Topic	COMPUTER PROGRAM
Class Description	**This class establishes the importance of learning how to code in today's context of robots powered by Artificial Intelligence. The student starts designing the game "Breakout" by drawing and positioning shapes of the game objects.**
Class	**PRO-T**
Class time	**50 mins.**
Goal	● Establish the philosophy behind the course. ● Interpret a program as a set of instructions given to a computer to do a particular task. ● Understand how instructions given to a computer are different from instructions given to a human. ● Start designing a simple Breakout Game. Create the objects of the game and animate the paddle.
Resources Required	● Teacher Resources ○ Earphones with mic ○ A water bottle ○ Notebook and pen ○ Table and Chair ○ Laptop Charger ○ Internet Speed: Download Speed >=20Mbps ● Student Resources ○ Laptop with internet connectivity ○ Earphones with mic ○ Notebook and pen ○ Internet Speed: Download Speed >20Mbps ● Etiquettes ○ Have a clean and clear background ○ Always keep your camera on ○ Sufficient Lighting on your face(light should be facing your face and not your back) ○ Keep your phone on silent(Refer How To Ace Trial Class Section for more details)

Class structure	Warm-Up Teacher-Led Activity Student-Led Activity Wrap-Up	10 mins 15 mins 15 mins 10 mins

WARM-UP SESSION - 10 mins

CONTEXT

- **Make a connection with the student.**
- **Interpret a program as a set of instructions issued to a computer to do a particular task.**
- **Understand how instructions issued to a computer are different from instructions issued to a human.**

Teacher Action	Student Action
Hi <student's name>! I am <Instructor's name>. I am going to be your instructor in this program. Let's get to know each other a little bit before we start. ***Note***: *Encourage the student to share about him/herself, his/her interests, etc.* *Also share a little about yourself.* First, I will share a brief introduction about myself. I would love to know more about you. I have done <XYZ certification > and my hobbies are <Teacher's interest areas>. <Student Name> Please tell me about which grade are you in and which areas interest you? *Wait for their response.*	*The student shares about himself/herself, about his/her interests, etc.*

Also, can you tell me your expectations from this class?	*The student shares about his/her expectations from the class.*
Set the Agenda: *Ask the student to invite the parent to join the class. In case the Parent is not present, or if they are busy, ask them to join in the last 10 minutes if possible.* Let me quickly tell you **what we are going to do in this Trial Class**: • I will be teaching you the basics of JavaScript - which is a very popular programming language that can be used to design websites, mobile applications and games. • You will start building a Single Player Breakout Game App using JavaScript. • Towards the end I will be presenting the course structure and student outcomes which will help you understand exactly what we offer.	*The student invites the parent to join the class.*
Let's start with a quick ice-breaker activity. Please click on the  button in the bottom right corner of your screen to start the question. A question will pop up on your screen and the student's screen. Click the  button so that the Question moves to the bottom of your screen and initiates a conversation with the student based on the question.	

After the conversation is over, click on the  button to close the question and continue with the class.	
So <student's name>, why do you want to learn to code? *Explore all the reasons why the student wants to learn to code. Some things the student might talk about are robotics, creating games, writing software and apps, and so on.* *Appreciate the reasons given by the student by using words like "amazing", "awesome", "cool" and so on.*	*The students shares his/her reasons.*
I would also like to add a reason why anyone would want to learn to code today. Have you heard of "**Artificial Intelligence**" and how computers are becoming as smart as humans? Today, in workplaces, we mostly work with other humans more than computers. However, 10-15 years later, we might be working with computers and robots around us more than humans on a daily basis. Can you tell me some real life examples where robots can be used to help humans? ● Robots can be used for constructions and building houses or making a road over the mountains. ● Robots can help patients to carry them around in the hospital. ● Robots can help us teach. ● Robots can help us secure our country, even our own houses. ● Robots can be used to avoid crimes.	**ESR:** Varied.

And of course we would want to live in a world where everyone can buy a robot which can do our household work too. Well, you can now see how computers and robots will change the way we live in the future, so we must know how to give instructions to the computer to get things done. Giving instructions to the computer is very different from giving instructions to a human. Let me give you **a demo** to help you understand how issuing instructions to a computer is different from issuing instructions to a human. *Place a water bottle next to you so that it is visible to the student through the webcam.*	
I am a human. Give me instructions to drink water from this bottle. *Follow the simple instructions given by the student to drink water from the bottle.* Now imagine I am a robot. Now give me instructions to drink water from the bottle. *Follow all the instructions literally.* *Make the situation hilarious for the student.*	**ESR:** Drink water from this bottle. **ESR**: Some examples: ● The student might ask you to drink water from the bottle. You will try to drink water without opening the bottle. ● The student might ask you to open the bottle. You will try to uncap the bottle without rotating the cap first.

	• The student might ask you to rotate the cap of the bottle. You will try to rotate the cap in the clockwise direction - which tightens the cap further.
Q: What do you think was the difference between giving instructions to a human vs giving instructions to a robot/computer? *Encourage the student to decode the difference between instructions given to humans vs instructions given to robots.* **A:** A robot needed more precise/exact instructions to complete a task. A human could understand us without the exact instructions. If we roughly summarize, the steps to ask a robot to drink water from a bottle can follow this order: • Identify the bottle on the table. • Pick up the bottle. • Rotate the cap/lid in the counter-clockwise(or anti-clockwise) direction until the cap loosens up. • Pull the cap/lid up and place it on the table. • Bring the bottle near your mouth and tilt the bottle to drink the water.	*The student tries to tell the difference between instructions given to humans vs instructions given to robots.*
Did you see how you struggled to give **exact instructions** to the robot? Well, while giving instructions to a robot/computer, we have to keep three things in mind:	**ESR**: Yes.

• First, the **instructions must be exact**, that means we must tell exactly what we want the computer to do. • Second, the **instructions must follow a proper sequence**. We should know which step to be performed first and what next. Like before we start drinking the water from the bottle we must open the cap/lid of the bottle. • Third, the **instructions must be in a language that the computer can understand**. We understood the first two points with the demo. Now, to understand the third point, can you tell me if computers can understand English? All around the world, humans speak different languages like Spanish(in Spain), French(in France), Hindi(in India), Mandarin(in China) in different parts of the world. We use these languages to communicate with each other. Using these languages, we can make another human understand what we want to say to them. Now, we both can speak <name of the common native language the student and teacher speaks>. But if we want to speak to Chinese(or any other country other than the student's and teacher's own country) people, we need to learn Mandarin, right? Similarly, if I want to give instructions to a computer, I need to learn a language that can be understood by computers.	**ESR**: Yes/No. **ESR**: Yes.

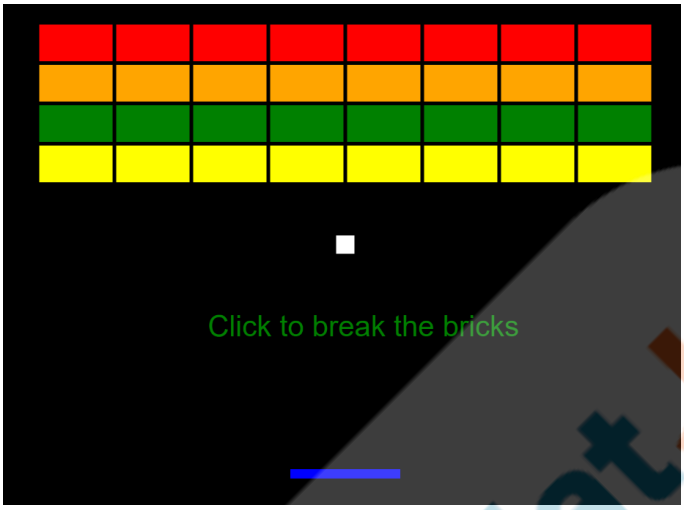
Certainly, **computers cannot understand instructions given in any human language**. What do you think, which languages are understood by computers? *Encourage the student to share any computer languages they might know or use.*	 **ESR**: Varied. *The student might mention JAVA, HTML, JavaScript, Python, etc.*
We are going to learn one such language in which we can give instructions to the computer- **JavaScript**. In fact, we are going to use this language to give instructions to the computer to design a game for us, create mobile applications and code a real satellite as well! Yes, you heard that right! CODE A REAL SATELLITE live from your laptop to outer space. Before going ahead, I want to ask you something. Let me share my screen, with this you will be able to see what's happening on my screen.	
Teacher Initiates Screen Share	
IMPORTANT	***Note**: Share your screen & Open [Teacher Activity 7 (Did you Know?)] and ask questions present in the slides to build the excitement.*
	IMPORTANT

IMPORTANT	***Note**: Ask questions present in the slides to build the excitement and wait for the response!* Yes, a "**satellite**". And you know what? In our **Space Explorer Program**, you will be able to communicate & code a real satellite,click a picture of the earth, predict weather or natural disasters and you can even track the position of the satellite. Sounds like a superpower, isn't it?	***ESR**: We can watch LIVE NEWS* with the help of a satellite.

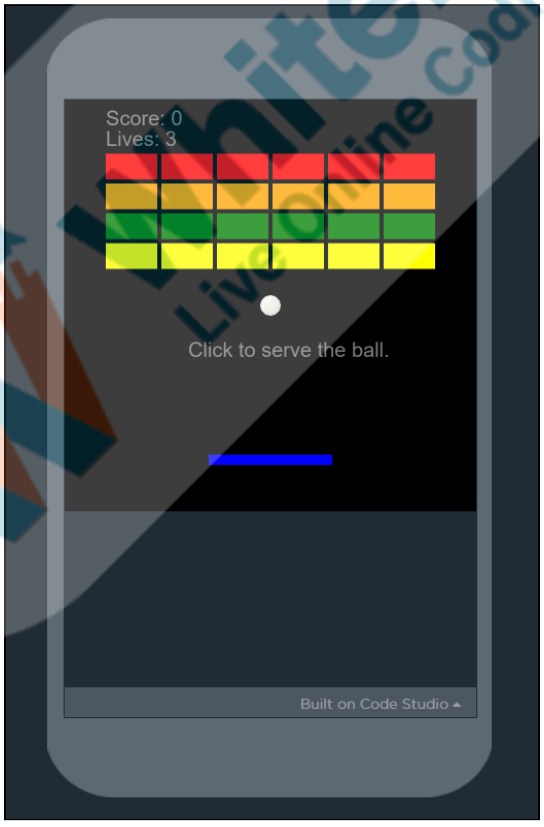
!! IMPORTANT SOP !!		
CHECK THE TRIAL STUDENT Ad type displayed SOURCE ON THE DASHBOARD **If the Ad type displayed = Roblox, then check whether the student has installed Roblox already.** **If installed then do the Roblox Activity below, else carry on with the rest of showcase and coding activities.**		
Roblox Activity	*Check the Student UTM, has the child signed up for a trial class by clicking a Roblox Ad?* *If **Yes**, Open **Teacher Activity 8** and **Play the Roblox Game*** *If **No**, Skip Teacher Activity 8*	**Roblox Activity**
TEACHER-LED ACTIVITY - 15 mins		
ACTIVITY • **Introduction to code.org platform.** • **Understand how to write a program to display the rectangular shape.** • **Learn by experiment how to change dimensions and move the shape.**		
Teacher Action		**Student Action**
We will start with learning how to code through designing games. Games offer us a lot of interesting and complex problems to solve as a programmer. After we learn how to design games, we will learn how to make mobile apps, and then we will learn about using Machine Learning (ML) and Artificial Intelligence (AI) in our programs.		

We will start with building a simple game and then progressively build complex games after every few classes. Let me show you the first few games we will be building in the course. *The teacher shows the links for the 4 games.* ● *T-Rex Runner [Teacher Activity 1]* ● *Pirate Invasion [Teacher Activity 2]* ● *Multiplayer Car Racing Game [Teacher Activity 3]* ***How to play a Pirate Invasion Game?*** ● *To play the Pirate Invasion game, use the **left and right arrow keys** to change the angle of the cannon, and use the **down arrow** key to shoot the cannonballs at the ships.* ***How to play a Multiplayer Car Racing Game?*** ● *The first time you are using a multiplayer car racing game you have to enter your name and generate a secret word.* ● *Share the secret word with the student.* ● *You'll have to generate a new secret word for each class.*	*The student observes.*
Before we can start coding, let's look at coding the simple Breakout Game! *The teacher can open the Breakout Game [Teacher Activity 4] for the student.*	

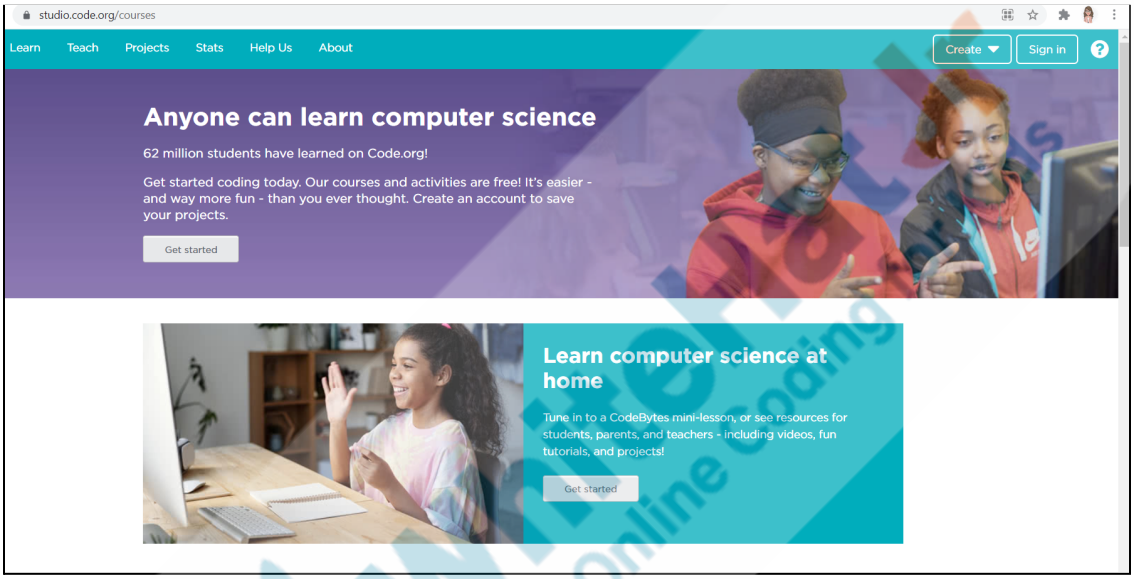
***Note 1**: Below is the image of the basic version of the Breakout game.*
***Note 2**: Click on the screen to start the game with ball movement.*

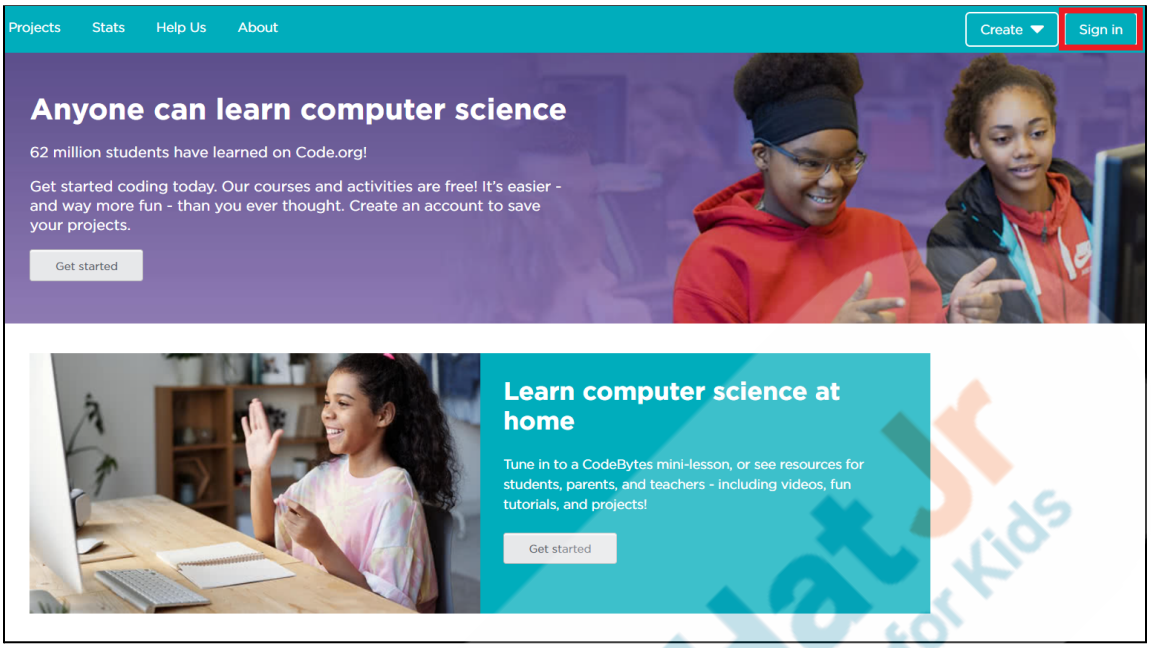


***Note**: Below is the image of the full version of the Breakout game with more features like scores and lives which students will learn to create in the upcoming classes.(This image is only shown for teacher reference.)*

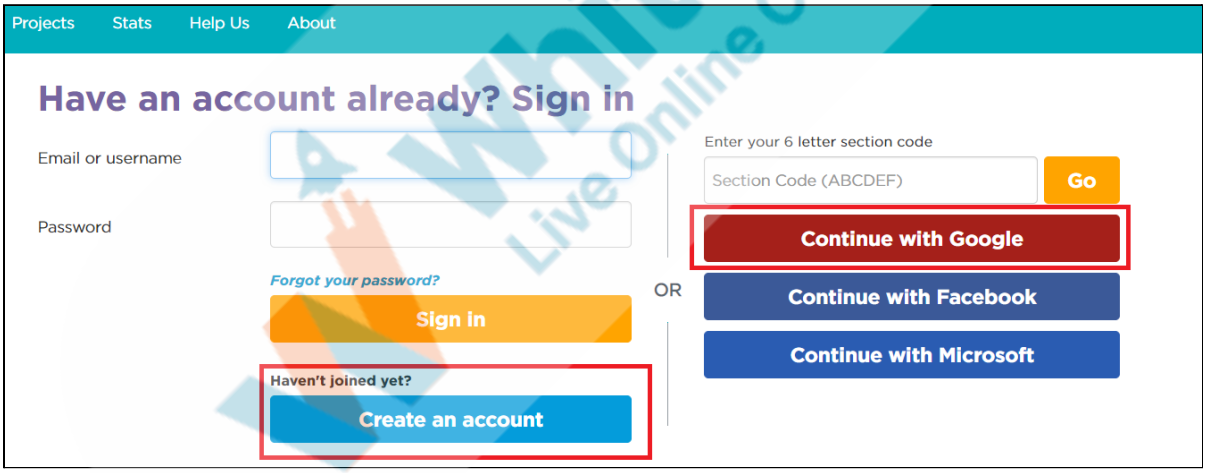


Even a simple game like this one has a lot of challenging things to program. Can you name what are the different components of this simple game? *The teacher should show/tell about the functionalities of the game which do not get covered in the student response.* *Use the following text to help you fill the gaps.* If you look at the game, there are many things in it. There are bricks and there is one paddle controlled by the mouse, there is a ball which is moving, and the ball is bouncing off the walls, there are sounds....so many things! Once we know what is happening in the game, where should we start? Suddenly the game appears to be complex now! While programming, we often have to deal with complex problems. We approach complex problems by breaking them down into several smaller simpler problems that we can solve. Remember, whenever you have a difficult and complex problem to solve, you must break it down into smaller problems - each of which is simple enough to be solved alone.	**ESR**: ● There are colorful bricks at the top which are moving downwards. ● There is a ball which is bouncing around. ● Once the ball hits the bricks at the top, the brick hit by the ball vanishes. ● There scores earned every time the ball hit the brick. ● There is a player paddle that moves whenever we move the mouse. **ESR:** Varied.

What is the simplest thing we can do in this game? Let's start by writing code to draw these objects. To write computer programs we are going to use an online platform called code.org *The teacher opens **[Teacher Activity 5]***	**ESR:** Draw the shapes of the paddle and the ball.
	
The teacher signs in if she already has a code.org account or creates a code.org account when using the code.org for the first time. • Click on the "**Sign in**" button in the top right corner.	

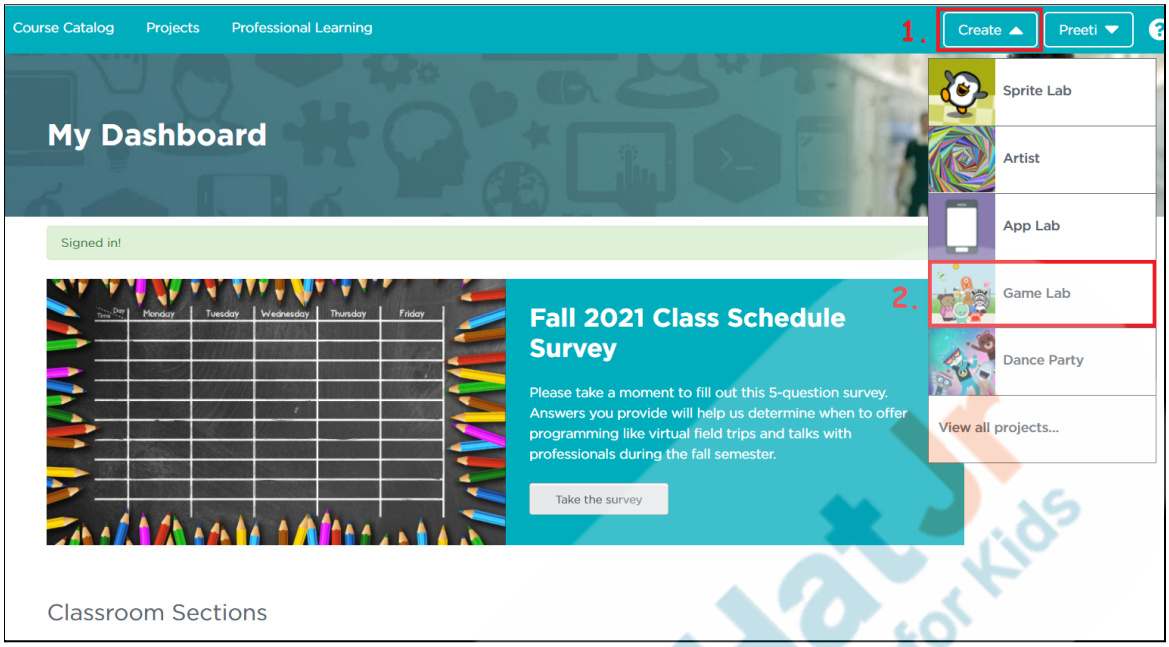


- Click on "**Create an account**", when using code.org for the first time after, and then **Sign in**.



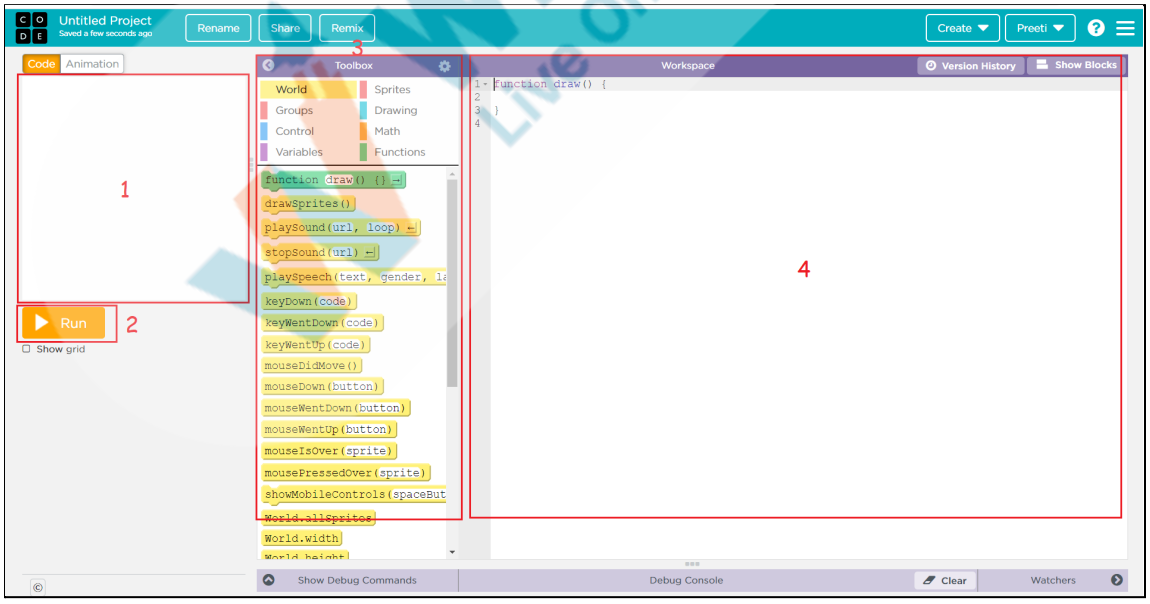
We will be using the **Game Lab** platform of code.org to develop simple games.

- On the code.org Dashboard:
    1. Click on **Create**(1 in image below)
    2. Click on **Game Lab**(2 in the image below)



Let's understand the platform before we can write the code here.

Note: *Teacher shows the code.org interface(explains the section 1, 2, 3 and 4 in the image below)*



Understanding the code.org interface:

1. **Code Output Section**: To see the result of the program.

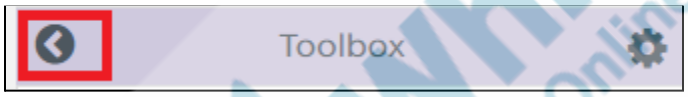
2. **Run/Reset Button**: Run Button is used to start the program.

   And **Reset Button** is used to stop the program, if the program is already running which can be shown later while doing the activity.

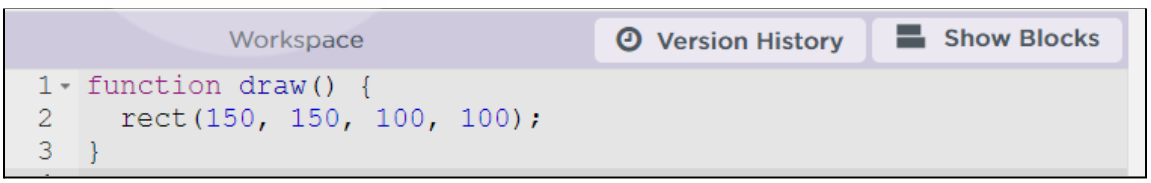
3. **Toolbox**: Toolbox is a helper box to use some **code blocks** directly which we see later.
   We can use the arrow on the toolbox to hide/show it.

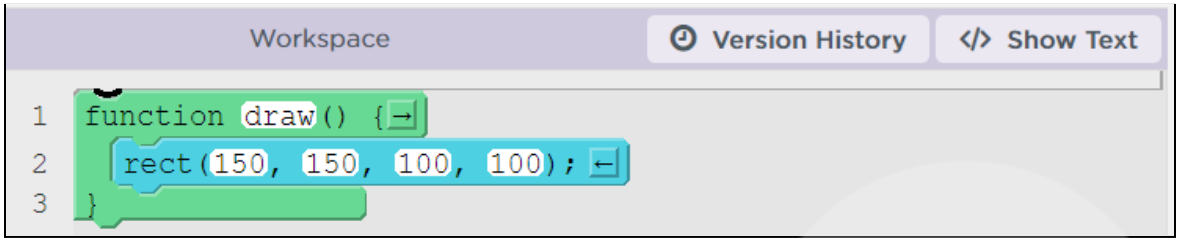
4. **Workspace**: Workspace is used to write the code in either the **Text format** or the **Block format**.

   These are just two different types of representation of the code lines in the code.org platform.

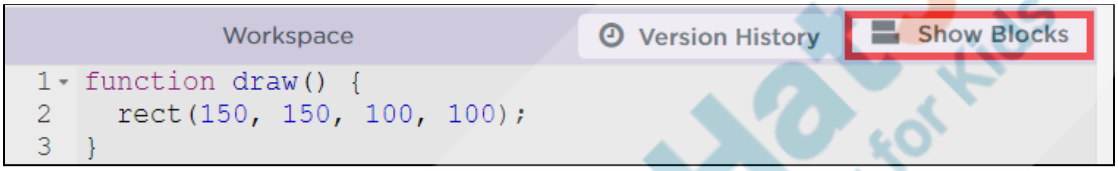
Text format:



Block format:

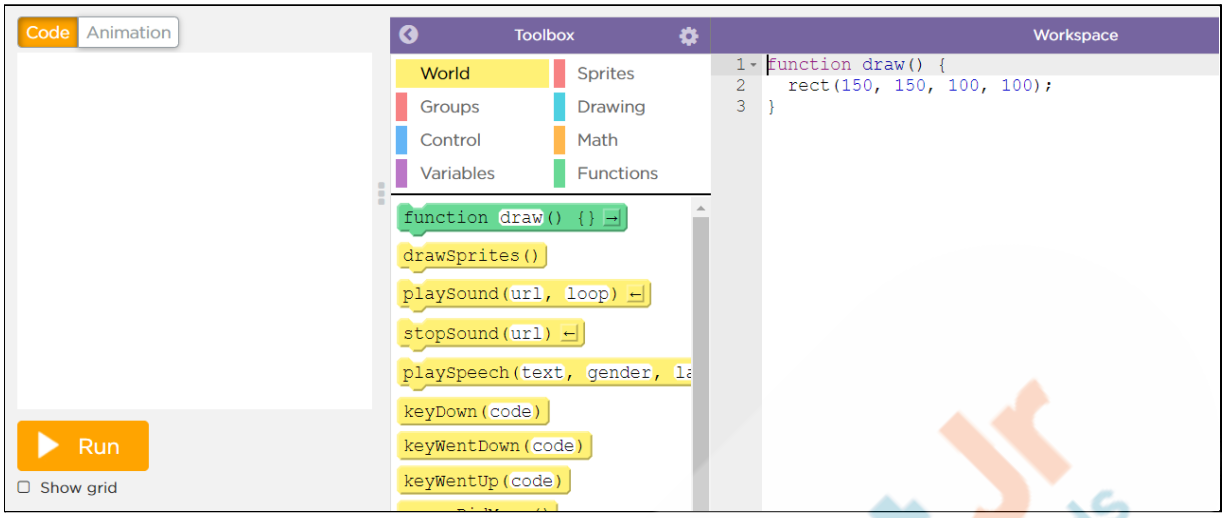


***Note**: Click on the **"Show Blocks" if you are in Text format mode** to see the Block format of the same code, or **"Show Text" if you are in the Block format mode**.*



Now we will start writing the code.

The teacher opens **Teacher Activity 6**.
*Make sure you are **Signed-in** to access the project link*.



Do you remember what game objects we need for the Breakout Game?	**ESR**: We need **bricks**, a **ball** and the **paddle**.
Amazing!	
Can you tell me what the shape of those objects is?	**ESR**: All the objects are of rectangle shape.
Superb!	
Now to draw a rectangle we need to give instructions to the computer.	
What do you think the **rect()** instruction does?	**ESR:** It will draw a rectangle.
Yes. Correct!	
And do you see a symbol at the end of the **rect()** instruction. This is a semi-colon(**;**). In JavaScript, a semi-colon(;) represents the end of an instruction, like a period(full stop) at the end of a sentence in English.	
Although, in JavaScript we will be able to run the program without semicolon but in some programming languages the	

instructions will not run without the use of semicolon at the end of the instructions. Now let's run the code. *Click on the **Run** Button to run the code.*	
	
Great! We have a rectangle on the screen. But we want a rectangle of different sizes for the bricks, the ball and the paddle, right? How do we do that? Perfect!	**ESR:** Yes. **ESR:** We can change the numbers inside the rectangle instruction.

Can you tell me what these numbers mean?

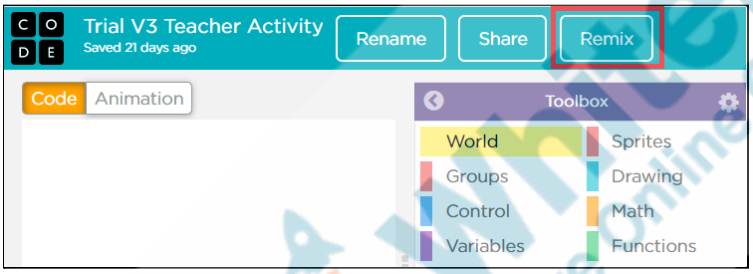
*The student takes some guess about what the numbers inside **rect()** mean.*

You know, the best thing about coding is that you don't have to wait for someone else to give you the right answers.

You can experiment, try things out, see the result and find the right answers on your own.

Let's experiment with these numbers ONE BY ONE and try to find out what they mean.

- Click on the "**Remix**" button to make the Workspace editable.

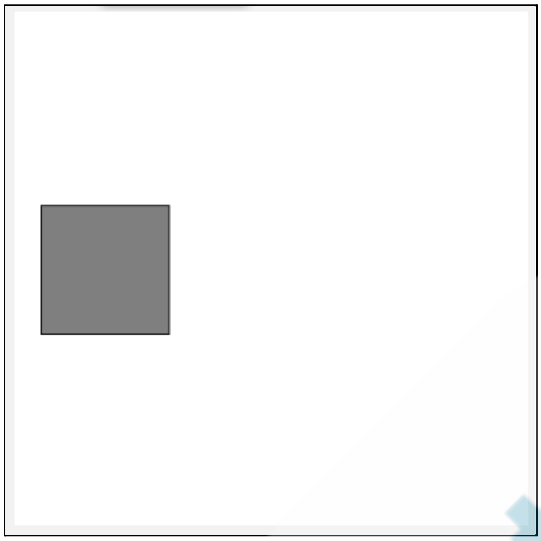


The student gives input on what the first number could be.
He/She reduces the first number.

Take input from the student on what the 1st number could be to shift the box.

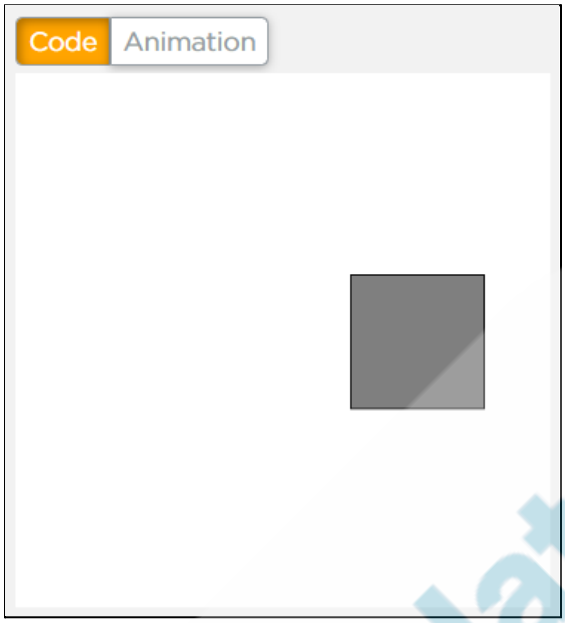
The teacher runs the code with the first number reduced.

```
function draw() {
  rect(20, 250, 10, 100);
}
```



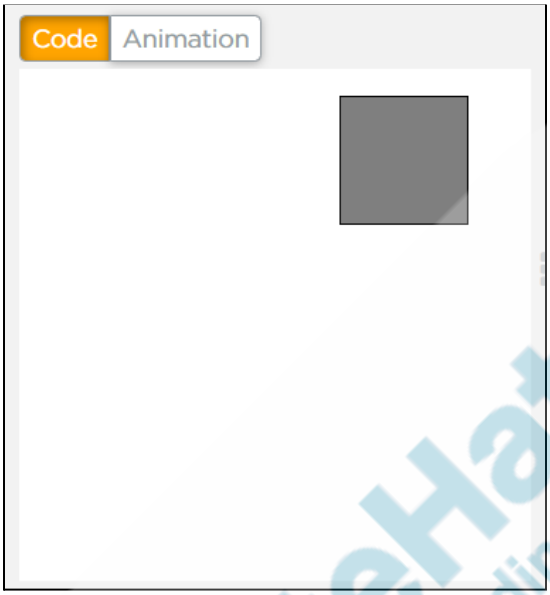
What happened? Let's increase the number and see what happens. Can you give me a higher number?	**ESR:** The rectangle shifted to the left. *The student gives a higher number for the first number.*

```
function draw() {
  rect(250, 250, 10, 100);
}
```

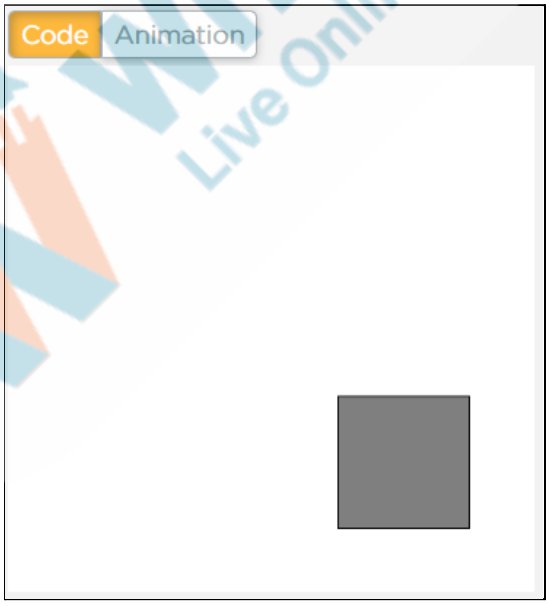


What happened?	**ESR:** The rectangle shifted to the right.
What do you think the 1st number stands for?	**ESR:** To position the rectangle left or right in the horizontal direction.
Yes! The first number stands to move the rectangle left or right, or we can say on the x-axis. *The teacher repeats the above steps for the second number.* What do you think the second number stands for?	
Change the 2nd number similar to the first number • *Reduce it first and run the program.* • *Increase it and run the program.* • *Ask the student to verify if their guess was correct.*	**ESR:** The second number can be used to move the rectangle up and down.

```
function draw() {
  rect(250, 20, 10, 100);
}
```



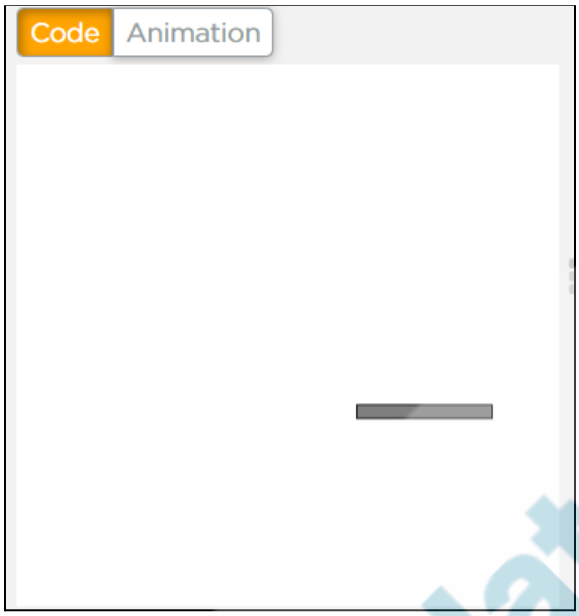

```
function draw() {
  rect(250, 250, 10, 100);
}
```



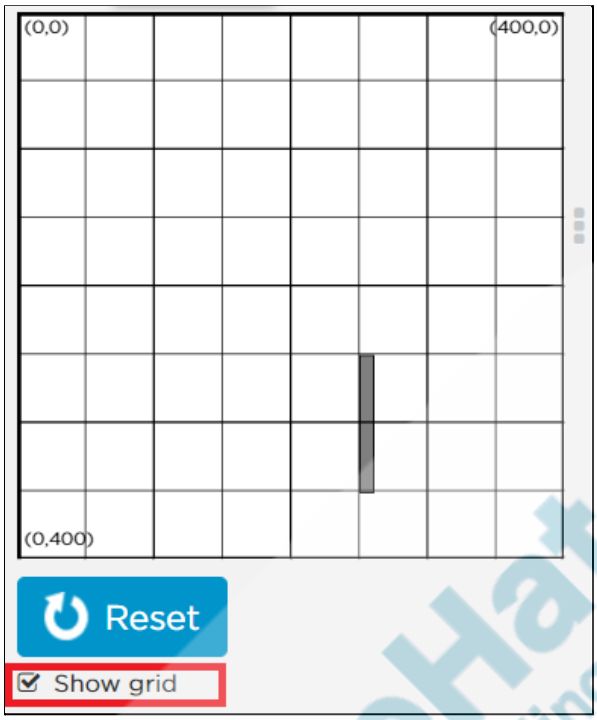
Let us experiment with the third and fourth numbers and see what happens to our rectangle. *The teacher repeats the above steps with the 3rd and 4th numbers.* ***Make sure to revert the 3rd number before changing the 4th number to show the difference.***	*The student gives inputs on what numbers she/he would like to test for the third and fourth numbers.*

```
function draw() {
  rect(250, 250, 10, 100);
}
```


```
function draw() {
  rect(250, 250, 100, 10);
}
```



What do you think the 3rd and 4th numbers stand for? Amazing! So, we now know how to draw a rectangle of different sizes and position them anywhere on the screen. Don't we? Understanding the x and y positions can be sometimes tricky when we are positioning objects on the screen. Let's understand how the x and y positions are represented on a computer screen. *The teacher clicks on the "**Show grid**" below the Code Output Section to display the x and y axes on the canvas.*	**ESR:** The 3rd and 4th numbers stand for the width and height of the rectangle. **ESR:** Yes.



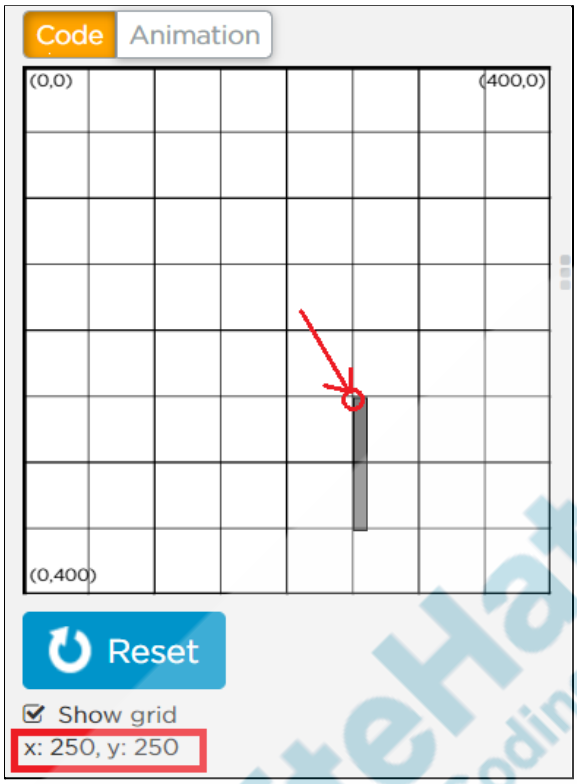
The (0,0) is the starting point of x and y positions, which is present in the top left corner of the output section.

The value of **x increases from left to right**.
The value of **y increases from top to bottom**(which is opposite from what we learn in mathematics).

The (400, 400) is the end point of x and y positions, which is the bottom right corner.

This output section on code.org is limited to a maximum value 400 for both x and y.

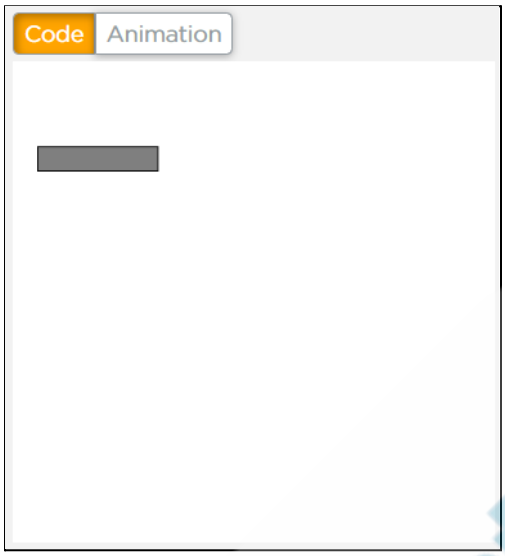
We can move our mouse on the output section to find out the values of x and y.



Do you think you can draw the 3 bricks, 1 ball and the paddle now on your own? Amazing! I am going to stop sharing my screen. You will start sharing your screen with me now, so that I can see what you are doing. While you will be drawing the objects, I can help and guide you. Does that sound good? Just make sure that you think loudly so that I can understand what's going on in your mind.	**ESR:** Yes. Let me try. **ESR:** Yes.

Let's start! *The teacher stops sharing the screen.*	*The student starts sharing her/his screen.*
Teacher Stops Screen Share	
STUDENT-LED ACTIVITY - 15 mins	
• **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 Full screen.**	
ACTIVITY • **Draw and position the bricks, player paddle and the ball.** • **Animate the player paddle so that it moves along with the mouse.**	
Teacher Action	**Student Action**
The student can go through the games given in the: *Student Activity 1 T-Rex Runner Game* *Student Activity 2 Pirate Invasion* *Student Activity 3 Multiplayer Car Racing Game* *Student Activity 4 Breakout Game*	
Guide the student to open the Student Activity 5. *Ask them to Sign in or Create a code.org account using their Google Account.* The one brick is drawn for you at the top.	

```
function draw() {
  rect(20, 70, 100, 20);
}
```



Now, you have to:

- Draw the 2 more rectangular bricks in the same line at top.
- Draw a player paddle at the bottom and move it with the mouse.
- Draw a square ball at the center.

Guide the student to draw and position 2 bricks at the top. Allow the student to experiment with the numbers.

Encourage the student to type the code.

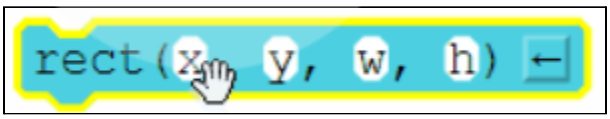
Let them adjust the width and height of the paddle first and then position them.

DO NOT GIVE THEM THE NUMBERS RIGHT AWAY.

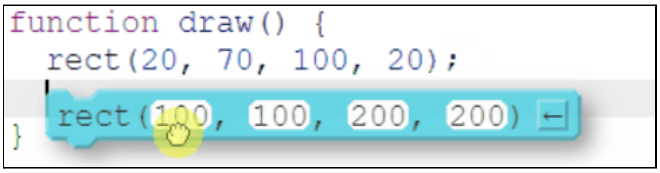
```
function draw() {
  rect(20, 70, 100, 20);
  rect(150, 70, 100, 20);
  rect(280, 70, 100, 20);
}
```

Encourage the student to type the code.
IF AND ONLY IF *the student is struggling a lot to type code in the text mode, guide the student to follow the steps below to work in block mode(Take the help from the link to understand the working steps):*

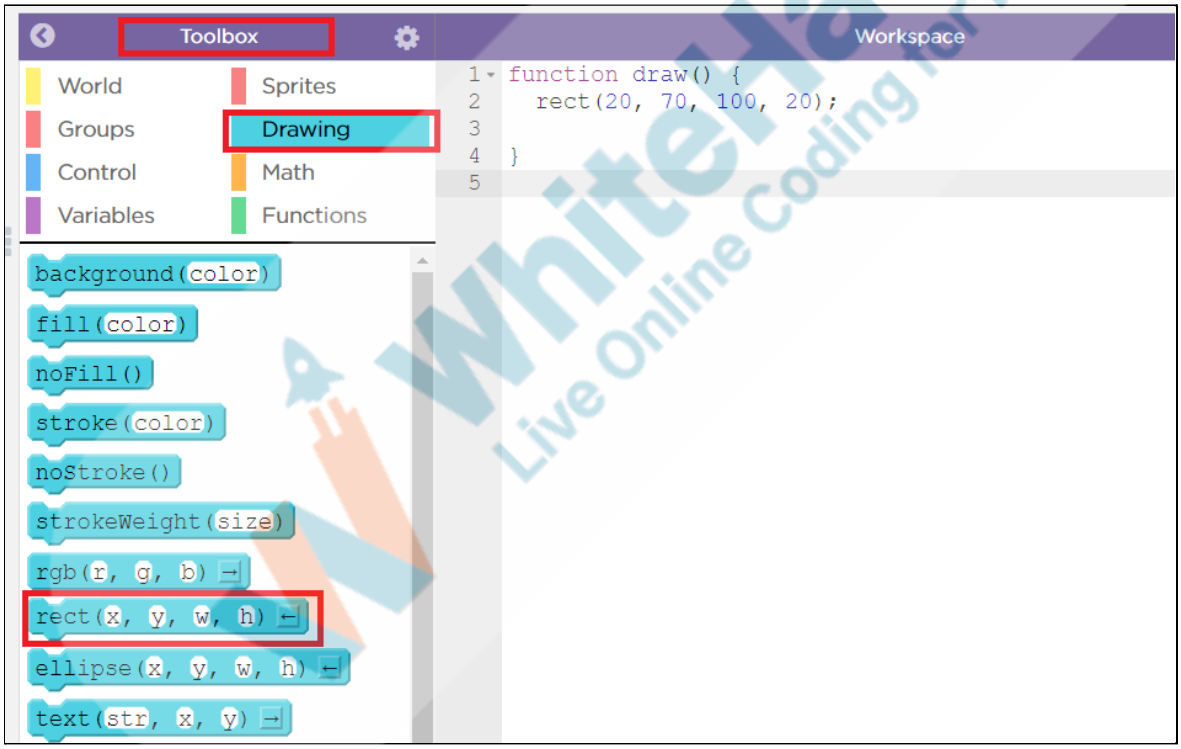
- Place the cursor at the appropriate line inside the **draw()** function first.
- Click on the “Drawing” in the toolbox.
- Scroll inside the Toolbox to find the **rect()** instruction. While scrolling, you will see an open hand symbol.



- Click and hold the mouse on the **rect()** inside the Toolbox to select the instruction. While holding the symbols turns into a fist(close hand).



- Drag and drop the **rect()** inside the **draw()** function without leaving the clicked state of the mouse before dropping.



Guide the student to draw and position the left computer-controlled player paddle. Allow the student to experiment with the numbers.

DO NOT GIVE THEM THE NUMBERS RIGHT AWAY.	

```
function draw() {
  rect(20, 70, 100, 20);
  rect(150, 70, 100, 20);
  rect(280, 70, 100, 20);
  rect(150, 370, 100, 10);
}
```

The player paddle is drawn at a fixed position in the horizontal direction right now. What is its position in that direction? *Guide the student to answer the x position of the player paddle.* Instead of drawing the paddle at a fixed position, we want to move it horizontally(left to right and right to left) whenever the mouse moves. Can you tell me which position of the player paddle, x or y, should be changed to move it continuously with the mouse in the horizontal direction?	**ESR:** It is 150. **ESR:** We should change x position.

To draw the player at different positions in the horizontal direction, there is something that stores the horizontal position (x position) of the mouse. We call it **mouseX**.

X stands for the x-axis(horizontal direction).
In place of **150** for the x position of the player paddle, let's put **mouseX**(or **World.mouseX**) there and see what happens.

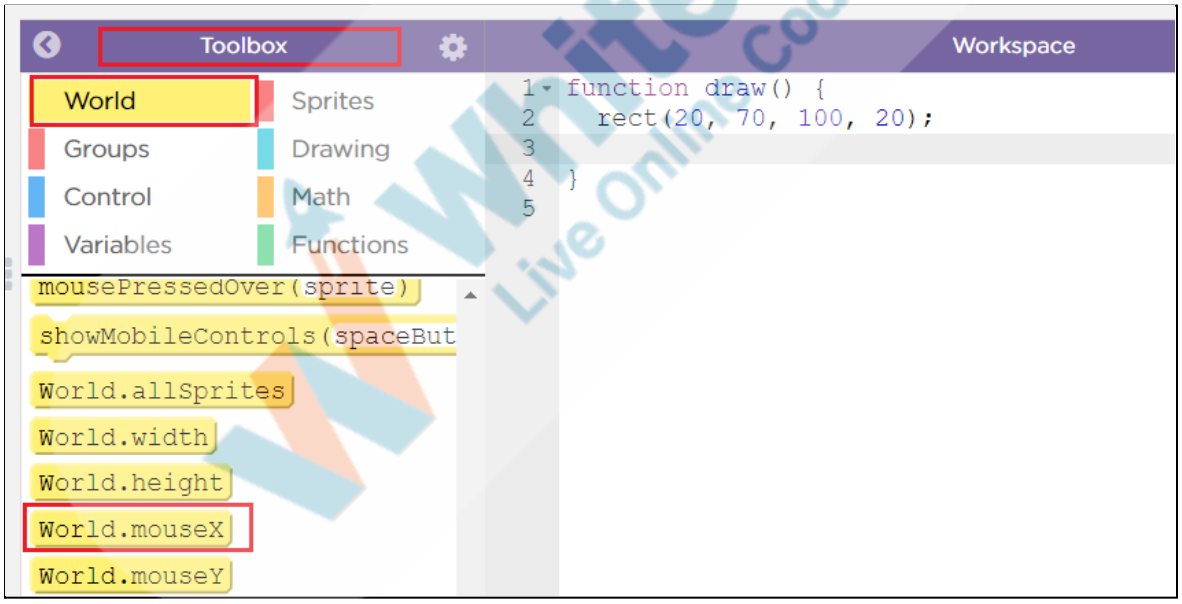
***Note**: **mouseX** is a property that can be accessed through the **World** object, which we will understand in upcoming classes.*

What do you think will happen if we run the code?

*The student replaces the **150** with **mouseX(or World.mouseX)**.*

ESR:The paddle will move with the mouse.

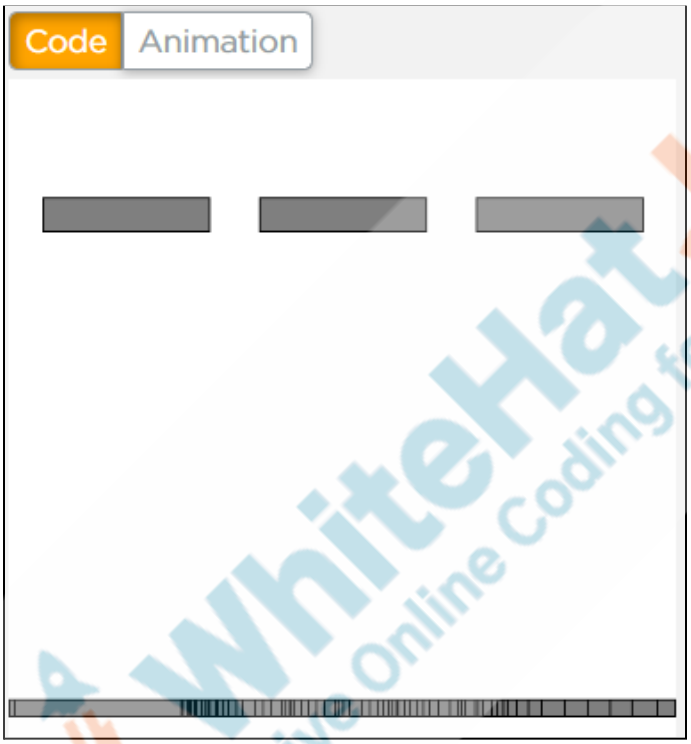
***Note**: The **mouseX** instruction can be found inside the "**World**".*



```
function draw() {
  rect(20, 70, 100, 20);
  rect(150, 70, 100, 20);
  rect(280, 70, 100, 20);
  rect(mouseX, 370, 100, 10);
}
```

OR

```
function draw() {
  rect(20, 70, 100, 20);
  rect(150, 70, 100, 20);
  rect(280, 70, 100, 20);
  rect(World.mouseX, 370, 100, 10);
}
```



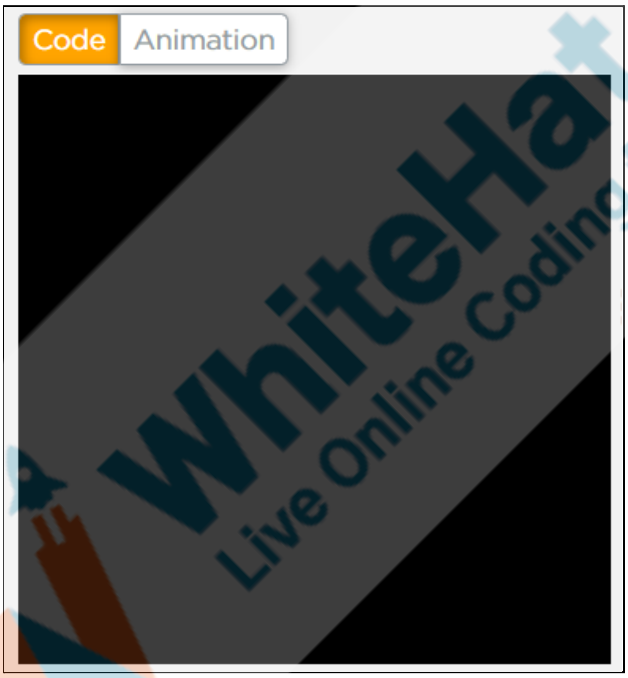
What's happening when you are moving your mouse? *Encourage the student to think and come up with a few possibilities.* Okay. Nice! But this is not what's happening. Keep trying! Let's try to figure it out by moving the mouse very slowly. What's happening when you move the mouse very slowly?	**ESR:** When we move the mouse, multiple lines are drawn. **ESR:** One rectangle moves and keeps drawing multiple lines.

Ask the student to move the paddle very slowly and observe how the rectangle moves when the mouse moves. Look more closely at the multiple lines, are these lines or multiple rectangles one over the other?	**ESR:** I think these are multiple rectangles.
Yes. Great! The paddle is moving with the mouse, but the old rectangles are still on the screen. Every time you move, a new rectangle is drawn at the new position of the mouse over the older rectangles. Why do you think this is happening?	**ESR:** Varied.
Encourage the student to think and come up with a few possibilities. Can you see that we have written every instruction inside a function, **draw()**?	**ESR:** Yes.
Every line of code inside the function **draw()** keeps running repeatedly by our program. That means all the 4 lines in the function **draw()** are running again and again. The 3 rectangles at top are fixed, but every time we move our mouse, a new rectangle is drawn for the player paddle. You can imagine the output section as a sheet of drawing, where you drew something. Now if you want to reuse the same drawing sheet again you will have to erase the old drawing or paint it with some color to make it clean, right?	**ESR:** Yes.
We will have to do the similar thing in our program too! So, how do we remove the older rectangles?	**ESR:** Varied.
We can paint the screen black before drawing anything on	

the screen. This way we will paint black over the old rectangles and then draw a new one on the top.

Guide the student to use background('black') instruction below the 4 rect() instructions and run the program.

```
function draw() {
  rect(20, 70, 100, 20);
  rect(150, 70, 100, 20);
  rect(280, 70, 100, 20);
  rect(World.mouseX, 370, 100, 10);
  background("black");
}
```



What happened? Can you try to think, why this would have happened? Let's see how we wrote the program. We have instructions in this order: 1. **rect()** 2. **rect()** 3. **rect()** 4. **background()** Here the program draws the 1st rectangle with the first **rect()** instruction and 2nd rectangle with the second **rect()** instruction and so on. After that, **background()** instructions turn the background to black at the end. And this is happening repeatedly. So we don't get to see any rectangles anytime, as these are painted by the black every time the computer draws them. Remember how we discussed at the beginning of the class the **order/sequences of instructions given to the computer matters**! So, where do you think we should write our **background()** instruction? Superb! *Guide the student to use background('black') instruction above the 4 rect() instructions and run the program.*	**ESR:** The whole output screen turned black! **ESR:** Varied. **ESR:** We should write it before all the **rect()** instructions.

```
function draw() {
  background("black");
  rect(20, 70, 100, 20);
  rect(150, 70, 100, 20);
  rect(280, 70, 100, 20);
  rect(World.mouseX, 370, 100, 10);
}
```

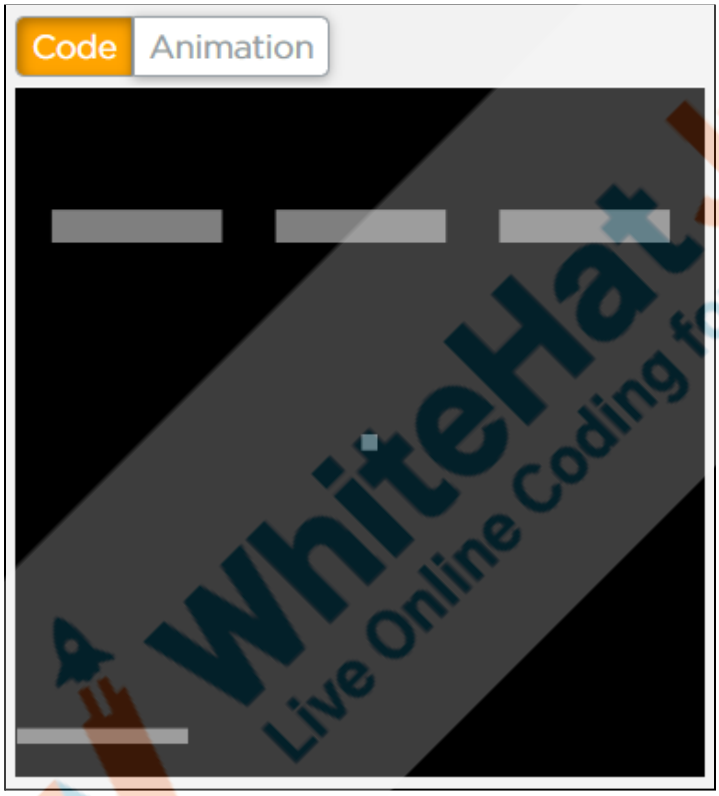
Complement the student.

Wow! Amazing!

The player paddle is now animated. It moves along with the mouse.

Now let's quickly add the ball at the center(**Optional, if there's extra time remaining**).

```
function draw() {
  background("black");
  rect(20, 70, 100, 20);
  rect(150, 70, 100, 20);
  rect(280, 70, 100, 20);
  rect(World.mouseX, 370, 100, 10);
  rect(200, 200, 20, 20);
}
```

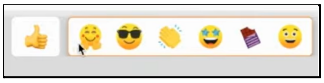


We have done quite a bit in one class!

Awesome <kid's name>! That is super cool!

Note: *If the student is a fast learner, ask the student to try out the activities given under the* ***ADDITIONAL ACTIVITIES*** *section below in the document.*

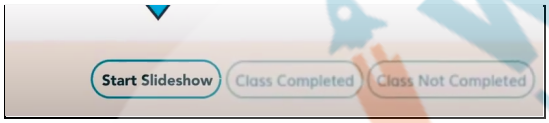
In the top left corner, press the  icon
It will open up a tray of emojis



Press sunglasses Emoji

WRAP-UP SESSION - 05 mins

Let's quickly review what we have learned in today's class. Can you quickly summarize all that you have learned today?		**ESR:** ● It is important to learn to code. ● Computers understand a different language, which can give exact instructions. ● We learned about **rect()** instruction to draw a rectangle ● We learned to move shapes with the mouse. ● We drew paddles and the ball.
You did excellent work today! We learned to start designing a game today. In the next class, we will learn more about coding the breakout game. Let's show this app to your parents. Please ask them if they can join us for some time?		***THE STUDENT INVITES THE PARENT.***
Student-Led Activity **Roblox Activity**	Can you click on Student Activity 8? Would you like to create games of this kind?	**Student Activity 8-BLAZING OBBY** *Allow the student to play for 5 mins max.* **ESR**: Yes

	We will create 3D games and publish them during the course. Are you excited about this?	**ESR**: Yes
Call the parents, introduce yourself, and celebrate the kid's accomplishment.		
Teacher- *"Hi, I feel proud to say that today, <child's name> coded his/her first game. <Child's name>, please show the game you created today."* • **Appreciate and compliment the student for trying and making strong progress to learn to code.** • **Get to know how they liked the session.** • **Review and check their understanding.** • **Challenge the student to make the ball move in the game on their own after the class. Do not share any code or help the student. Just keep encouraging the student with statements like: What do you think we can do to move the ball / make it bounce etc.?" ; "Seems like a good idea. Let's see what happens when you do this."**		
Let me give you a quick summary of what we will be learning in the course. Please click on the **Start Slideshow** button on the panel.  **Slide 1 Script: Your Kid's Journey** Over the next few slides I will be walking you through: • What you will learn. • What you will build. • Key Milestones you will accomplish along the Way. Our course follows a **project-based learning** philosophy where kids apply the concepts they learn to build apps and	**Curriculum Overview** ***INVITE THE PARENT TO SEE.***	

games throughout the course. Creating apps and games makes their learning practical instead of theoretical.

Slide 2 Script: Course Structure

Our course is structured into 3 modules:

1. Game Development
2. App Development
3. Space Tech + Artificial Intelligence

1. **Game Developer Certification**: In the 1st module of 8 classes, your kid will be building the Breakout Game by adding more elements. We will build on the pong game created in this trial class and in the process learn **core coding logic concepts** like Sequences, Functions, and Loops. This dramatically improves their logic and concentration. Let me show you the final version of this game.

Slide 3 Script: Game Development Project Build

- In the trial class, we started with designing and structuring the Breakout game. But in the course, the kid will create a much more interesting Breakout game version along with 1 person playing this game, the computer can automatically predict the movement of the ball and there will be a scoring system as well.

Slide 4 Script: Game Developer Certification

- At the end of this module, you will become a Certified Game Developer.

Slide 5 Script: Roblox

- Roblox is a fantastic 3D game environment which I personally enjoy making games in. Games created on Roblox are fun to play, and even more fun to create.

• After the 19th class checkpoint, a set of bonus classes will unlock for you. In these classes you will build and publish your first Roblox game. • You will get exclusive access to curated webinar sessions from gaming professionals. • ROBLOX helps develop skills like Game Design, Critical Thinking, Creative Ideation, and Storytelling. • You'll also get access to publish their games on the Roblox platform! **Slide 6 Script: EXCLUSIVE: Sonam Wangchuk's Class** • In the special **Sonam Wangchuk** class, students will learn about how solar energy is used as an eco-friendly source to light the remote corners of Ladakh. They also use the power of coding to demonstrate what goes into solar energy projects right from installing solar panels to using them for charging electric cars and bikes. **Slide 7 Script: App Development** 2. **App Development Certification**: In the 2nd module, your kid will be further strengthening their foundational coding concepts by learning Object-Oriented Programming, Databases, and Physics Engine libraries and applying the same to build advanced Gaming Apps. Let me show you the apps that they will be building. **Slide 8 Script: App Development Project Build** • As you can see, they will create games such as Pirate Invasion, Feed the Bunny, as well as a Multiplayer Car Racing App. **Slide 9 Script: App Developer certification** • Your kid will become among the **youngest in the world to be a certified Android and iOS App Developer in this module.**	

Slide 10 Script: Space Tech + Artificial Intelligence

3. **SpaceTech + Artificial Intelligence:** In the 3rd module, we explore data and how to use data to make complex decisions. Any problem can be described with the help of data. We will use python programming in this course to visualize the data, analyze them and make decisions with the help of data. We will also learn a few machine learning and AI algorithms to predict future events based on past data!

   Let me show you some of the Apps for you to get a better understanding.

Slide 11 Script: Space Tech + Artificial Intelligence Apps

- As seen here, your kid will build a Storytelling App using a database where stories can be added and this app will read the stories using Text to Speech API.

- Along with your kid will create an ISS tracker app that will track and show the location of the International Space Station.

Slide 12 Script: Space Tech + Artificial Intelligence Apps

- As promised, by the end of this module, your kid will be able to build full entrepreneur-ready Tech Apps.

Slide 13 Script: Space Explorer Program

- Your kid will be building an Earth rotation, Smart Energy App, Total and partial eclipse as well as Lunar Phase.

Slide14 Script: Space Bootcamp

- While the kid starts learning coding from basics. They will get exclusive access to space themed

webinars & hackathons in our Space Bootcamp program.
- Renowned **ex-NASA scientist Donald James** will be delivering webinars to share their space experiences and guide the kids for a better future followed by the Hackathon.

Slide 15 Script

- As promised, by the end of this module, your kid will be able to build full entrepreneur ready Tech Apps.

Slide 16 Script

- Like I said at the beginning we have a project-based learning philosophy and to ensure that you are able to learn and demonstrate the independent application of concepts by themselves we have quizzes and projects after every class, as well as Capstone Classes and Projects at the end of every sub-module.

Slide17 Script: Live Report Card

- In addition, we will be sending you a **Live Report Card** wherein we will be measuring your kid's progress on various parameters like **Coding Proficiency, Logic, Creativity,** and **Concentration**. I will be giving you personalized feedback based on the same. You can click on the screen to view a sample report card.

Slide18 Script: Individual vs Group Learning

- Our curriculum offers **Individual Learning** where you kid learns through individual practice with one instructor in the live class & **Group Learning** where you kid would learn with other kids with one instructor in the live class.

***Note**: Discuss other aspects like scheduling class, PTMs etc of the Individual and Group Learning with the parent using the slide.*

Slide19 Script: Whitehat Jr Superstars • Now let me show you the app created by one of our 6 year old students, Hirranya Rajani. Hirranya's brother was differently abled. Within 40 hours of learning to code with Whitehat, Hirranyaa made a sign language app for people with hearing disability which translates text written in the App into sign language. All handicapped kids in her community are now using the app. • This App has been featured widely in international media. Isn't it amazing that 8-year-olds are creating apps like these within just 40 hours of learning to code! **Slide 20 Script: Whitehat Jr YouTube Channel** • Thank you for your time today. You can watch our YouTube Channel(Whitehat Jr) to learn more about us.	
Teacher Guides Student to Stop Screen Share	
FEEDBACK • **Appreciate student's efforts in the class.** • **Ask the student to make notes for the reflection journal along with the code they wrote in today's class.**	
Teacher Action	**Student Action**
We are just getting started on this exciting journey of coding. I am eagerly looking to teach you more in the next class. Meanwhile, you can experiment with different colors in the **background()** instruction and try to move the ball.	*Make sure you have given at least 2 hats-off during the class for:* 

Remember, in programming, we can experiment and learn a lot of stuff." To provide you with more details about the course as well as answer any questions that you may have our academic counselor will get in touch with you. **If they are not able to reach out to you, do you have a preferred time?** ***<Note down the preferable time when parent will be reachable for further communication and update the same to the academic counsellor>*** My schedule is almost full, but I'd love to have your kid as my student, since your kid is exceptionally bright with true entrepreneurial entrepreneurship potential! Thank you for your time today. Kindly stay on the panel and do not close this page when I end the class--our entire curriculum along with the details will be displayed on the panel.	 
Teacher Clicks 	
ADDITIONAL ACTIVITIES	
Additional Activity 1: **Ask the student what if the background is turned to white.** Do you think we will get multiple rectangles or one rectangle moving with the mouse? *Wait for the response from the student and then let the student experiment.*	**ESR:** Varied.

We will get one rectangle moving with the mouse, as it **does not depend on the color we choose inside the background()** instruction.

```
function draw() {
  background("white");
  rect(20, 70, 100, 20);
  rect(150, 70, 100, 20);
  rect(280, 70, 100, 20);
  rect(World.mouseX, 370, 100, 10);

}
```

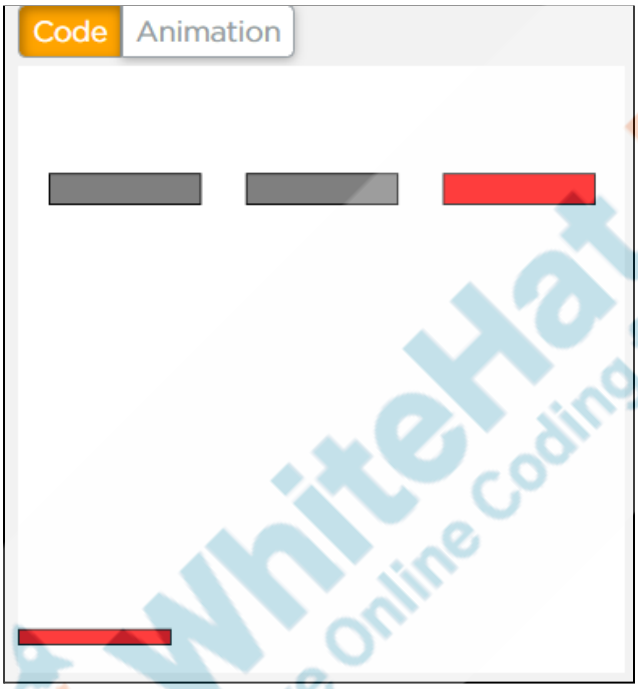
Additional Activity 2:

Guide the student to try fill("ColorName") instruction.
We can use the **fill()** instruction to fill colors inside the rectangles.

Note 1: The **fill()** instructions fill colors to every shape drawn below that.

```
function draw() {
  background("white");
  rect(20, 70, 100, 20);
  rect(150, 70, 100, 20);
  fill("red");
  rect(280, 70, 100, 20);
  rect(World.mouseX, 370, 100, 10);

}
```



Note 2: Use the **fill()** instruction above every **rect()** instructions if you want to fill the different colors to the rectangles.

```
function draw() {

  background("white");

  fill("red");
  rect(20, 70, 100, 20);

  fill("green");
  rect(150, 70, 100, 20);

  fill("red");
  rect(280, 70, 100, 20);

  fill("blue");
  rect(World.mouseX, 370, 100, 10);

}
```

ACTIVITY LINKS		
Activity Name	**Description**	**Link**
Teacher Activity 1	T-Rex Runner Game	https://editor.p5js.org/whitehatjr/present/ePJrHCACM
Teacher Activity 2	Pirate Invasion	https://whitehatjr.github.io/PiratesInvasion/
Teacher Activity 3	Multiplayer Car Racing Game	https://procodingclass.github.io/p5-multiplayer-car-race-game/
Teacher Activity 4	Breakout Game	https://editor.p5js.org/pro-curriculum-team/full/5bG1TiGII
Teacher Activity 5	Teacher Coding Activity	Teacher Coding Activity
Teacher Activity 6	Final Reference Code	Final Reference Code
Teacher Activity 7	Did You Know?	https://s3-cdnwhjr.whjr.online/18747c68-eb2e-471f-9f08-dfe88fa99b5f.html
Teacher Activity 8	Blazing-OBBY	https://www.roblox.com/games/7351712185/Blazing-OBBY
Teacher Activity 9	Instructions to Setup Roblox on Your Computer	https://s3-whjr-curriculum-uploads.whjr.online/9249e9fd-ac50-42ec-9465-01a58917ba22.pdf
Teacher Activity 10	Final Reference Code With Additional Activity	Final Reference Code With Additional Activity
Student Activity 1	T-Rex Runner Game	https://editor.p5js.org/whitehatjr/present/ePJrHCACM
Student Activity 2	Pirate Invasion	https://whitehatjr.github.io/PiratesInvasion/
Student Activity 3	Multiplayer Car Racing Game	https://procodingclass.github.io/p5-multiplayer-car-race-game/
Student Activity 4	Breakout Game	https://editor.p5js.org/pro-curriculum-team/full/5bG1TiGII
Student Activity 5	Student Coding Activity	Student Coding Activity
Student Activity 6	Blazing-OBBY	https://www.roblox.com/games/7351712185/Blazing-OBBY
Wrap up	Curriculum Overview	https://s3-whjr-curriculum-uploads.whjr.online/effb386f-e38b-48e4-b82f-8763ff2640ab.html
Wrap up	Curriculum Overview (With Teacher Cues)	https://s3-whjr-curriculum-uploads.whjr.online/454d4f20-b7f5-4095-b66c-2d10fe32ea6b.html

Glossary for Teacher Reference

Sr. no.	Abbreviations/ Technical terms	Explanation
1	AI (Artificial Intelligence)	It refers to giving human intelligence to machines, which enable them in learning and problem-solving.
2	ML (Machine learning)	Machine learning is an application of artificial intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed.
3	URL	A Uniform Resource Locator (URL). A URL incorporates the domain name, along with other detailed information, to create a complete address (or "web address") to direct a browser to a specific page online called a web page.
4	Domain name	The domain name is a component of a uniform resource locator (URL) used to access websites, for example: **URL**: http://www.example.net/index.html. **Top-level domain**: net. **Second-level domain**: example.
5	Algorithm	A process or set of rules to be followed in calculations or other problem-solving operations, especially by a computer. You can also consider algorithms as a step-by-step process to solve a problem
6	React Native	A programming language based on JS and HTML which is used to create mobile applications
7	Student's Report Card	It is used to keep a track of all the students' progress. Which segregates as Proficiency, Logic, Creativity, and Concentration
8	Capstone Classes	These are revision classes placed after some classes such as C8, which has the intention to revise all the concepts which have been taught in previous classes
9	Silicon Valley Program	This is a competition between the students at whitehat, which has the challenge to create an app or game which can enable them to solve a real-life problem. The best out of them are selected and given a tour of NASA (virtually) and get a chance to showcase their app/games to various entrepreneurs
10	Canvas	It can be considered as the output viewing box of the editor.

11	Space Tech	In these classes, we will be using data from space to predict space events like the collision of an asteroid with a planet
12	Physics Engine	A physics engine is computer software that provides an approximate simulation of certain physical systems, such as rigid body dynamics (including collision detection) and soft body dynamics.
13	Pirate Invasion Game	In this game, we learn the concept of a physics engine and use it to create a pirate invasion game where we have to defend our tower by shooting down the pirate boats.
14	Multiplayer Car Racing Game	We learn the concept of real-time database using this game and understand how to include the database in our games/apps and use this logic to create a multiplayer game
15	Stellar App	A React Native Space Tech app that gives updates on the constellations and planets visible in the sky on any given night at a given location.
16	Storytelling App	We use React Native to create this app, in this App users can add stories and the App will read the stories by converting Text to Speech. Users can also choose between a light and dark theme.
17	ISS Tracker App	This app is made using React Native, which tracks the position of the International Space Station and shows it on the map. It also gives a visual warning to earth of the upcoming meteors.
18	Movie Recommendation App	This app is built using React Native. It studies user's movie-watching patterns using machine learning algorithms and suggests awesome movies based on the user's data.
19	Exoplanet App	A statistical data-driven analysis on identifying habitable exoplanets that could potentially be our next home and creating a catalog app providing information about these planets and neighboring stars.
20	Feed the Bunny	In this game, we use the physics engine concepts to connect different bodies. The goal of the game is to break the rope, so that bunny can eat the fruit while collecting the maximum stars possible.
21	Breakout Game	We designed this gaming app in the first 7 classes which help us understand simple game movement and game dynamics and also various programming logic.
22	ESR	Expected Student Response.