



Topic	Slingshot	
Class Description	Students create a slingshot effect in their Angry Birds game. Students also learn to employ the mouse events like mouseDragged and mouseReleased.	
Class	C28	
Class time	45 mins	
Goal	<ul style="list-style-type: none"> • Create a slingshot effect in the Angry Birds game. • Learn to use the mouse trigger events like mouseDragged and mouseReleased. 	
Resources Required	<ul style="list-style-type: none"> • Teacher Resources <ul style="list-style-type: none"> ○ Laptop with internet connectivity ○ Earphones with mic ○ Notebook and pen • Student Resources <ul style="list-style-type: none"> ○ 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 mins 5 min
CONTEXT <ul style="list-style-type: none"> • Review the concept of constrained bodies studied in the previous class. 		
Class Steps	Teacher Action	Student Action
Step 1: Warm Up (5 mins)	<p>Quickly let's review what we learned in the last class.</p> <p>Do you remember what we did in the last class?</p>	ESR: Yes, we learned how to create two constrained bodies.

	Let's also review the code from the last class.	<i>The student reviews the code from the last class.</i>
	<p>Before that I have an exciting quiz question for you! Are you ready to answer this question?</p> <p>Teacher click on the  button on the bottom right corner of your screen to start the In-Class Quiz.</p> <p>A quiz will be visible to both you and the student.</p> <p>Encourage the student to answer the quiz question.</p> <p>The student may choose the wrong option, help the student to think correctly about the question and then answer again.</p> <p>After the student selects the correct option, the  button will start appearing on your screen.</p> <p>Click the End quiz to close the quiz pop-up and continue the class.</p>	<p>ESR: Yes</p>
	<p>Last class, we had a log attached to the bird. So there were two bodies which were moving together.</p> <p>But in our game of Angry Birds, we want to launch the Angry Birds from a fixed Point.</p>	-

	<p>In this class, we will learn how to do that.</p> <p>Let's get started.</p>	
Teacher Initiates Screen Share		
<p style="text-align: center;"><u>CHALLENGE</u></p> <ul style="list-style-type: none"> • Read the documentation for <code>mouseDragged</code>, <code>mouseReleased</code> and <code>Matter.constraint</code>. • Replace <code>bodyB</code> with <code>pointB</code> to define the origin of projection for the bird. 		
Step 2: Teacher-led Activity (15 min)	<p><i>Teacher helps the student clone the code from the Student Activity 1.</i></p> <p><i>Teacher opens the link and starts the live collaboration.</i></p>	<p><i>Student clones the project from Student Activity 1 and opens it in VS Code.</i></p> <p><i>Enables the live share and shares the link with the teacher.</i></p>
<pre>\$ git clone https://github.com/whitehatjr/AngryBirdsStage3</pre>		
	<p>We had created the Chain class last time around. The chain object created using this class was used to tie the bird and the log in a constraint.</p> <p>Now, we will be working to create Slingshot.</p> <p>Let's modify this class and make it a SlingShot class.</p> <p><i>Teacher modifies the Chain in the Chain.js file to Slingshot.</i></p> <p><i>Teacher renames the file to Slingshot.js.</i></p>	<p><i>The student observes and learns.</i></p>

Teacher modifies the index.html file to include 'Slingshot.js'.

```

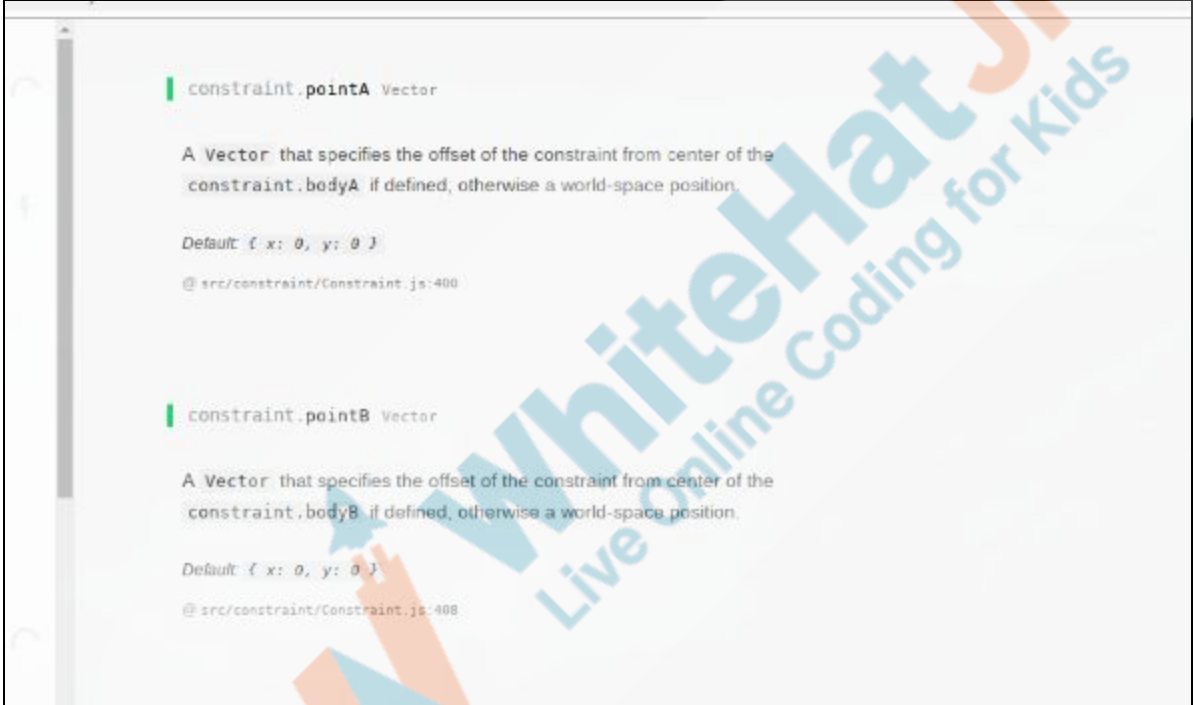
JS Slingshot.js x
JS Slingshot.js > Slingshot
1 class Slingshot{
2   constructor(bodyA, bodyB){
3     var options = {
4       bodyA: bodyA,
5       bodyB: bodyB,
6       stiffness: 0.04,
7       length: 10
8     }
9     this.sling = Constraint.create(options);
10    World.add(world, this.sling);
11  }
12
13  display(){
14    var pointA = this.sling.bodyA.position;
15    var pointB = this.sling.bodyB.position;
16    strokeWeight(4);
17    line(pointA.x, pointA.y, pointB.x, pointB.y);
18  }
19
20 }
  
```

```

index.html > html > head > script
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <script src="p5.min.js"></script>
5   <script src="p5.dom.min.js"></script>
6   <script src="p5.sound.min.js"></script>
7   <script src="matter.js"></script>
8   <script src="BaseClass.js"></script>
9   <script src="Ground.js"></script>
10  <script src="Box.js"></script>
11  <script src="Pig.js"></script>
12  <script src="Log.js"></script>
13  <script src="Bird.js"></script>
14  <script src="Slingshot.js"></script>
15  <link rel="stylesheet" href="js Slingshot.js" style="display: none;">
16  <meta charset="utf-8">
17 </head>
18 <body>
19   <script src="sketch.js"></script>
20 </body>
21 </html>
22
  
```

We are using two bodies - bodyA and bodyB.

We don't want the bird body to be

	<p>attached to any other body. Rather, we want it to be attached to and projected from a point. Let's see if we have that option in the Matter.Constraint.</p> <p>Let's look at the documentation.</p>	<p><i>The student looks through the documentation for Matter.constraint.</i></p>
		
	<p>It seems we can just specify pointB instead of bodyB and the body will be constrained from that point. Let's experiment and see.</p> <p>Modify the Slingshot class to include pointB everywhere in the code.</p> <p>Also change the script.js file to change Chain to Slingshot.</p>	<p><i>The student observes and learns.</i></p>

	<p>Pass the coordinates of a point as one of the parameters to the Slingshot() constructor.</p> <p>The coordinates can be anywhere you want the slingshot to appear.</p>	
--	--	--



```

1  class SlingShot{
2      constructor(bodyA, pointB){
3          var options = {
4              bodyA: bodyA,
5              pointB: pointB,
6              stiffness: 0.04,
7              length: 10
8          }
9          this.pointB = pointB
10         this.sling = Constraint.create(options);
11         World.add(world, this.sling);
12     }
13
14     display(){
15         var pointA = this.sling.bodyA.position;
16         var pointB = this.pointB;
17         strokeWeight(4);
18         line(pointA.x, pointA.y, pointB.x, pointB.y);
19     }
20
21 }

```

```

38
39     bird = new Bird(100,100);
40
41     //log6 = new Log(230,180,80, P1/2);
42     slingshot = new SlingShot(bird.body,{x:200, y:100});
43 }
44
45 function draw(){
46     background(backgroundImg);
47     Engine.update(engine);
48     strokeWeight(4);
49     box1.display();
50     box2.display();
51     ground.display();
52     pig1.display();
53     log1.display();
54
55     box3.display();
56     box4.display();
57     pig3.display();
58     log3.display();
59
60     box5.display();
61     log4.display();
62     log5.display();
63
64     bird.display();
65     platform.display();
66     //log6.display();
67     slingshot.display();
68 }
69

```




	<p>The Bird seems to be constrained to a point now. But we want the bird to fly and launch itself like a projectile when we release the mouse button.</p> <p>Also, we don't want the bird to always move. We want it to move only when we are dragging the mouse.</p> <p>How do we do that? Any ideas?</p>	<p>ESR: We use mouse events like mousePressed, mouseDragged etc.</p>
	<p>The bird is moving everywhere with the mouse because of what we have instructed in the Bird Class.</p> <p>Let's comment on this instruction.</p> <p><i>Teacher comments the lines where Birds' x and y positions are set to mouseX and mouseY.</i></p>	<p><i>The student observes and learns.</i></p>


```

1  class Bird extends BaseClass {
2    constructor(x,y){
3      super(x,y,50,50);
4      this.image = loadImage("sprites/bird.png");
5    }
6
7    display() {
8      //this.body.position.x = mouseX;
9      //this.body.position.y = mouseY;
10     super.display();
11   }
12 }
13

```

Now, our bird doesn't move with the mouse. However, we want the mouse to move when the mouse is dragged on the bird.

We will use a function called `mouseDragged` for this.

Teacher opens the doc from [Teacher Activity 3](#)

This function `mouseDragged` gets invoked whenever you drag the mouse. When we drag the mouse, we will set the position of the bird to be equal to `mouseX` and `mouseY`.

We use `Matter.body.setPosition()` to set the position of the bird body.

Teacher writes the code for the function `mouseDragged()` and runs the code.

The student observes and learns.

Student opens the doc and follows along with the teacher (Student Activity 4)

```
Matter.Body.setPosition(body, position)
```

Sets the position of the body instantly. Velocity, angle, force etc. are unchanged.

Parameters

`body` Body

`position` Vector

@ src/body/Body.js:416

```
js sketch.js
43
44
45
46 background(backgroundImg);
47 Engine.update(engine);
48 strokeWeight(4);
49 box1.display();
50 box2.display();
51 ground.display();
52 pig1.display();
53 log1.display();
54
55 box3.display();
56 box4.display();
57 pig3.display();
58 log3.display();
59
60 box5.display();
61 log4.display();
62 log5.display();
63
64 bird.display();
65 platform.display();
66 //log6.display();
67 slingshot.display();
68 }
69
70 function mouseDragged(){
71   Matter.Body.setPosition(bird.body, {x: mouseX , y: mouseY});
72 }
73
```

Interesting isn't it! The bird seems anchored to a point as if it is attached to a rubber.

The student observes and learns.

When we release the mouse it moves to and fro but is still connected to the point.

We now just want to make the bird fly and detach it from the constraint when the mouse is released.

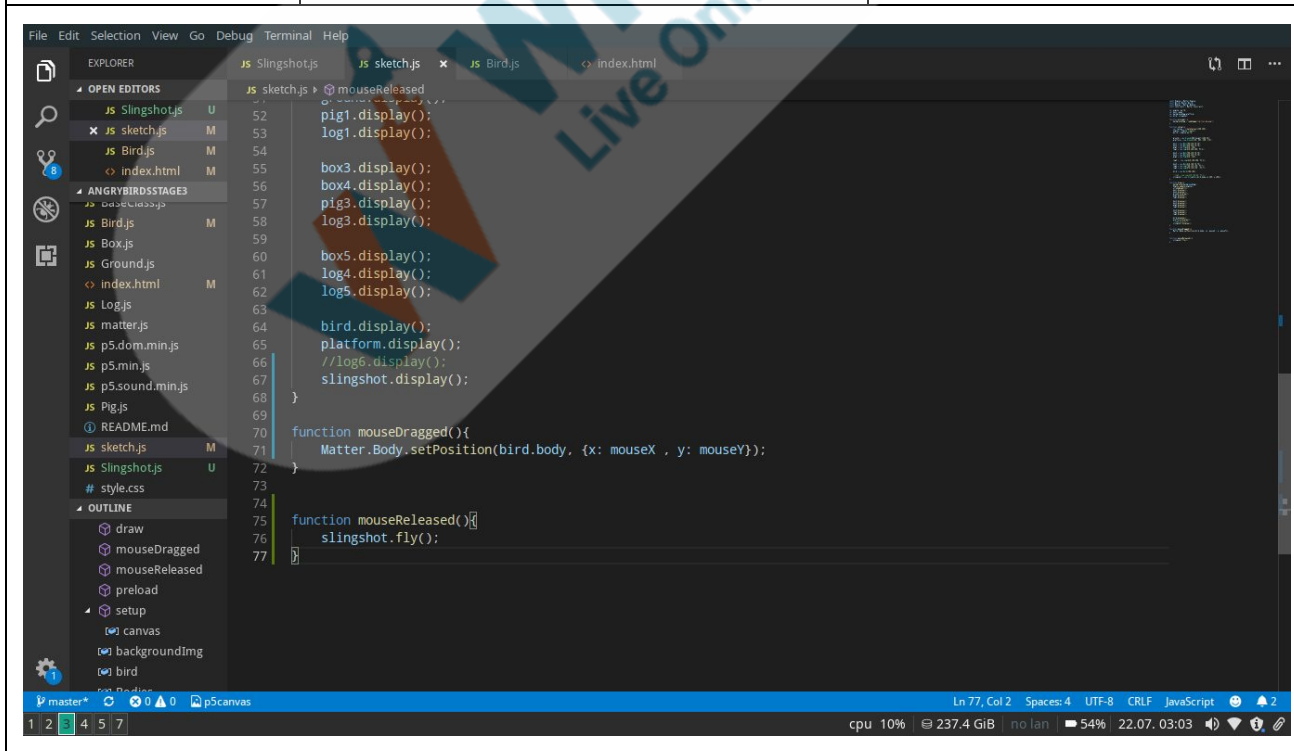
We will give instructions inside another function called `mouseReleased()`

Teacher opens the doc and goes through the doc([Teacher Activity 2](#)).

This function is also called when the mouse is released. For now, let's simply call `slingshot.fly()` inside this function. We haven't defined `slingshot.fly()` but we will.

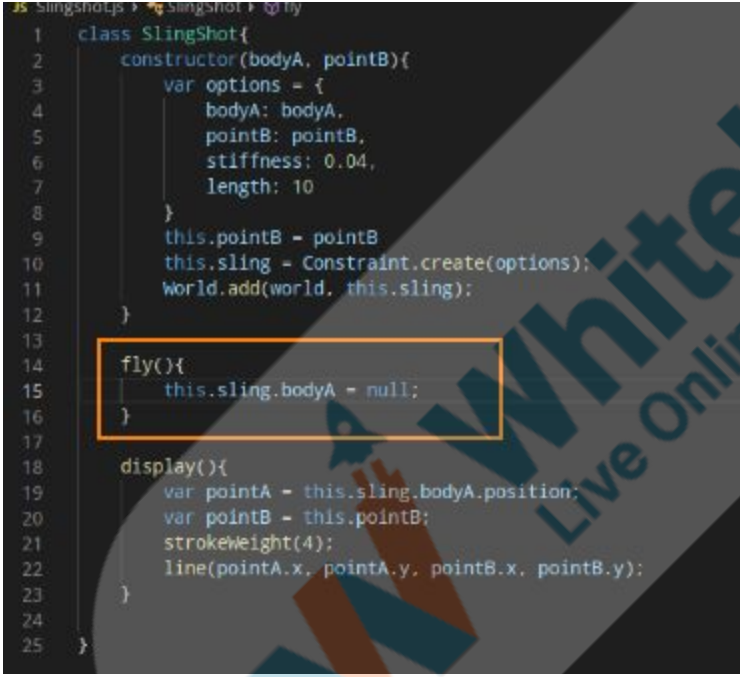
Teacher writes the code.

Student opens the doc and follows with the teacher (Student Activity 3)



```

function mouseReleased() {
  slingshot.fly();
}
  
```

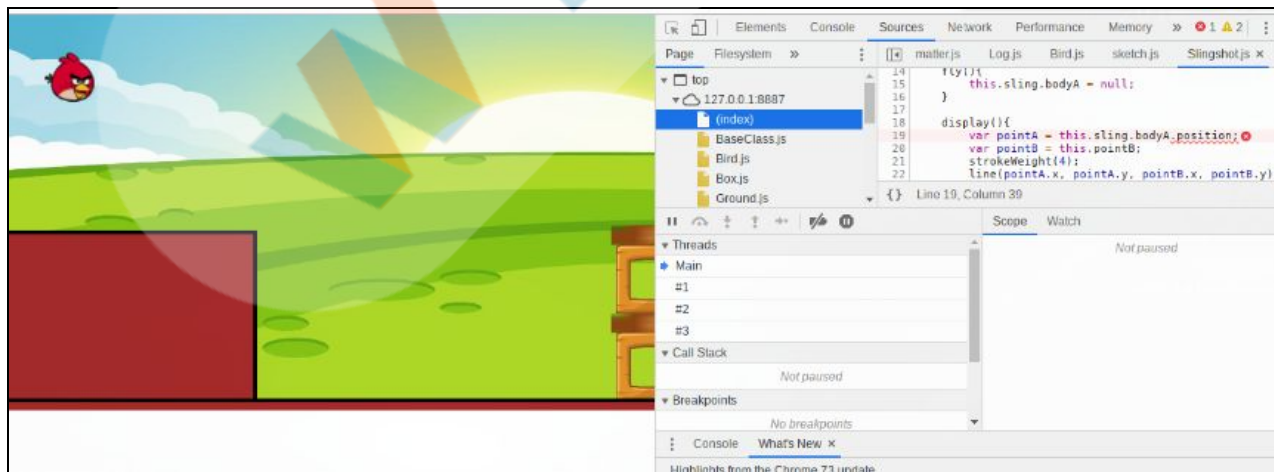
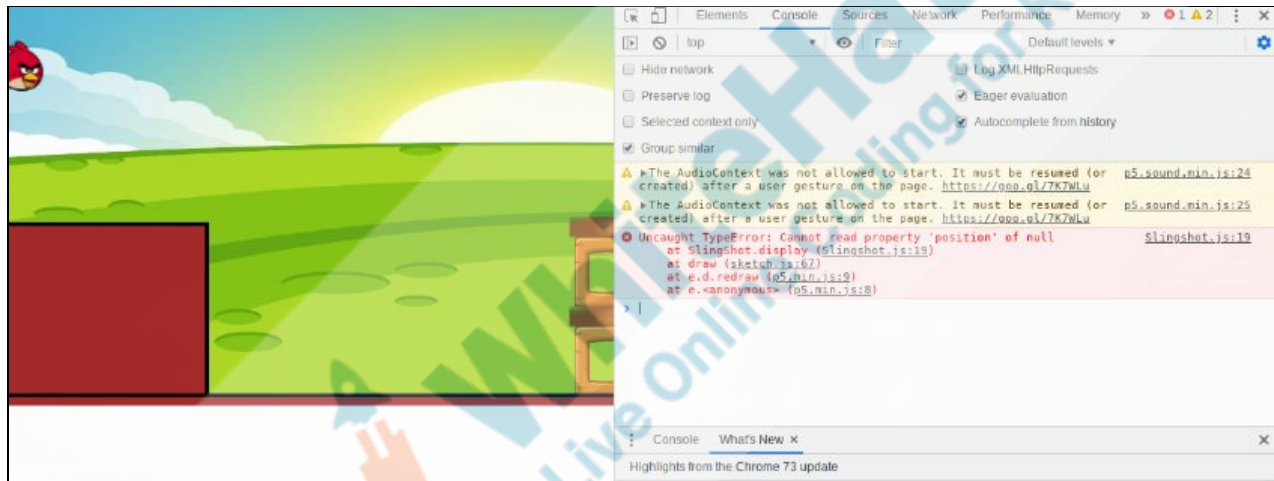
	<p>Let's quickly define slingshot.fly inside the Slingshot class.</p> <p>We will just change the bodyA to null.</p> <p>'null' implies nothing in javascript.</p> <p>Earlier the bodyA was the bird.</p> <p>Attaching nothing to bodyA will free the bird from the constraint.</p> <p><i>Teacher writes the code and runs it.</i></p>	<p><i>The student observes and learns.</i></p>
 <pre> 1 class Slingshot{ 2 constructor(bodyA, pointB){ 3 var options = { 4 bodyA: bodyA, 5 pointB: pointB, 6 stiffness: 0.04, 7 length: 10 8 } 9 this.pointB = pointB 10 this.sling = Constraint.create(options); 11 world.add(world, this.sling); 12 } 13 14 fly(){ 15 this.sling.bodyA = null; 16 } 17 18 display(){ 19 var pointA = this.sling.bodyA.position; 20 var pointB = this.pointB; 21 strokeWeight(4); 22 line(pointA.x, pointA.y, pointB.x, pointB.y); 23 } 24 } 25 </pre>		
	<p>Something weird is happening, everything stops when we are releasing the mouse. What seems to be wrong.</p>	<p><i>The student responds with their explanation.</i></p>
	<p>We can check the error by pressing Ctrl + Shift + J in our chrome browser.</p>	<p><i>The student observes and learns.</i></p>

It seems that even when the bodyA has become null, we are trying to access its position inside the display.

We only want to use the display() function in slingshot if the bodyA is not null.

We can use if() condition here 'null' evaluates to false.

Teacher writes the code and runs.



```

1  class SlingShot{
2      constructor(bodyA, pointB){
3          var options = {
4              bodyA: bodyA,
5              pointB: pointB,
6              stiffness: 0.04,
7              length: 10
8          }
9          this.pointB = pointB
10         this.sling = Constraint.create(options);
11         World.add(world, this.sling);
12     }
13
14     fly(){
15         this.sling.bodyA = null;
16     }
17
18     display(){
19         if(this.sling.bodyA){
20             var pointA = this.sling.bodyA.position;
21             var pointB = this.pointB;
22             strokeWeight(4);
23             line(pointA.x, pointA.y, pointB.x, pointB.y);
24         }
25     }
26 }
27
  
```

Wow! We have bird sling ready. We need to add the image of the catapult and show the catapult rubber getting extended. But we are so close.

Can you do this on your own now?

ESR:

Yes

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

- Create a slingshot effect in the angry birds game.

Step 3: Student-Led Activity (15 mins)	<i>Guide the student to modify the Chain Class to SlingShot class.</i>	<p><i>Student modifies the Chain Class to make it a Slingshot class.</i></p> <p><i>The student also adds the class to index.html.</i></p>
	<i>Guide the student to replace the bodyB in the code to pointB.</i>	<ul style="list-style-type: none"> - Student replaces bodyB in the code to pointB. - Student modifies the constructor and other places where bodyB was used to pointB. - Student changes the sketch.js file to create a Slingshot object with bodyA as bird and pointB. <p><i>Student runs the code and sees the output.</i></p>
	<p><i>Guide the student to edit the Bird Class.</i></p> <p><i>Guide the student to add mouseDragged and mouseReleased functions inside script.js</i></p>	<p><i>Student edits the Bird class so that the bird does not move everywhere with the mouse.</i></p> <p><i>The student uses the mouseDragged() and mousePressed() functions as event triggers in script.js when mouse is dragged and mouse is released.</i></p> <p><i>The student writes code to change the position of the bird when the mouse is dragged.</i></p>

		<i>The student writes code to release the bird when the mouse is released.</i>
	<p>Guide the student to set the bodyA to null when fly() function is invoked.</p> <p>Guide the student to edit the display() function so that it draws the line only when bodyA is not null.</p>	<p>Student changes the Slingshot class to attach the bodyA to null when the fly() function is invoked.</p> <p>Student changes the display function to display the line only when the bodyA is not null.</p>
Teacher Guides Student to Stop Screen Share		
<p style="text-align: center;"><u>FEEDBACK</u></p> <ul style="list-style-type: none"> • Encourage the student to make reflection notes in markdown format. • Complement the student for her/his effort in the class. • Review the content of the lesson. 		
Step 4: Wrap-Up (5 min)	<p>Amazing work! We are so close to finishing off our Angry Birds Game.</p> <p>It has been such an exciting learning journey while creating this game.</p> <p>Do you have any questions or doubts?</p>	<p>Student asks his/her questions or doubts.</p> <p>Teacher helps reconcile the questions.</p>
	<p>Next class, we will be building the mighty Catapult.</p> <p>We will also be adding scores and other features like making the pig disappear when it is hit.</p>	

	How are you feeling?	ESR: varied
	<p>You get a hats off.</p> <p>See you in the next class then.</p>	<p><i>Make sure you have given at least 2 Hats Off during the class for:</i></p> <div>Creatively Solved Activities  +10</div> <div>Great Question  +10</div> <div>Strong Concentration  +10</div>
Project Overview	<p>Note: This is a tiered project with multiple tasks. All students must do the main task. The main task is very similar to the projects that are already live. Each tiered project has two or more additional tasks which are optional.</p> <p>PLUCKING MANGOES</p> <p>Goal of the Project:</p> <p>Today you learnt how to make an elastic constraint called a slingshot for the Angry Bird.</p> <p>In this project, you will have to practice and apply what you have learnt in the class and create a mango plucking game in which you have to add a launcher for the stone, which can be moved with a mouse.</p> <p>Story:</p>	<p><i>Student engages engages with the teacher over the project.</i></p>

	<p>For this summer season Juno is visiting his granny's home. There he saw a mango tree in granny's garden and wanted to eat them. Help him pluck some mangoes by throwing a stone.</p> <p>I am very excited to see your project solution and I know you will do really well.</p> <p>Bye Bye!</p>	
<div>Teacher Clicks</div> <div>✕ End Class</div>		
Additional Activities	<p><i>Encourage the student to write reflection notes in their reflection journal using markdown.</i></p> <p>Use these as guiding questions:</p> <ul style="list-style-type: none"> • What happened today? <ul style="list-style-type: none"> - 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? 	<p><i>Student uses the markdown editor to write her/his reflection as a reflection journal.</i></p>

Activity	Activity Name	Links
Teacher Activity 1	Matter.Constraint Documentation	http://brm.io/matter-js/docs/classes/Constraint.html
Teacher Activity 2	Mouse Released (p5 documentation)	https://p5js.org/reference/#/p5/mouseReleased
Teacher Activity 3	Mouse Dragged (p5 documentation)	https://p5js.org/reference/#/p5/mouseDragged
Teacher Activity 4	Teacher Reference	https://github.com/whitehatjr/AngryBirdsStage4
Student Activity 1	Angry Birds Stage 3.5	https://github.com/whitehatjr/AngryBirdsStage3.5
Student Activity 2	Matter.Constraint Documentation	http://brm.io/matter-js/docs/classes/Constraint.html
Student Activity 3	Mouse Released (p5 documentation)	https://p5js.org/reference/#/p5/mouseReleased
Student Activity 4	Mouse Dragged (p5 documentation)	https://p5js.org/reference/#/p5/mouseDragged