Topic	Scope of variables	
Class Description	Students will learn about the global and local scope of variables. Students will write a reset function for the game to reset the game from within the game itself.	
Class	C16	
Class time	45 mins	
Goal	● Change the scope of some variables from local to global to be used anywhere in the code. ● Write a reset function to restart the game when the reset icon is pressed. ● Set up local environment to run Trex code on the local machine	
Resources Required	● Teacher Resources ○ p5 login ○ Laptop with internet connectivity ○ Earphones with mic ○ Notebook and pen ● Student Resources ○ p5 login ○ Laptop with internet connectivity ○ Earphones with mic ○ Notebook and pen	
Class structure	Warm Up Teacher-led Activity Student-led Activity Wrap up	5 mins 15 min 15 min 5 min

CONTEXT

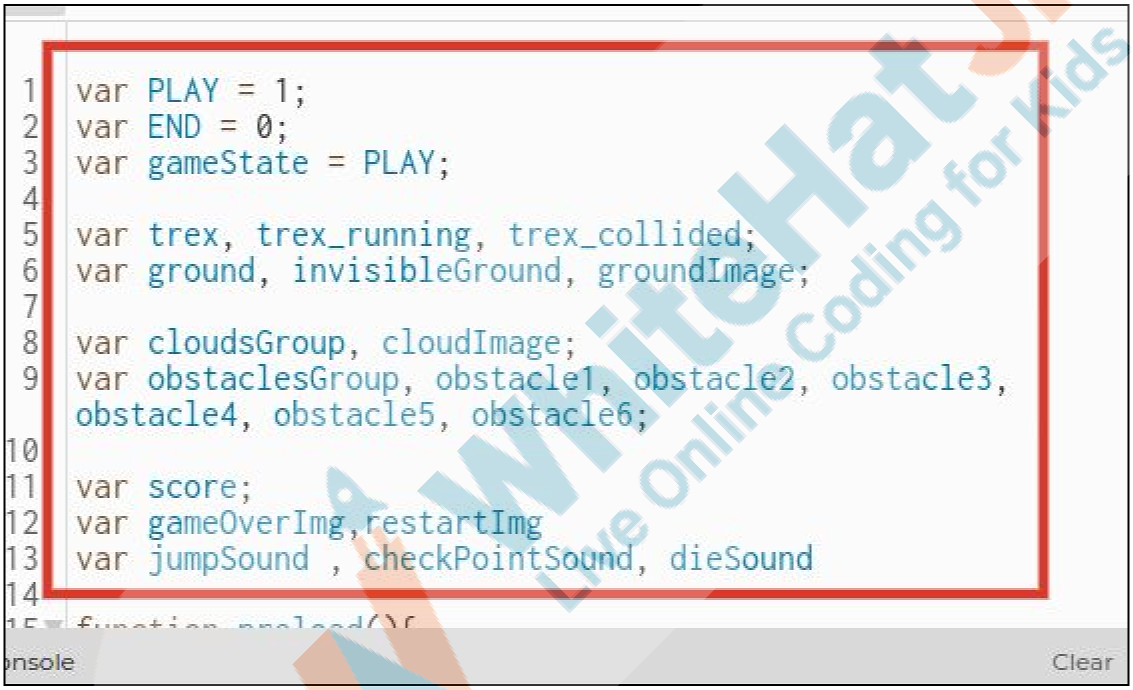
- **Set context for writing the reset function which restarts the game when the reset icon is pressed.**
- **Prep the student by setting up local development environment on their machine.**

Class Steps	Teacher Action	Student Action
Step 1: **Warm Up** **(5 mins)**	We had almost finished the Trex runner game, remember? Which game are we going to start building next?	**ESR:** Yes! **ESR:** Angry Birds!
	But before we start working on developing the game - Angry Birds - we will do two important things. 1. Currently, in our Trex game, if we want to reset the game after the game is over; we have to stop and play the code again. We will be writing code so that we can press on the reset icon in the game to reset the game. 2. Our code, right now, runs on the p5 editor. Ideally we would want our code to run anywhere on the browser so that we can upload it on a website and share it with friends who, then can play our game. We will be seeing how to do that. We'll also set up a local development environment on our machine so that we can code outside the p5 environment on our local machine! Is this exciting?	**ESR:** Yes!!
	Let's get started !!	
Teacher Initiates Screen Share		

CHALLENGE

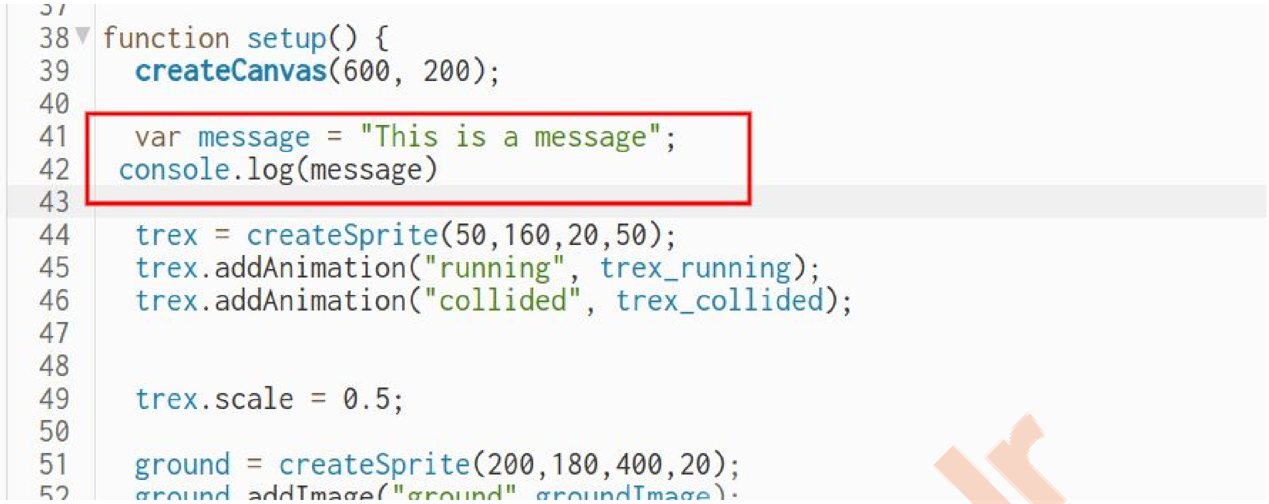
- **Change the scope of the variable to allow it to be accessed anywhere in the code.**

Step 2: **Teacher-led Activity** **(15 min)**	Before we start with today's class, we will learn about scope in programming? Have you ever wondered why we declare all the variables at the top of our program, rather than inside function setup() or function draw()?	*The student can come up with his/her responses.*



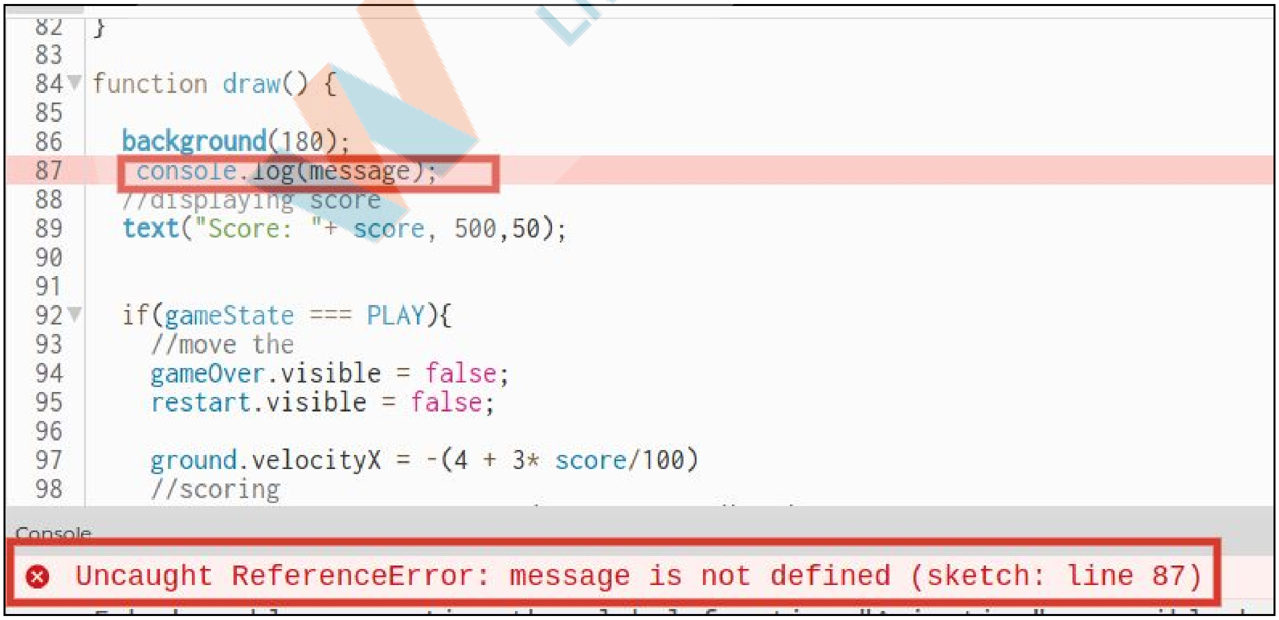
```
var PLAY = 1;
var END = 0;
var gameState = PLAY;

var trex, trex_running, trex_collided;
var ground, invisibleGround, groundImage;

var cloudsGroup, cloudImage;
var obstaclesGroup, obstacle1, obstacle2, obstacle3, obstacle4, obstacle5, obstacle6;

var score;
var gameOverImg,restartImg
var jumpSound , checkPointSound, dieSound

function preload(){
```

	Let's understand why. Let's try to declare a variable called message inside function **setup()** which stores a message. Let's print it inside the function **setup()** itself.	*The student helps the teacher in writing this code.*



```
function setup() {
  createCanvas(600, 200);

  var message = "This is a message";
console.log(message)

  trex = createSprite(50,160,20,50);
  trex.addAnimation("running", trex_running);
  trex.addAnimation("collided", trex_collided);


  trex.scale = 0.5;

  ground = createSprite(200,180,400,20);
  ground.addImage("ground",groundImage);
```

	Is the message printed on the console?	**ESR:** yes!
	How many times?	**ESR:** Once
	That's because - setup() is called only once.	
	What do you think will happen if we log the message inside **draw()**	**ESR:** The message will be printed many times!
	Let's see if that is what happens.	

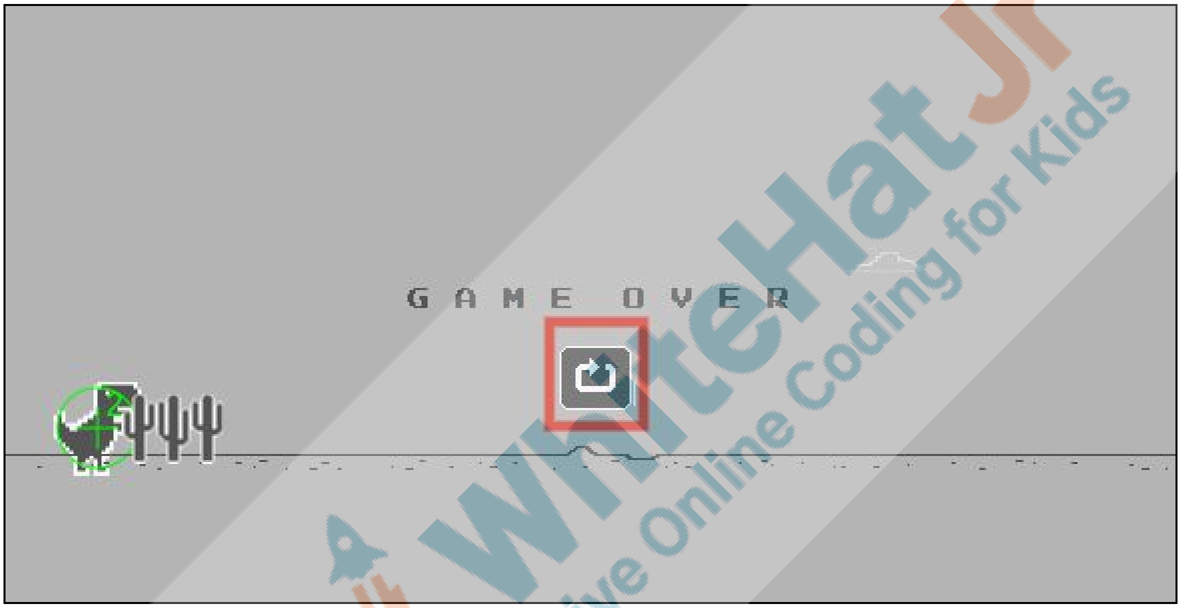


```
82  }
83
84  function draw() {
85
86    background(180);
87    console.log(message);
88    //displaying score
89    text("Score: "+ score, 500,50);
90
91
92    if(gameState === PLAY){
93      //move the
94      gameOver.visible = false;
95      restart.visible = false;
96
97      ground.velocityX = -(4 + 3* score/100)
98      //scoring
```

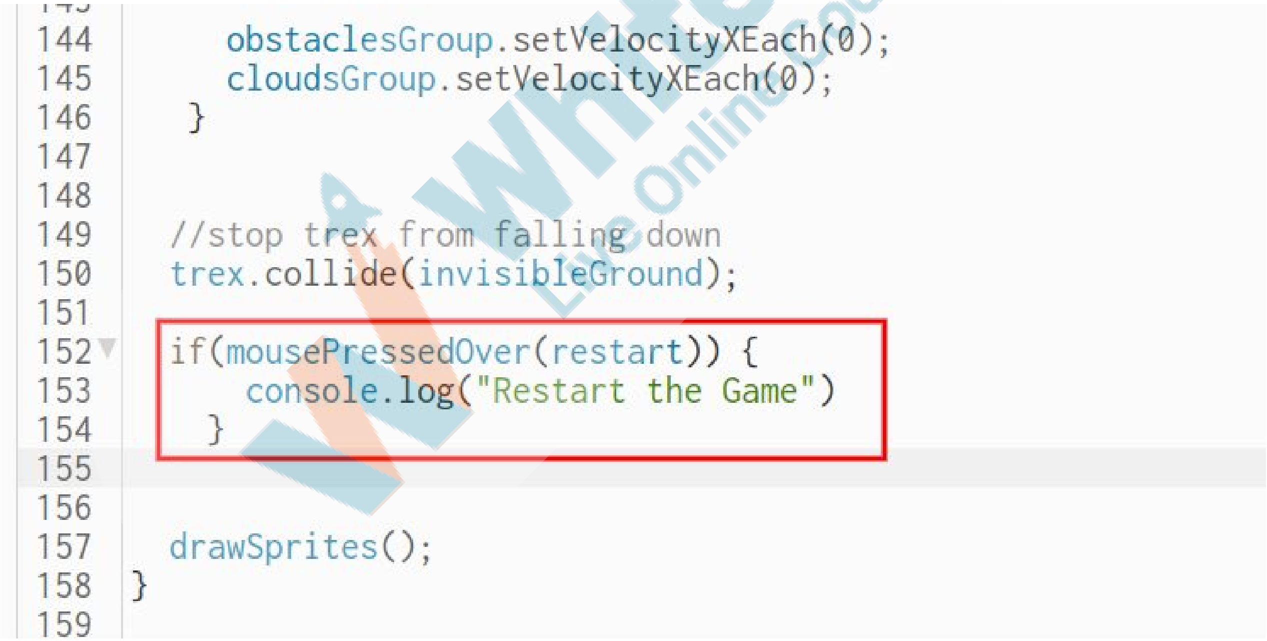
Console

```
Uncaught ReferenceError: message is not defined (sketch: line 87)
```

	What do you see? But we have defined the “message”, haven’t we - we just printed it on the console!	It throws an error. “message is not defined” *Allow the student to think about this and why it might be happening.*
	Variables in the program have a scope. They live and die within the scope. A variable declared inside a function or any block of code starting and ending with { } (for example - for loop) have scope defined only inside the function (or curly brackets). It cannot be accessed by any other function or any other block of code directly. Such variables are said to have local scope. Variables declared outside the functions at the top are said to have global scope. This means you can access them anywhere in the code. To learn more about scoping of variables in javascript, you can refer to the student reference link **Student Activity 2.**	*The student asks questions about scoping of variables.*
	Let us get started with the task for today. Teacher opens **Teacher Activity 1** and plays the game from the last class. Look at the game we designed in our last class. We see the Game over and restart icons when the game ends.	*Student observes and tries to understand the goal of the lesson.*

<table>
<tr><td></td><td>Currently, however the restart icon doesn’t work. We have to manually stop the code and play it again for us to re-play the game.
The challenge for us is to simply reset the game by pressing the reset icon inside the game. Sounds simple, right?
Let's see what challenges we will face in doing this.</td><td></td></tr>
<tr><td colspan="3"> </td></tr>
<tr><td></td><td>What we need is to tell the computer is that if someone clicks on the reset icon in the game, reset the game to start from the beginning, right?
How can we say that in the code?</td><td><b>ESR:</b>
Varied response.</td></tr>
<tr><td></td><td>Let me try to write this down:
We must find an instruction which detects the mouse pressed over a sprite (since reset icon is a sprite).
Let us look into the World tab in our toolbox.</td><td><i>Student detects the mousePressedOver(sprite) instruction.</i></td></tr>
</table>

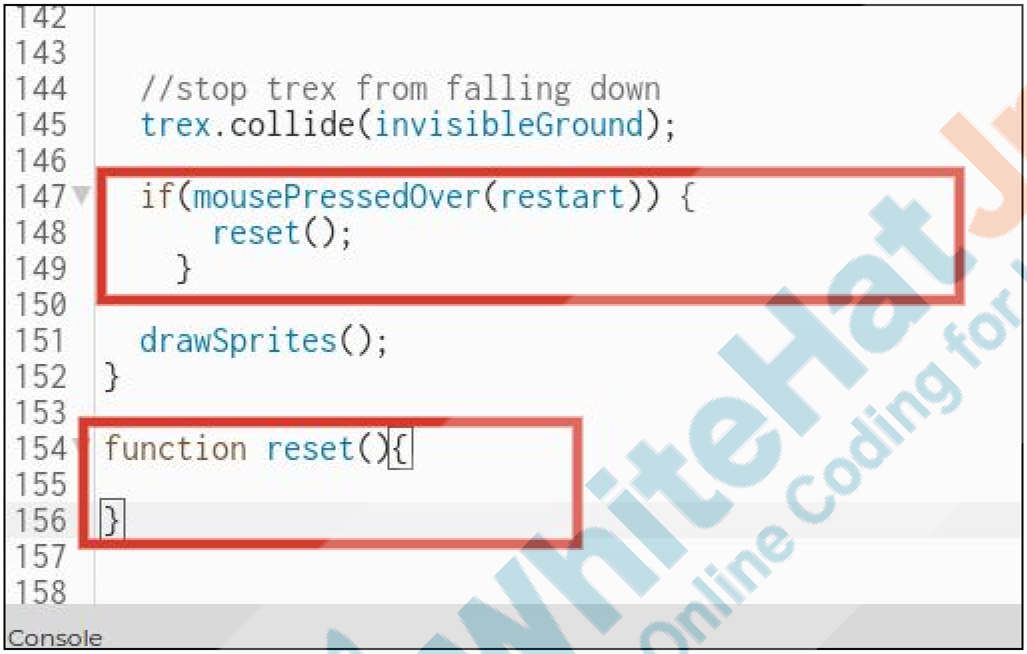
	Let us tell the computer that if the player presses the mouse over the reset sprite, reset the game. *Teacher writes the code for this.* There is an instruction in the p5.play library called mousePressedOver() which detects a mouse pressed over any sprite. It accepts the sprite as an argument and returns true if the mouse is pressed over the sprite and false otherwise. Let us try to use mousePressedOver and console log something on our screen when the mouse is pressed over the restart sprite.	*Student observes and asks questions.*



```
     obstaclesGroup.setVelocityXEach(0);
     cloudsGroup.setVelocityXEach(0);
   }


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

  if(mousePressedOver(restart)) {
      console.log("Restart the Game")
    }


  drawSprites();
}
```

	Let's run the code and check the output. *Teacher presses the reset icon when the game is over.*	*Student observes the "Restart Game" message printed on the console whenever Game Over is pressed.*

	Instead of printing something on the console, we can also call a reset() function which resets everything in the game to its original state. For now we can write an empty reset() function which we will call in our code.	*Student observes*



```
  //stop trex from falling down
  trex.collide(invisibleGround);

  if(mousePressedOver(restart)) {
      reset();
    }

  drawSprites();
}

function reset(){

}
```

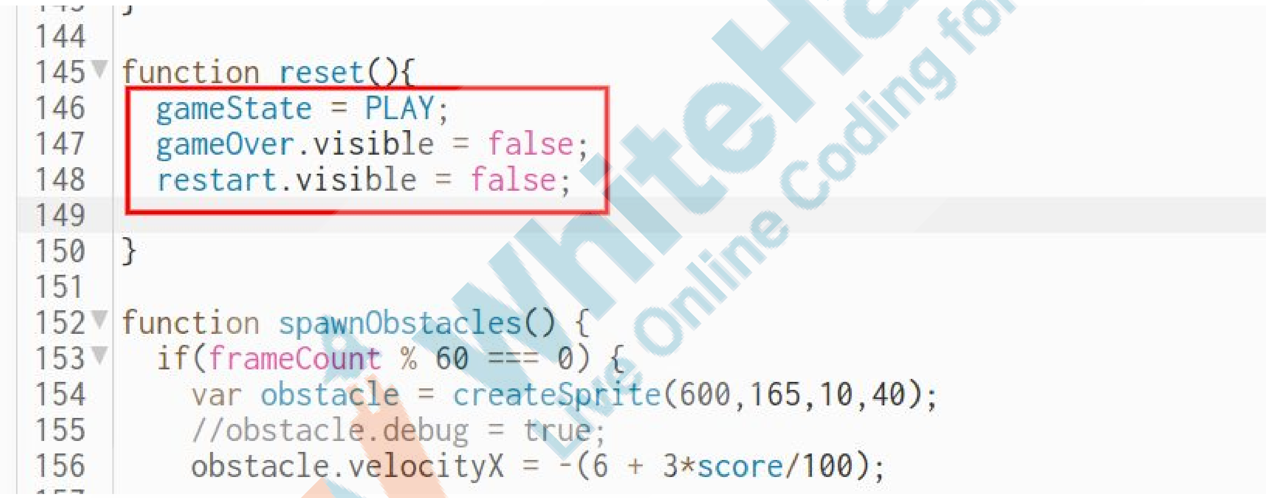
	Now, we need to write code inside the reset function which will reset the game back to its original state. Can you do that on your own?	**ESR:** Yes I can try.
Teacher Stops Screen Share		
	Now it's your turn. Please share your screen with me.	

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

ACTIVITY

- **Write a reset function to reset the game when the reset icon is pressed.**

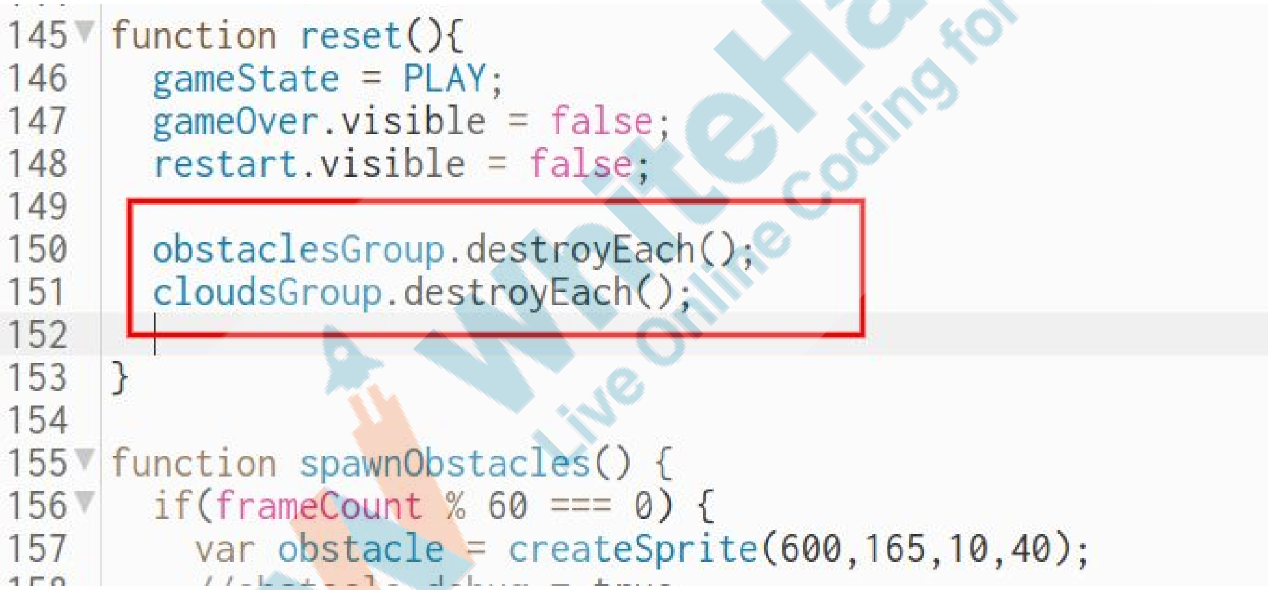
Step 3: **Student-Led Activity** **(15 min)**	What are the different things that we need to reset the game?	**ESR:** - Change the gameSate back to PLAY. - Make the gameOver and restart invisible again.
	Let us do these and see what happens. *Guide the student to change the gameState to PLAY and make the gameOver icons invisible.*	*Student opens Student Activity 1.* *Student writes code to make the changes and then runs to see what happens.* *The student presses the restart icon inside the game to see the output.*



```
}

function reset(){
  gameState = PLAY;
  gameOver.visible = false;
  restart.visible = false;

}

function spawnObstacles() {
  if(frameCount % 60 === 0) {
    var obstacle = createSprite(600,165,10,40);
    //obstacle.debug = true;
    obstacle.velocityX = -(6 + 3*score/100);
```

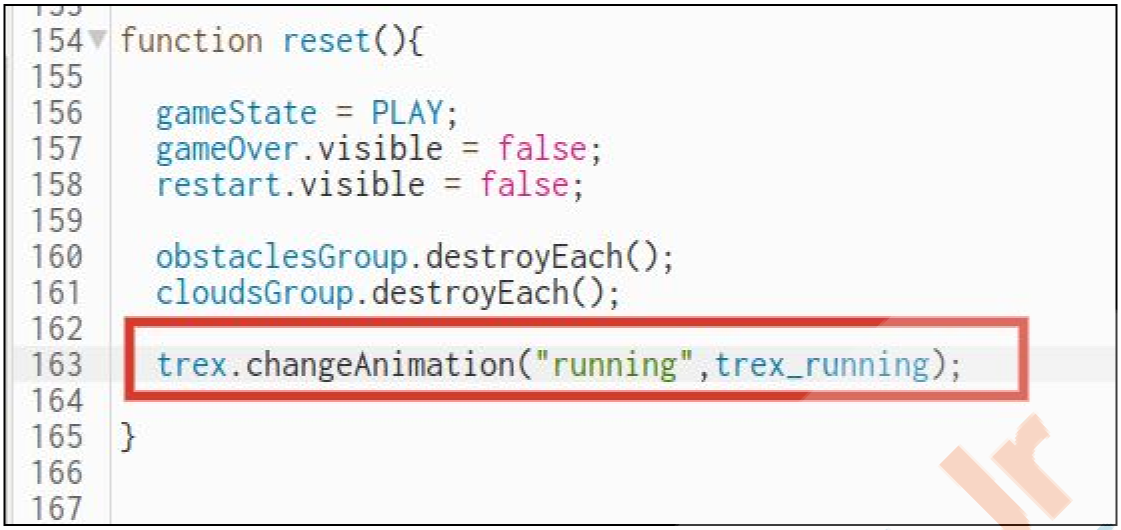
	What is happening? Why do you think this is happening?	**ESR:** The ground scrolls for a while and again gameState changes back to END. **ESR:** varied

	Remember, in the last class - we had changed the lifeTime of the obstacles so that they never disappear. As long as the obstacles are there, trex is going to collide with the obstacle immediately and the gameState will change back to END. Let us destroy all the obstacles and clouds when we reset the game. *Guide the student to write the code to destroy all the obstacles and clouds in the game.* ***Note:*** *Review group sprites and how their properties work with the student.*	*The student writes code to destroy the obstacles and clouds using the group properties.* *Student runs the code to see the output.*



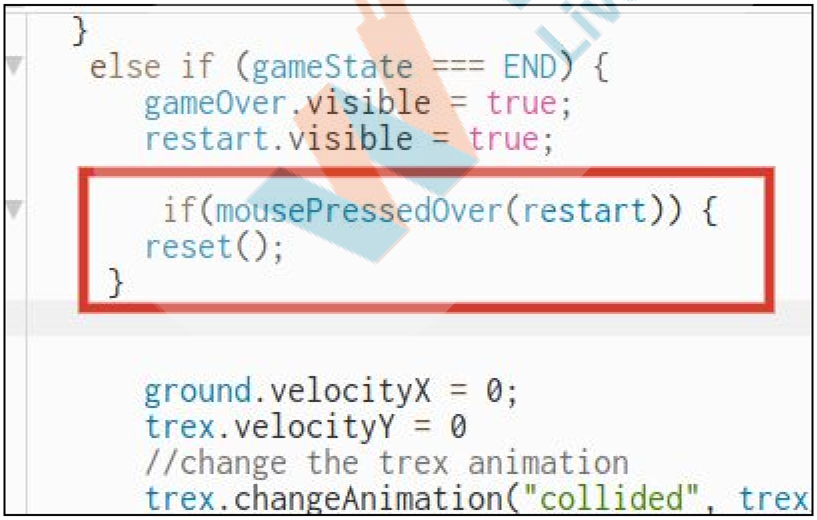
```
function reset(){
  gameState = PLAY;
  gameOver.visible = false;
  restart.visible = false;

  obstaclesGroup.destroyEach();
  cloudsGroup.destroyEach();
  
}

function spawnObstacles() {
  if(frameCount % 60 === 0) {
    var obstacle = createSprite(600,165,10,40);
```

	What's wrong now? So let's change it to running animation at reset.	**ESR:** The rex animation is still set to 'collided'. *Student writes code to change the trex animation.* *Student runs the code to see the output.*



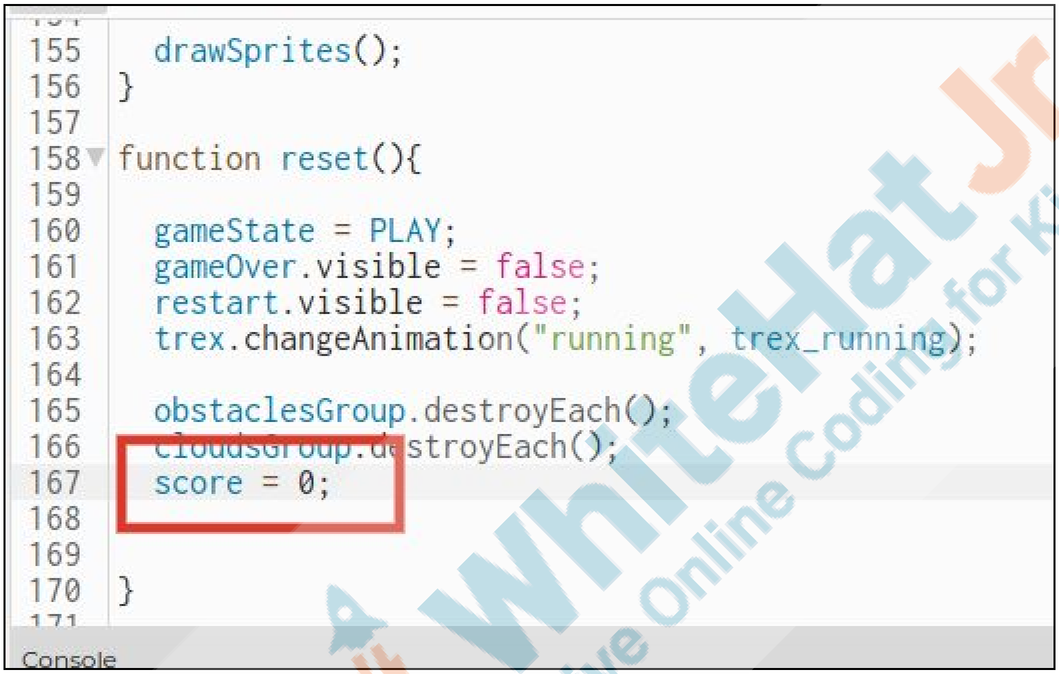
```
function reset(){

  gameState = PLAY;
  gameOver.visible = false;
  restart.visible = false;

  obstaclesGroup.destroyEach();
  cloudsGroup.destroyEach();

  trex.changeAnimation("running",trex_running);

}
```

	Can you notice that even though the “restart icon” is invisible during PLAY state in the game, you can still press in the area to restart the game anytime? What can we do to fix this?	*Student tries to press on the invisible “restart” icon in the game.* **ESR:** We can move the condition where we check the mouse pressed over reset icon inside Game End state only.
	Let’s do this and see if that works.	*Student writes code, runs and tests it.*



```
  }
  else if (gameState === END) {
    gameOver.visible = true;
    restart.visible = true;

     if(mousePressedOver(restart)) {
    reset();
   }


    ground.velocityX = 0;
    trex.velocityY = 0
    //change the trex animation
    trex.changeAnimation("collided", trex
```

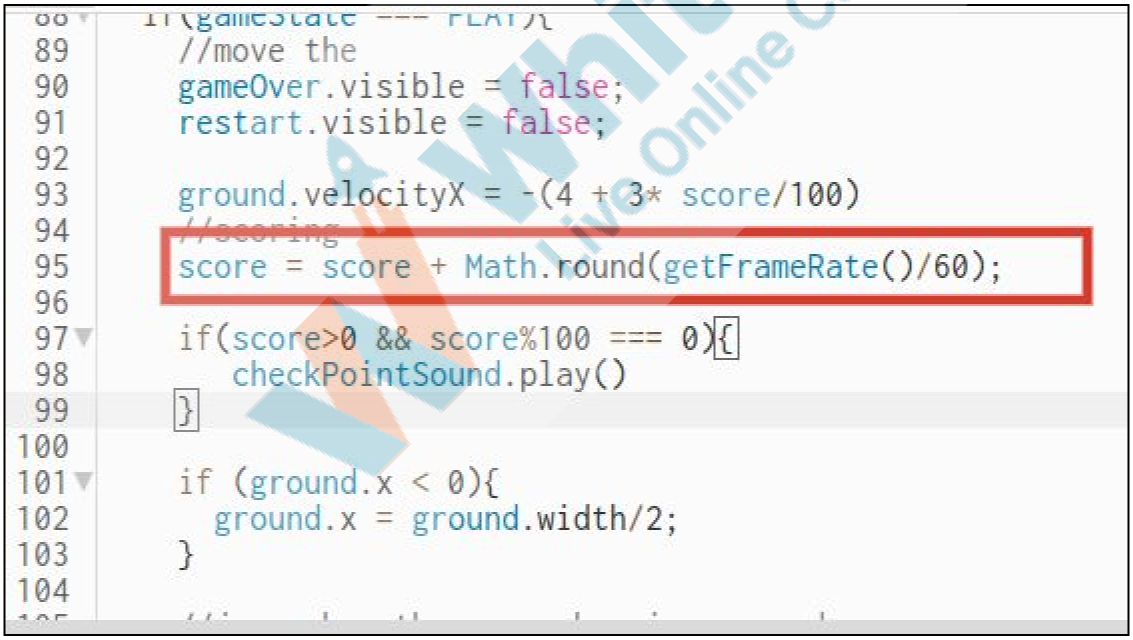
	Everything seems to be working well. But there is still an issue. Can you notice it?	**ESR:** The score does not get reset.
	What do you think we can do to reset the score in the game? Let's do it and see what happens.	**ESR:** Set score = 0 in the reset. *Student writes code and checks the output.*



```
  drawSprites();
}

function reset(){

  gameState = PLAY;
  gameOver.visible = false;
  restart.visible = false;
  trex.changeAnimation("running", trex_running);

  obstaclesGroup.destroyEach();
  cloudsGroup.destroyEach();
  score = 0;


}
```

	What happened? Can you guess why this is happening? In our game, our count value depends on the frameCount. The frameCount goes on increasing even when the game is in END state. Can we use something else to update the count so that it resets and starts from 0 when the game is reset.	**ESR:** The score temporarily resets to 0 but again restarts from its old value. **ESR:** varied **ESR:** varied

	Let us try to update count using frameRate. frameRate is the number of frames, frameRate is nearly constant throughout the game and nearly equal to 60. **Note:** It might be different on different systems. For some it might be 30. We could write : score = score + Math.round(frameRate/60) This will increase the count by 1 every frame. When the game resets, count becomes 0 and then the count starts from 0 again. Let us change the code and see if it works this time.	*Student makes the change in the code and runs to see the output.*



```
if(gameState === PLAY){
  //move the
  gameOver.visible = false;
  restart.visible = false;

  ground.velocityX = -(4 + 3* score/100)
  //scoring
  score = score + Math.round(getFrameRate()/60);

  if(score>0 && score%100 === 0){
     checkPointSound.play()
  }

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

<table>
<tr><td></td><td>All working now! Great job! We did it. There might still be some bugs in the game. You can find and work on it as your Bug Hunter Bounty Project.

For now, take pride in your work.</td><td>-</td></tr>
<tr><td colspan="3"><b>Teacher Guides Student to Stop Screen Share</b></td></tr>
<tr><td colspan="3"><b><u>FEEDBACK</u></b>
● <b>Encourage the student to make reflection notes in markdown format.</b>
● <b>Complement the student for her/his effort in the class.</b>
● <b>Review the content of the lesson.</b></td></tr>
<tr><td><b>Step 4:
Wrap-Up
(5 min)</b></td><td>Let us quickly review what we learned today.

What did we learn about scope today?</td><td><b>ESR:</b>
We learned about the scope of variables - global scope and local scope.
- Global scope variables can be used anywhere within the code.
- Local scope variables can be used within the block of code between { } in which they are created.</td></tr>
<tr><td></td><td>You get Hats Off for your excellent work!

In the next class, we will learn how to write code on our local machine!
We are going to have more fun in our next class when we will learn more about how to write code on our local machine. Later, we will also modify our Trex code slightly so that our</td><td><i>Make sure you have given at least 2 Hats Off during the class for:</i>
Creatively Solved Activities +10
Great Question +10
Strong Concentration +10</td></tr>
</table>

	game can be run on a mobile phone. Looking forward to our next class.	
Project Overview	**MONKEY GO HAPPY - 1** **Goal of the Project:** In Class 16, you learned about the scope of local and global variables. You also learned to write the reset() function for the t-rex game and rewrote the pong game in the p5 editor. In this project, you have to practice and apply what you have learned in the class and create a game where a monkey is searching for food. Add obstacles and a scoring system to make the game interesting. **Story:** A monkey has escaped from the zoo and is very hungry. Help the monkey collect Bananas by jumping over obstacles. I am very excited to see your project solution and I know you will do really well. Bye Bye!	*Students engage with the teacher over the project.*
	Teacher Clicks ✖ End Class	
Additional Activities	*Encourage the student to write reflection notes in their reflection journal using markdown.* Use these as guiding questions:	*Student uses the markdown editor to write her/his reflection as a reflection journal.*

	● What happened today? - Describe what happened - Code I wrote ● How did I feel after the class? ● What have I learned about programming and developing games? ● What aspects of the class helped me? What did I find difficult?	

Activity	Activity Name	Links
Teacher Activity 1	Trex Stage 7	https://editor.p5js.org/whitehatjr/sketches/BEIUXMHyM
Teacher Activity 2	Trex Stage 8 (Reference)	https://editor.p5js.org/whitehatjr/sketches/SwQvxw0ey
Student Activity 1	Trex Stage 7.5	https://editor.p5js.org/whitehatjr/sketches/CgIIV0sb0
Student Activity 2	Scoping Reference	https://www.w3schools.com/js/js_scope.asp