PLEASE NOTE

END YOUR CLASS WITH WOW FACTOR.

Amaze your student with a FUN WITH TECH

Find the Details in VA

The 5 min activity can increase your chance of Student Renewal

Topic	**CONDITIONAL PROGRAMMING**
Class Description	**The student will create a paddle sprite and bounce the ball off it on collision. The student will learn about conditional programming and use if-else statements to control the movement of the sprite object.**
Class	**PRO-C3**
Class time	**55 mins**
Goal	• Write instructions to create a paddle sprite in the game and control it using the movement of the mouse. • Write bounceOff instructions such that the ball bounces back only from the selected edges.

Resources Required	• Teacher Resources
	  ○ Code.org login
	  ○ Laptop with internet connectivity
	  ○ Earphones with mic
	  ○ Notebook and pen
	  ○ Smartphone
	• Student Resources

	○ Code.org login ○ 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** **10 mins**

WARM-UP SESSION - 10 mins

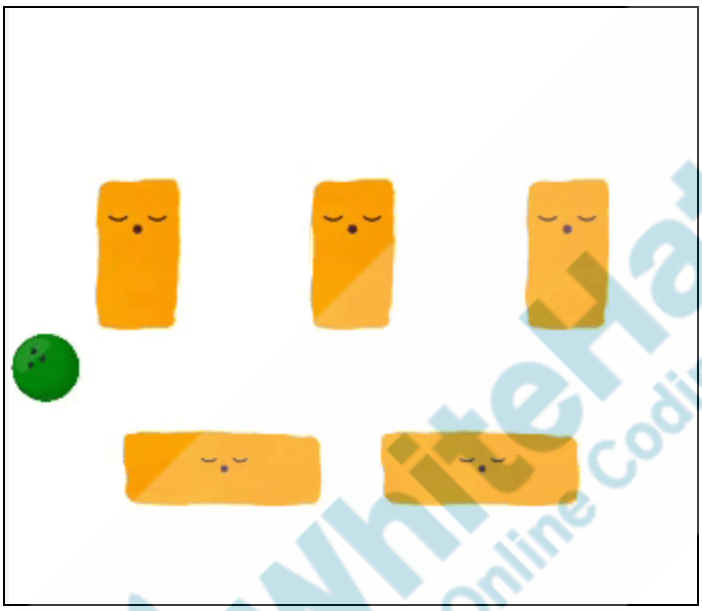
Teacher starts slideshow from slides 1 to 9
Refer to speaker notes and follow the instructions on each slide.

Teacher Action	Student Action
How have you been? Are you excited to learn something new? ***Run the presentation from slide 1 to slide 3.*** **Following are the warm-up session deliverables:** • Connect students to the previous class. • Warm-Up Quiz Session.	**ESR**: Varied Response. Click on the slide show tab and present the slides.

Q&A Session - Click on in-class quiz

Question	Answer
Which of the following statements can be used to make the edges sprites ?	**A**

A. createEdgeSprites() B. edges = sprites() C. edges() D. spriteEdges()	

A. Which of the following line(s) code is used to make the ball bounce off the vertical_box1 and vertical_box2?
 
A. vertical_box1.bounceOff(ball);
vertical_box2.bounceOff(ball);
B. ball.bounceOff(vertical_box1, vertical_box2);
C. ball.bounceOff(vertical_box1);
ball.bounceOff(vertical_box2);
D. ball.collide(vertical_box1);
ball.collide(vertical_box2);
	C

Continue the warm-up session	
Activity details	**Solution/Guidelines**

Run the presentation from slide 4 to slide 8 to set the problem statement. **Following are the warm-up session deliverables:** • Sprite properties • Use of Conditional Programming	Narrate the slides by using hand gestures and voice modulation methods to bring in more interest in students.
Teacher ends slideshow 	
TEACHER-LED ACTIVITY 1 - 10 mins	
Teacher Initiates Screen Share	
CHALLENGE • **Use of conditional programming.** • **Use of if-else statements.**	
Teacher Action	**Student Action**
In the last few classes, we have learned about variables, and written a few customized functions to change sprite properties. Today I will introduce you to **Conditionals** which help computers decide between different options and make a decision.	
Let us learn how to write conditionals in programming to check if a given condition is true or not. *The teacher explains the syntax to the child.* --- ***If(condition)*** ***{*** ***//action if the condition is true.*** ***}*** ***else***	

{ ***//action if the condition is false.*** ***}*** --- Let's write a simple compare function using the **'if-else statement'** to find the larger number out of two numbers. *Teacher opens Teacher activity 1 and makes the necessary changes to the code as shown below:*	

```
Workspace
1  function compare(num1, num2) {
2
3  if(num1 > num2)
4  {
5    console.log(num1+" is greater than "+ num2);
6  }
7  else
8  {
9    console.log(num2+" is greater than "+ num1);
10 }
11
12 }
13 function draw() {
14
15 }
16
17 compare(9,5);

Debug Console                    Debug Sprites: Off
"9 is greater than 5"
```

Did you notice how we have trained our computer to compare two numbers using the 'if-else statement' and comparison operator? Step 1: Check if **num1** is greater than **num2**. Step 2: If this is **true**, **num1** is greater than **num2** Step 3: Else **num2** is greater than **num1**. Let's try out a few more cases.	**ESR**: Yes.

Teacher calls the compare function with a few more different input values from the student. ****For faster kids, teacher also passes an exceptional case where both numbers are equal, **5,5** as shown below:*	

********** *Only if you have covered the last case of comparing two equal numbers 5,5(i.e. for faster kids)*** Did you notice the error in the last case when two numbers are equal? Our function is not sufficient to find out if two numbers are equal. That's a logical bug in our algorithm. Can you tell me how can we solve it in our function? We can include one more **if** statement at the beginning to train the computer to check if two numbers are equal or not. In the future classes, we will learn to write nested if conditions as well. **********	 **ESR**: Yes **ESR**: Varied.
Now that we know how to write if conditions, I have a challenge for you: Are you ready? Write a function to check if the number is positive or negative. Can you tell me what will be the condition for the same?	 **ESR:** Yes **ESR**: If the number is **greater** than **0** then it is a positive number else it's a negative number.
Teacher Stops Screen Share	
STUDENT-LED ACTIVITY 1 - 5 mins	
Now it's your turn. Please share your screen with me.	
• **Ask Student to press ESC key to come back to the panel** • **Guide Student to start Screen Share** • **Teacher gets into Fullscreen**	
CHALLENGE • **Write a function to check if the number is positive or negative.**	

- **Call the function multiple times.**

Teacher shows slideshow  **for slide 10**
Refer to speaker notes and follow the instructions on each slide.

Teacher Action	Student Action
Guide the student to open the Student Activity 1 link and login into code.org if he/she hasn't still. Guide the student to understand the input. This time we will have just **one** number as an input. So the number of parameters during function definition and calling will be only **one**. Guide the student to make changes in the code to write a function that checks if a number is positive or negative. 1. Compare the value of the input with value '**0**' 2. **If** the value is greater than '**0**', it's **positive**. 3. **Else** it's **negative**. *The teacher asks the student to make multiple calls to the function by giving positive and negative numbers as input in the function calls.* *Guide the student to make changes in the code to make multiple calls to the function by using different numbers as shown below:*	*The student opens the link on the platform.*

Workspace

```
1  function check(num1) {
2
3  if(num1 > 0)
4  {
5    console.log(num1+" is positive.");
6  }
7  else
8  {
9    console.log(num1+" is negative.");
10 }
11
12 }
13 function draw() {
14
15 }
16
17 check(5);
18 check(-9);
```

Debug Console | Debug Sprites: Off

```
"5 is positive."
"-9 is negative."
```

Awesome! You seem to understand 'conditions' pretty well. Let's proceed with our game development.	

Teacher Guides Student to Stop Screen Share

TEACHER LED ACTIVITY 2 - 10 mins

CHALLENGE

- **Learn to move the sprite object as per the mouse movement.**

Teacher starts slideshow  **from slides 11 to 13**

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

Teacher Action	Student Action

In the last class, we made the ball move and bounce off the edges. Today we will create a paddle in our game. But before that we will learn how to define event triggered functions like **mousePressed**, **keyPressed**, etc. *The teacher clicks on Teacher Activity Link 2 and opens the code from the previous class.* In our current game, the ball starts moving as soon as we run the program and there are chances that the player is still not ready to play. So let's add a feature in our game which makes the ball move only when the mouse is pressed. In game lab, there is already a function called **mousePressed()** which gets called whenever the mouse is pressed. Let's re-define this function as per our requirement. Go to the **Functions** tab in **Toolbox** and drag the '**Define a Function**' block in the workspace. Rename it to **mousePressed()** and move the already written **velocityX** and **velocityY** code at the top inside this function. *The teacher writes the code to define a function for **mousePressed()**.*	*The student observes and learns from the code.*
CODE:	

```
Workspace
1  var ball;
2
3  ball = createSprite(100,100,20,20);
4  ball.setAnimation("volleyball2_1");
5  ball.scale = 0.1;
6
7  createEdgeSprites();
8
9  function draw() {
10   background("white");
11   ball.bounceOff(edges);
12   drawSprites();
13
14 }
15
16 function mousePressed() {
17   ball.velocityX = 4;
18   ball.velocityY = 3;
19
20 }
21
22
```

OUTPUT:

Great! Did you see the ball only starts once I click the mouse button now?

Let's play around a bit more with our **mousePressed()** function.

In Game lab, we can use world objects to get the real-time mouse position on the canvas.
Can you see **mouseX** & **mouseY** properties under the **World** tab?

ESR: Yes

We can use the mouse position to change the position of the ball whenever the mouse is clicked. The teacher assigns **World.mouseX** and **World.mouseY** to **ball.x** and **ball.y** respectively inside the **mousePressed()** function.	**ESR**: Yes

Workspace

```
1  var ball;
2
3  ball = createSprite(100,100,20,20);
4  ball.setAnimation("volleyball2_1");
5  ball.scale = 0.1;
6
7  createEdgeSprites();
8
9  function draw() {
10   background("white");
11   ball.bounceOff(edges);
12   drawSprites();
13
14 }
15
16 function mousePressed() {
17   ball.velocityX = 4;
18   ball.velocityY = 3;
19
20   ball.x = World.mouseX;
21   ball.y = World.mouseY;
22
23 }
24
25
```

Did you notice whenever we click on the canvas, the ball begins to move from that point? Isn't it interesting? You will be using mouse coordinates to move the paddle today. Do you have any doubts? Great! It's your turn now to create a paddle sprite in the game.	**ESR:** Yes.

Teacher Stops Screen Share

STUDENT-LED ACTIVITY 2 - 10 mins

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

CHALLENGE

- **Create the player paddle object in the game.**

- **Ask Student to press ESC key to come back to the panel**
- **Guide Student to start Screen Share**
- **Teacher gets into Fullscreen**

Teacher starts slideshow  **from slides 14 to 16**
Refer to speaker notes and follow the instructions on each slide.

Teacher Action	Student Action
Ask the student to open their game which they had created in the last class. If a student doesn't find it, ask them to click on the Student Activity Link 2 and Remix the file.	*Student opens Student Activity 2 and clicks on **View code -> Remix**.*
You will start with the player paddle. How will you create the player paddle object in the game? *Teacher instructs the child to drag **createSprite()** instruction from the **Toolbox** and rename the variable to **paddle**.* *Teacher should now discuss the starting x and y position of the paddle as well as the length and height of the paddle while creating the paddle.* **var paddle = createSprite(200, 350, 120, 10);**	**ESR:** Using a sprite object.
What should be the '**x position**' of the player paddle? Should it remain static or should it change? *Assign the **mouse's x-position** to the **paddle's x-position** inside the **draw()** function.*	**ESR:** It should change with the mouse's pointer.
What if the player drags the paddle out of the screen from the right side or left side? *Instruct the child to try it out and figure out the error of the paddle going*	**ESR:** Varied.

out of the screen. How can we fix it? We can limit the movement of the paddle to the edges of the canvas. The x position starts from the center of the paddle. The width of the paddle sprite is **120**. So can you tell me what's the width of the sprite from the center? Now, to check that the paddle doesn't move outside the canvas: 1) Write an **if-else** condition to check if the paddle is at a distance of **60** from the **left side**. If **paddle.x** goes **less** than **60**, we will change it again to **60**. 2) Similarly, we will write a condition to check if the paddle's **x-position** is at a distance of **340** from the **right side**. If the **paddle.x** goes **above 340**, we will change it to **340** again.	**ESR:** Varied **ESR: 60** *The student makes the necessary changes in the code and runs the code.*
CODE:	

Workspace

```
1  var ball;
2
3  ball = createSprite(100, 100, 20, 20);
4  ball.setAnimation("volleyball2_1");
5  ball.scale = 0.1;
6
7  ball.velocityX = 0;
8  ball.velocityY = 0;
9
10 var paddle = createSprite(200, 350, 120, 10);
11
12 createEdgeSprites();
13
14 function draw(){
15   background("white");
16
17   paddle.x = World.mouseX;
18
19   if(paddle.x < 60){
20     paddle.x =60;
21   }
22
23   if(paddle.x > 340){
24     paddle.x =340;
25   }
26
27   drawSprites();
28   ball.bounceOff(edges);
29 }
30
31 function mousePressed(){
32   ball.velocityX = 4;
33   ball.velocityY = 2;
34 }
35
```

OUTPUT:

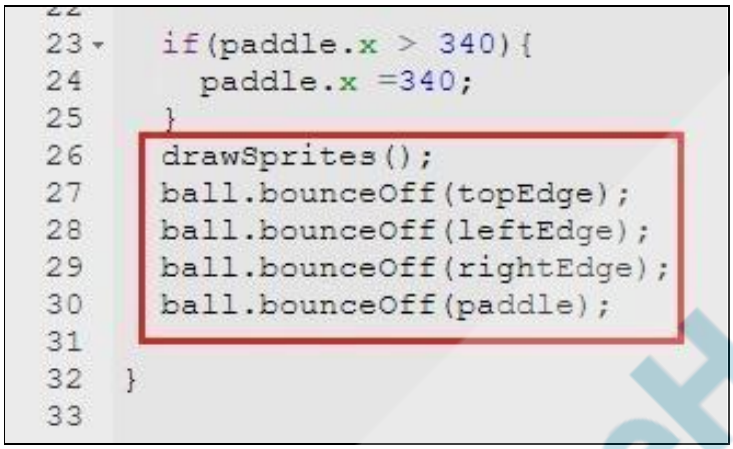
Good work. Our paddle is not going out of the screen now. Did you observe there's something wrong with our game? Can you tell me what it is?	**ESR:** The ball is not bouncing on the paddle.
You're right. Currently, the ball is bouncing only from the edges of the canvas, and it passes through the paddle. What changes can we do to our code to make it bounce from the paddle as well?	**ESR**: Varied.
We need to add the **bounceOff()** function such that the ball bounces on touching the paddle. Now, add the **bounceOff()** function.	*Student runs the code to see the output.*

CODE

```
  background( white );

  paddle.x = World.mouseX;

  if(paddle.x < 60){
    paddle.x =60;
  }

  if(paddle.x > 340){
    paddle.x =340;
  }
  drawSprites();
  ball.bounceOff(edges);
  ball.bounceOff(paddle);

}
```

OUTPUT

Awesome! Can you see that even if the paddle misses the ball, the ball comes back to the game by bouncing off the bottom edge? We should bounce the ball only from the three edges and leave the fourth one. Can you guess the three edges?	**ESR**: Yes. **ESR**: Varied

Teacher explains the four edges to the student: **topEdge**, **bottomEdge**. **leftEdge**, **rightEdge**. *The teacher instructs the student to bounce the ball off only from the left, right, and top.*	The student writes code to make the ball bounce from **left**, **right**, **top** edges.
CODE  22 23 if(paddle.x > 340){ 24 paddle.x =340; 25 } 26 drawSprites(); 27 ball.bounceOff(topEdge); 28 ball.bounceOff(leftEdge); 29 ball.bounceOff(rightEdge); 30 ball.bounceOff(paddle); 31 32 } 33	**OUTPUT**  Code Animation Reset
Teacher Guides Student to Stop Screen Share	
WRAP-UP SESSION - 10 mins	
FEEDBACK • **Appreciate and compliment the student for trying to learn a difficult concept.** • **Get to know how they are feeling after the session.** • **Review and check their understanding.**	
Teacher starts slideshow  **from slides 17 to 29** Refer to speaker notes and follow the instructions on each slide.	
Activity details	**Solution/Guidelines**
Run the presentation from slide 17 to slide 29. **Following are the WARM-UP session deliverables:** • **Explain the facts and trivias.** • **Next class challenge.** • **Project for the day.**	- - - - *Guide the student to develop the project and share with us.*

• **Additional Activity.**	
QnA Session - Click on in-class quiz	
Question	**Answer**

Can you identify the correct instructions for bouncing the paddle from the left and the right edge of the canvas?
 	
A. paddle.bounceOff(left);
paddle.bounceOff(right);
 	
B. paddle.bounceOff(top);
paddle.bounceOff(bottom);
 	
C. paddle.bounceOff(leftEdge);
paddle.bounceOff(rightEdge);
 	
D. paddle.bounceOff(topEdge);
paddle.bounceOff(bottomEdge);
	C

The following piece of code implements _____ in the program?
 	
`If(condition)`
`{`
`//action if condition is true.`
`}`
`else`
`{`
`//action if condition is false.`
`}`
 	
A. function
B. Conditional programming
C. loop
D. None of the above
	B

Which properties can we use from the world objects to get the real time mouse position on the canvas? **A. mouseX and mouseY** B. MouseX and MouseY C. mousex and mousey	**A**

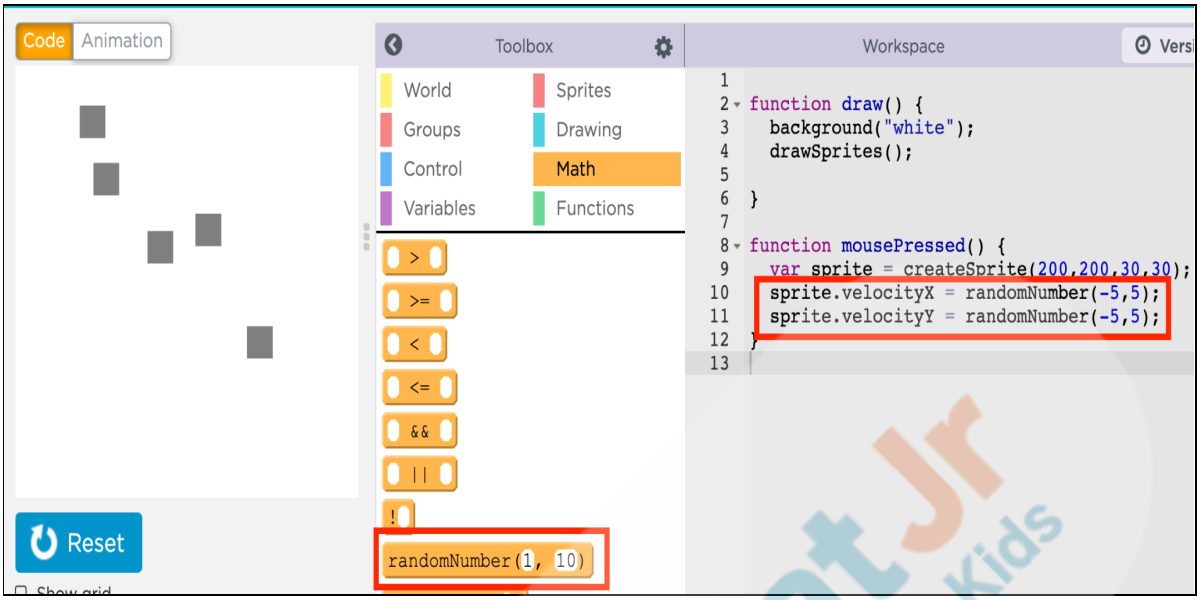
D. Mousex and Mousey	
End the quiz panel	
FUN WITH TECH FOR STUDENT TO PERFORM (MUST)	
• **Ask the student to press ESC key to come back to the panel** • **Guide the student to start Screen Share** • **The teacher gets into full screen**	
You have successfully completed today's challenge. You were really quick. It is now time to open our FUN WITH TECH.	
The teacher can guide the student to open a link from Student Activity 2. What do you see here? Yes, have you ever wondered how pilots learn to fly the plane? They do not start flying actual planes, but they use a simulator to learn to fly. This is an example of a Virtual Flight Simulator you will be developing during the course. *The teacher can share the following instructions about how to play this.* 1. Click on the screen to drag around. You'll see the different parts of the 3D game world. 2. You'll see a plane flying around the terrain. You'll be the pilot of this plane to move it around the terrain. 3. Use the arrow keys to move the plane around. 4. The UP arrow key will move the plane up. 5. The DOWN arrow key will move the plane down. 6. The LEFT arrow key will move the plane to the left. 7. The RIGHT arrow key will move the plane to the right. 8. Collect the rings floating around to increase the score.	*Student opens Student Activity 2.* **ESR:** An aeroplane.

9. You'll have 3 mins to collect 20 rings. 10. You'll lose the game if the plane collides with the bird. *While students is playing the game Teacher can mention:* It is a 3D VR game which is created using Aframe web framework built over HTML and the language used is again JavaScript. VR stands for Virtual Reality. VR is the use of computer technology to create a 3D simulated environment. Many VR games are mainly built for play stations, Oculus rift and Google cardboard. Some of the famous games are Shooting Games, Star Wars: Squadron, Half Life. Great! For now, you can stop sharing the screen and Let's move ahead. Did you enjoy playing the game? You will also build such a game in your advanced/future classes.. For now, you can stop sharing the screen and Let's move ahead. *For teacher reference: this game will be created in classes 153.*	**ESR:** Yes.
You get Hats off for your amazing performance today. Alright, we seemed to have a lot of learning in the class today. In the next class, we will create bricks in the game so that the player can break them using the ball. Isn't that interesting!	Make sure you have given at least 2 Hats Off during the class for:   

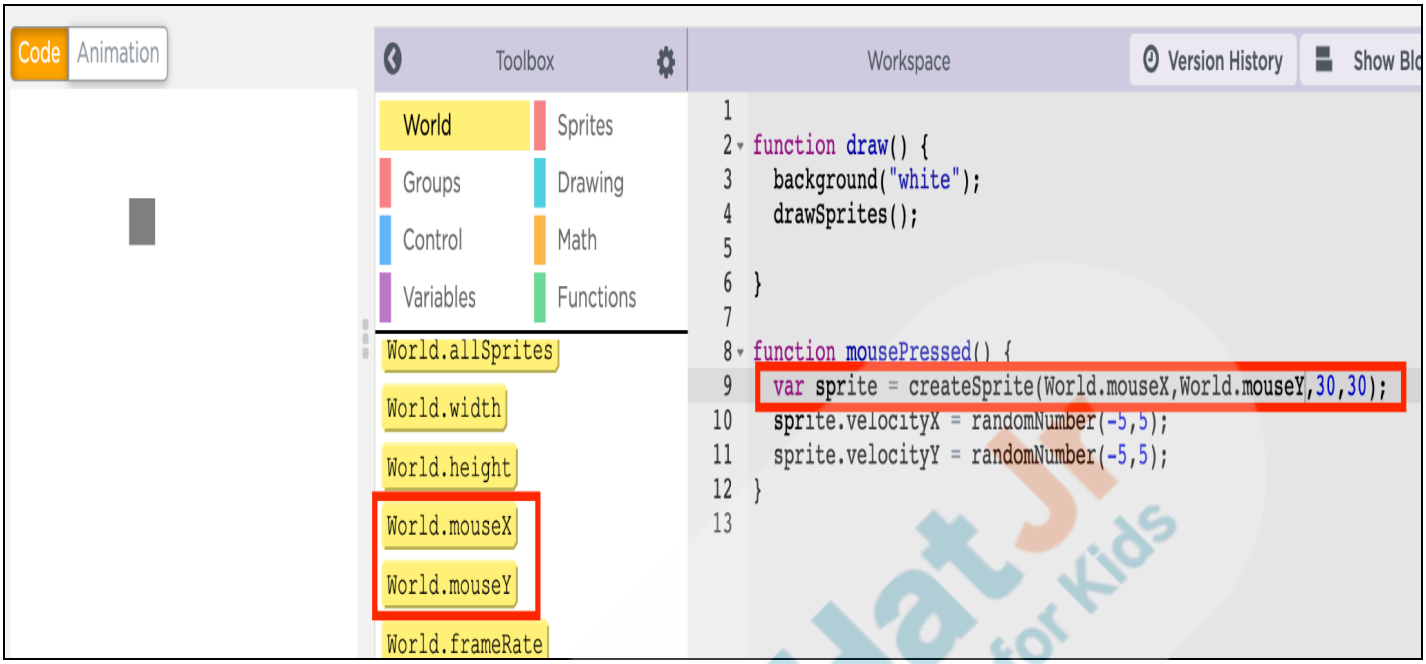
Project Overview **UNDER THE WATER** **Goal of the Project:** In Class 3, you learned to use Conditional Programming (if statements) to add control to the game elements. Here, you will design an underwater scene with moving fishes. In this project, you will have to practice and apply what you have learned in the class and create moving fishes inside the scene and make them reappear using the 'if condition'. **Story:** Daisy loves to play with fishes and spends hours watching them swim in the house aquarium. Her Dad accompanies her and teaches her how to clean the fish tank and feed the food to the fishes. Daisy has now learned conditionals and is eager to try her hand at creating an underwater scene where fishes keep swimming endlessly and show it to her Dad. Will you help Daisy build such a scene? I am very excited to see your project solution and I know you will do really well. Bye Bye!	*The student engages with the teacher over the project.*
Teacher ends slideshow 	
ADDITIONAL ACTIVITY **Teacher to share the screen**	
Let's play around with the sprites and also revise what we have learned till now. To make it fun, I will tell you the task, and you have to help me with the code.	Student observes.

Are you ready? *The teacher clicks on Additional Activity and opens the code from the previous class.*	**ESR**: Yes
Task 1: Whenever I click the mouse on the canvas, it should create a new sprite of size **30*30** at the **center**. Can you tell me which function is used to create a new sprite? Great! Tell me in which function I should write this instruction? Good job. I am already impressed. *Teacher does the necessary changes in the code and runs it.*	**ESR**: **createSprite(200,200,30,30)** **ESR**: **mousePressed()**
	
Now let's assign velocity also to this new sprite so that they fire away as soon as they are created. Which property shall we use for the same? *Teacher makes necessary changes in the code and runs the code.*	**ESR**: **velocityX** & **velocityY**

Did you observe on every mouse click, there is a new sprite being created at the center which moves away in the same direction every single time?	**ESR**: Yes.
Wouldn't it be fun if every sprite fires away in a different direction?	**ESR**: Yes.
Task 2: We need to make the **velocityX** and **velocityY** randomly generated in the program instead of giving a fixed/constant value. In programming, we have a random function for the same. Can you look for a **'randomNumber'** function in the **'Maths'** tab on my screen?	**ESR**: Yes.
A **randomNumber** function generates a random number between the two numbers. In our example, we will give random values between **-5** to **5** to **velocityX** & **velocityY**. *Teacher makes the changes in the code and runs it.* We click on the canvas to see multiple sprites firing in different directions.	Student observes the output.



Did you notice every time the new sprite is created, it now moves in different directions at different speeds?	**ESR**: Yes
Task 3: Now to create the new Sprite at the position where the mouse is clicked and not at the center. How can we change the position of the sprite? In Game lab, we can use world objects to get the real-time mouse position on the canvas.	**ESR**: By changing the x, y value of sprite.
Can you see **mouseX** & **mouseY** properties under the **World** tab?	
The teacher replaces the parameters **200**, **200** in **createSprite()** by **World.mouseX** and **World.mouseY** in the same order and runs the code.	**ESR**: Yes. The Student observes and learns.



Activity	Activity Name	Links
Teacher Activity 1	Compare Function	https://studio.code.org/projects/gamelab/WzAFOu77Tm1W2uWEzInmdq3QEDinE4OqnxNyQ2912hg
Teacher Activity 1 (Ref Code)	Compare Function	https://studio.code.org/projects/gamelab/WzAFOu77Tm1W2uWEzInmdtlucfVLc4id0t3RB_3DlpI

Teacher Activity Link 2	Mouse Triggered Events	https://studio.code.org/projects/gamelab/MUXcKWMziOz4h5REjG0f5M7xC9MufN-X_jzRcDfftFo
Teacher Activity Link 2 (Ref Code)	Mouse Triggered Events	https://studio.code.org/projects/gamelab/fNykX4SJylrTRciOW5am_ohE0-2IA-2_YdD_DVA_K0A
Student Activity Link 1	Check number is positive or negative	https://studio.code.org/projects/gamelab/fNykX4SJylrTRciOW5am_kWbgaW0YqZx3GqKZ70CvoU
Teacher Ref Code	Check number is positive or negative	https://studio.code.org/projects/gamelab/fNykX4SJylrTRciOW5am_jZT0GSrGD0Iha688OwhANc
Student Activity Link 2	Player Paddle	https://studio.code.org/projects/gamelab/vyCtLgHso8Ij4ftxSWARmQxzMiS6YVXpjK4T9gugUVQ
Teacher Ref Code (Player Paddle)	Player paddle	https://studio.code.org/projects/gamelab/vyCtLgHso8Ij4ftxSWARmX8J6X7j7S0hQUbJ7fKToAo
Teacher Reference visual aid link	Visual aid link	https://curriculum.whitehatjr.com/Visual+Project+Asset/PRO_Fun+with+tech/BJFC-PRO-V3-C3-withcues.html
Teacher Reference In-class quiz	In-class quiz	https://s3-whjr-curriculum-uploads.whjr.online/ccfbf895-8a34-4380-977b-842e1623e062.pdf
Additional Activity	Boilerplate	https://studio.code.org/projects/gamelab/Dr6fXTAVu_5ITFpevLUV7FZuOaxzdABwlX8J2tytP98
Additional Activity (Ref code)	Generate Sprites at Random Position	https://studio.code.org/projects/gamelab/Dr6fXTAVu_5ITFpevLUV7PBK3ojZTWvGslRbdfAe_yE
Teacher Activity 4	FUN WITH TECH	https://procodingclass.github.io/virtual-flight-simulation/
Student Activity 2	FUN WITH TECH	https://procodingclass.github.io/virtual-flight-simulation/