4);

ball3.bounce(ball4);

//draw the sprites
drawSprites();
}
```

**Teacher Guides Student to Stop Screen Share**

**Quiz time - Click on in-class quiz**

Question	Answer
<p>Select the correct syntax to show the ball sprite on the canvas</p> <p>A. <code>var ball= createSprite(200, 200,10,10); function draw() { background("white"); drawSprite(); }</code></p>	<b>A</b>

<p>B. <code>var ball= createSprite(200, 200,10,10);</code>  <code>function draw() {</code>  <code>background("white");</code>  <code>display();</code>  <code>}</code></p> <p>C. <code>var ball= createSprite(200, 200,10,10);</code>  <code>function draw() {</code>  <code>background("white");</code>  <code>show();</code>  <code>}</code></p> <p>D. <code>var ball= createSprite(200, 200,10,10);</code>  <code>function draw() {</code>  <code>background("white");</code>  <code>}</code></p>	
<p>Guess the direction in which the ball will move based on the following code-</p> <pre>var ball= createSprite(200, 200,10,10); function draw() { background("white"); ball.velocityX =-2; drawSprite(); }</pre> <p>A. Ball will move right.        B. Ball will move left.        C. Ball will move up        D. Ball will move down</p>	<p><b>B</b></p>
<p>Choose the correct code syntax to make the ball bounce off the walls.</p> <p>A. <code>var ball= createSprite(200, 200,10,10);</code>  <code>ball.velocityX=2;</code>  <code>ball.velocityY =3;</code></p> <pre>function draw() { background("white"); createEdgeSprites(); edges.bounce(ball); drawSprites(); }</pre>	<p><b>C</b></p>

```

}
B. var ball= createSprite(200, 200,10,10);
   ball.velocityX=2;
   ball.velocityY =3;

   function draw() {
     background("white");
     createEdgeSprites();
     ball.bounce(edges);
     drawSprites();
   }
C. var ball= createSprite(200, 200,10,10);
   ball.velocityX=2;
   ball.velocityY =3;

   function draw() {
     background("white");
     createEdgeSprites();
     ball.bounceOff(edges);
     drawSprites();
   }
D. var ball= createSprite(200, 200,10,10);
   ball.velocityX=2;
   ball.velocityY =3;

   function draw() {
     background("white");
     createEdgeSprites();
     ball.bounce();
     drawSprites();
   }

```

**End the quiz panel**

**WRAP UP SESSION - 5 Mins**




**Teacher starts slideshow**

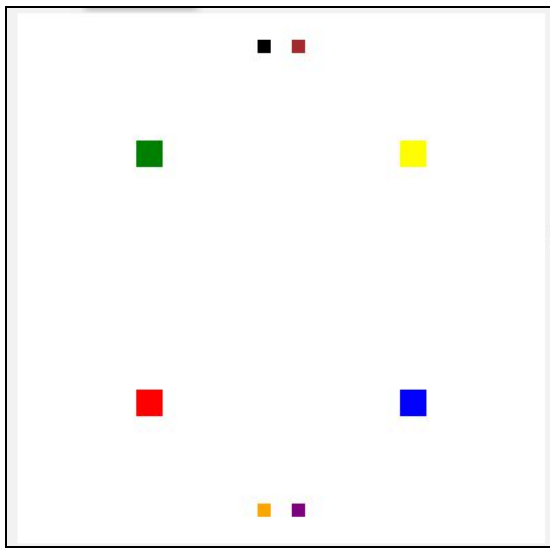
**from slide 19 to slide 28**

**Activity details**

**Solution/Guidelines**

**Run the presentation from slide 19 to slide 28**

<b>Following are the warm up session deliverables:</b> <ul style="list-style-type: none"> <li>• Explain the facts and trivias</li> <li>• Next class challenge</li> <li>• Project for the day</li> <li>• Additional Activity</li> </ul>		Guide the student to develop the project and share with us.
<div>  </div> <p>Teacher ends slideshow</p>		
<b>ADDITIONAL ACTIVITIES</b>		
<ul style="list-style-type: none"> <li>• Ask Student to press ESC key to come back to the panel</li> <li>• Guide Student to start Screen Share</li> <li>• Teacher gets into Fullscreen</li> </ul>		
Activity details		Solution/Guidelines
<b>Additional Activities</b>	<i>Encourage the student to experiment with <code>sprite.bounce()</code>, <code>sprite.bounceOff()</code>, <code>sprite.collide()</code> and <code>sprite.displace()</code> in their code and see how these functions give different behaviours to the ball.</i>	<i>The student modifies code to experiment and see the change in behavior of the ball when <code>bounce()</code>, <code>bounceOff()</code>, <code>displace()</code> and <code>collide()</code> are used.</i>
	<i>Allow the student to explore <code>Sprite.shapeColor</code> property of sprites.</i>  <i>Guide the student to create a trippy ball animation as below.</i>	<i>The student writes code to create a trippy ball animation with balls of different colors.</i>



	<i>How much of the Pong Game can you create on your own?</i>	<i>The student tries to code for the Pong Game paddles and the ball.</i>
	<i>What other kinds of animations can you create using the sprites we just learned?</i>	<i>The student attempts to create other kinds of animations using moving sprites.</i>
	<i>Guide the student to write the same code on p5 editor.</i>  <i>Guide the student to write the remaining code for bouncing balls in the p5 editor.</i>	<i>Student uses <b><u>Student activity 2</u></b> to write the same code in the p5 editor.</i>
<b>Code in p5:</b>		

```

1 <!DOCTYPE html>
2 <html lang="en" dir="ltr">
3   <head>
4     <meta charset="utf-8">
5     <title>Bouncing Ball</title>
6     <script src="p5.js"></script>
7     <script src="p5.dom.min.js"></script>
8     <script src="p5.sound.min.js"></script>
9     <script src="p5.play.js"></script>
10    <script src="sketch.js"></script>
11  </head>
12  <body>
13
14  </body>
15 </html>

```

```

1 var canvas, ball, edges
2
3 function setup(){
4   canvas = createCanvas(500,500)
5   ball = createSprite(5,5,10,10);
6
7   ball.velocityX = 2;
8   ball.velocityY = 2;
9 }
10
11 function draw(){
12   background("white");
13   edges = createEdgeSprites();
14
15   ball.bounceOff(edges);
16   // ball.bounceOff(edges[1]);
17   // ball.bounceOff(edges[2]);
18   // ball.bounceOff(edges[3]);
19
20   drawSprites()
21 }

```

Activity	Activity Name	Links
Teacher Activity 1	Teacher Playground	<a href="https://studio.code.org/projects/gamelab/nfIILEERcRuhBLd1hmhGcbGliKBI_z6AfTp8MniGVdk/edit">https://studio.code.org/projects/gamelab/nfIILEERcRuhBLd1hmhGcbGliKBI_z6AfTp8MniGVdk/edit</a>



Student Activity 1	Student Playground	<a href="https://studio.code.org/projects/gamelab/nlJRaRQEb5VjvgNy1VYXfFA-1d1XliJAo597WgM-tm0/edit">https://studio.code.org/projects/gamelab/nlJRaRQEb5VjvgNy1VYXfFA-1d1XliJAo597WgM-tm0/edit</a>
Student Activity 2	Additional Student Activity	<a href="https://editor.p5js.org/whitehatjr/sketches/P_UakkuC">https://editor.p5js.org/whitehatjr/sketches/P_UakkuC</a>
Teacher Reference	Trippy Ball Animation	<a href="https://studio.code.org/projects/gamelab/JLvJEjyiKnZsZa8dAzcg3UQ3m9SZ2HuhU_2WJf5I_S0/edit">https://studio.code.org/projects/gamelab/JLvJEjyiKnZsZa8dAzcg3UQ3m9SZ2HuhU_2WJf5I_S0/edit</a>
Teacher Reference visual aid link	Visual aid link	<a href="https://curriculum.whitehatjr.com/Visual+Project+Asset/PRO_VD/PRO+C2+Expert+Class+without+cues.html">https://curriculum.whitehatjr.com/Visual+Project+Asset/PRO_VD/PRO+C2+Expert+Class+without+cues.html</a>
Teacher Reference In-class quiz	In-class quiz	<a href="https://curriculum.whitehatjr.com/Visual+Project+Asset/PRO_VD/Pro-C2_Deepali.docx.pdf">https://curriculum.whitehatjr.com/Visual+Project+Asset/PRO_VD/Pro-C2_Deepali.docx.pdf</a>