Topic	Catapult and the Rubber Band	
Class Description	**Students will add the catapult and the rubber band in the game.**	
Class	**C29**	
Class time	**45 mins**	
Goal	● Place the images of the catapult in the game ● Use a color picker to pick colors for the rubber ● Draw the rubber band for the catapult	
Resources Required	● Teacher Resources ○ Laptop with internet connectivity ○ Earphones with mic ○ Notebook and pen ● Student Resources ○ 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 mins**

CONTEXT

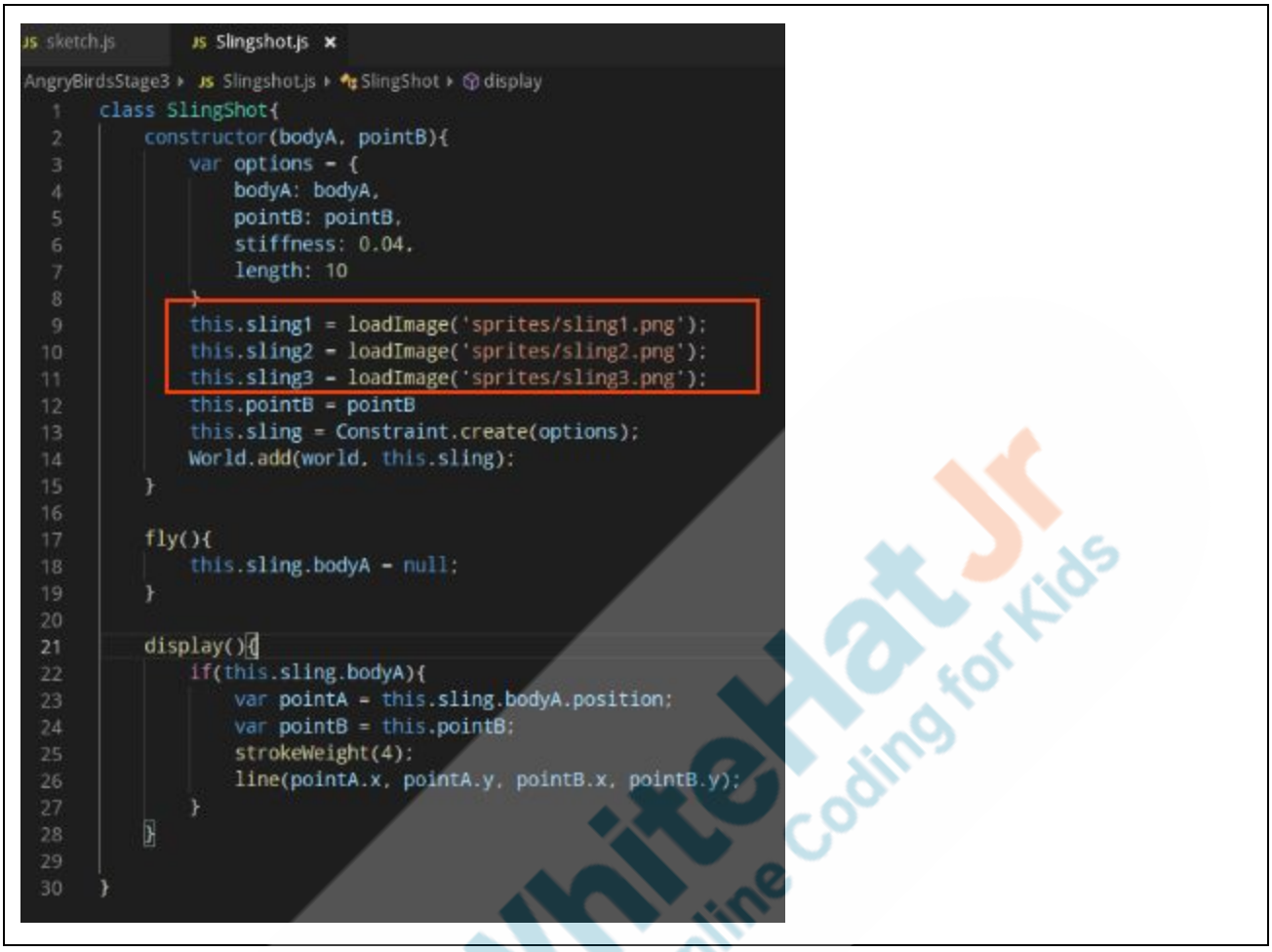
- **Review the concept of constrained bodies and mouse trigger events from the previous classes.**

Class Steps	Teacher Action	Student Action
Step 1: Warm Up (5 mins)	Today, we will be making the catapult for our game - which means the catapult body and the rubber bands. Any ideas on how we can do that?	**ESR:** varied

	Okay these seem to be great ideas. We will see in this class one of the different ways of doing it. But before we go ahead, like everytime let's review the code from the previous class.	*The student reads through the different portions of the previous class code.*
	Before moving forward I have an exciting quiz question for you! Are you ready to answer this question? Teacher click on the  button on the bottom right corner of your screen to start the In-Class Quiz. A quiz will be visible to both you and the student. Encourage the student to answer the quiz question. The student may choose the wrong option, help the student to think correctly about the question and then answer again. After the student selects the correct option, the  button will start appearing on your screen. Click the End quiz to close the quiz pop-up and continue the class.	**ESR:** **Yes**
Teacher Initiates Screen Share		

CHALLENGE • **Load and add the catapult image in the game.** • **Use color picker to give correct r, g, b values to the rubber band line.**		
Step 2: Teacher-led Activity (15 min)	Alright, now that we are warmed up, we want to start with making the catapult. *Teacher helps the student to open the code from the previous class or clone the repo from Student Activity 1 using git clone.* *Teacher opens the link shared by the student and starts live collaboration.*	*Student opens the code from the previous class or clones the repo from Student Activity 1 using git clone.* *The student opens the code in VS Code .* *Student clicks on live share and copies the link and shares it with the teacher.*
$ git clone https://github.com/whitehatjr/AngryBirdsStage4		
	We see that there are three images added to the sprites directory - sling1.png, sling2.png and sling3.png. We will be needing to use these images to create our catapult.	*Student observes.*
	Since, the catapult itself doesn't interact with any object in the game, we can keep it as a static image - we need not create a body for this. We can load and place the image in the game using image() function in p5.js.	

	Let's look at the documentation (**Teacher Activity 2**) to understand how to use image(). *Teacher looks through the p5 documentation for the image() function.*	*Student reads through the documentation.*
	Ok, so we will load the images inside the constructor for slingshot class and position it inside the display() function. *Teacher writes code to load the images in the constructor of slingshot.* *Teacher writes code to position the images in the game using image() function.* Use trial and error to position the image. *Teacher runs the code and shows the output.*	The student observes and learns.

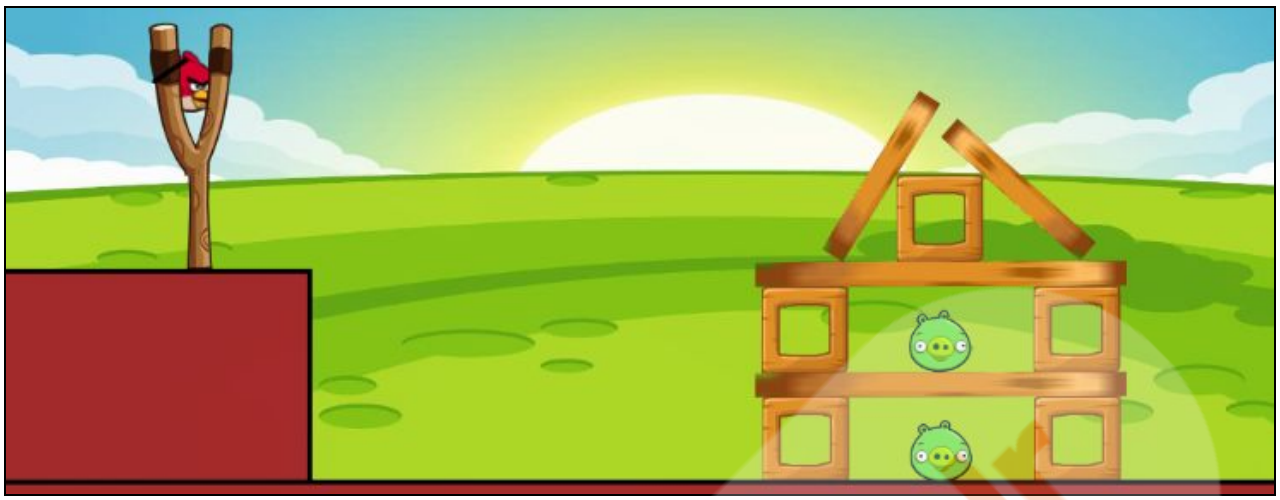


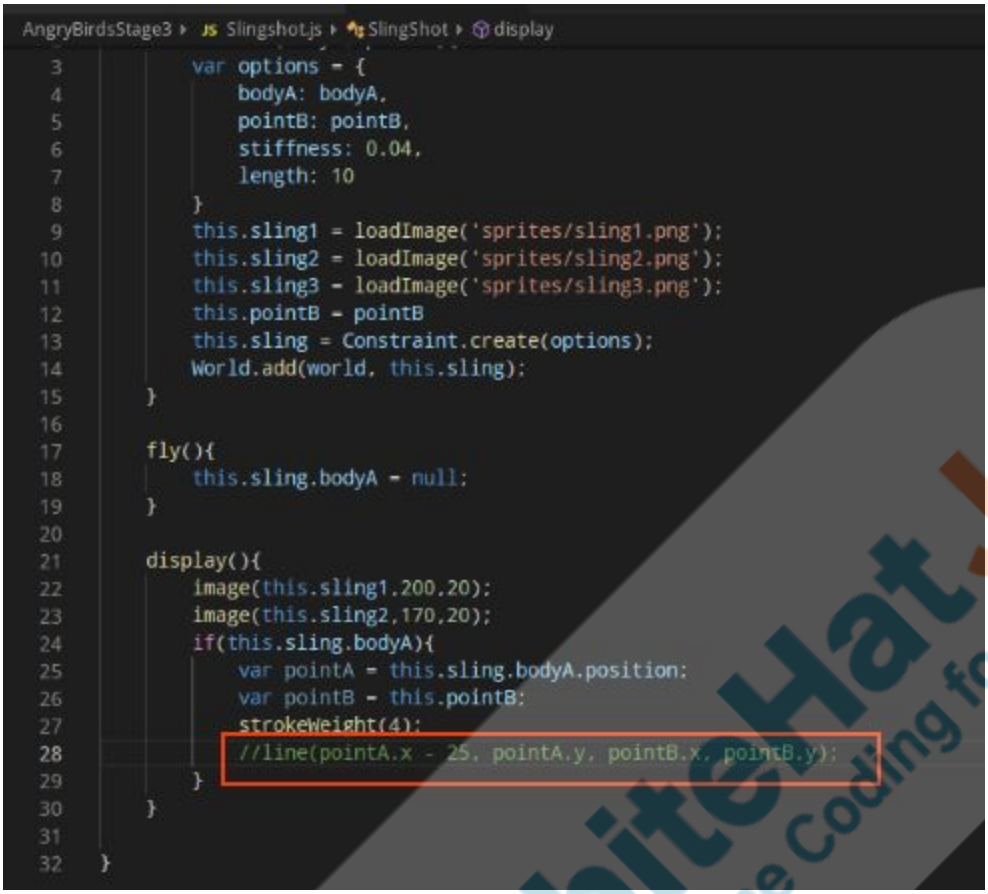
AngryBirdsStage3 ▸ Slingshot.js ▸ SlingShot ▸ display

```
class SlingShot{
    constructor(bodyA, pointB){
        var options = {
            bodyA: bodyA,
            pointB: pointB,
            stiffness: 0.04,
            length: 10
        }
        this.sling1 = loadImage('sprites/sling1.png');
        this.sling2 = loadImage('sprites/sling2.png');
        this.sling3 = loadImage('sprites/sling3.png');
        this.pointB = pointB
        this.sling = Constraint.create(options);
        World.add(world, this.sling);
    }

    fly(){
        this.sling.bodyA = null;
    }

    display(){
        image(this.sling1,200,20);
        image(this.sling2,170,20);
        if(this.sling.bodyA){
            var pointA = this.sling.bodyA.position;
            var pointB = this.pointB;
            strokeWeight(4);
            line(pointA.x, pointA.y, pointB.x, pointB.y);
        }
    }

}
```

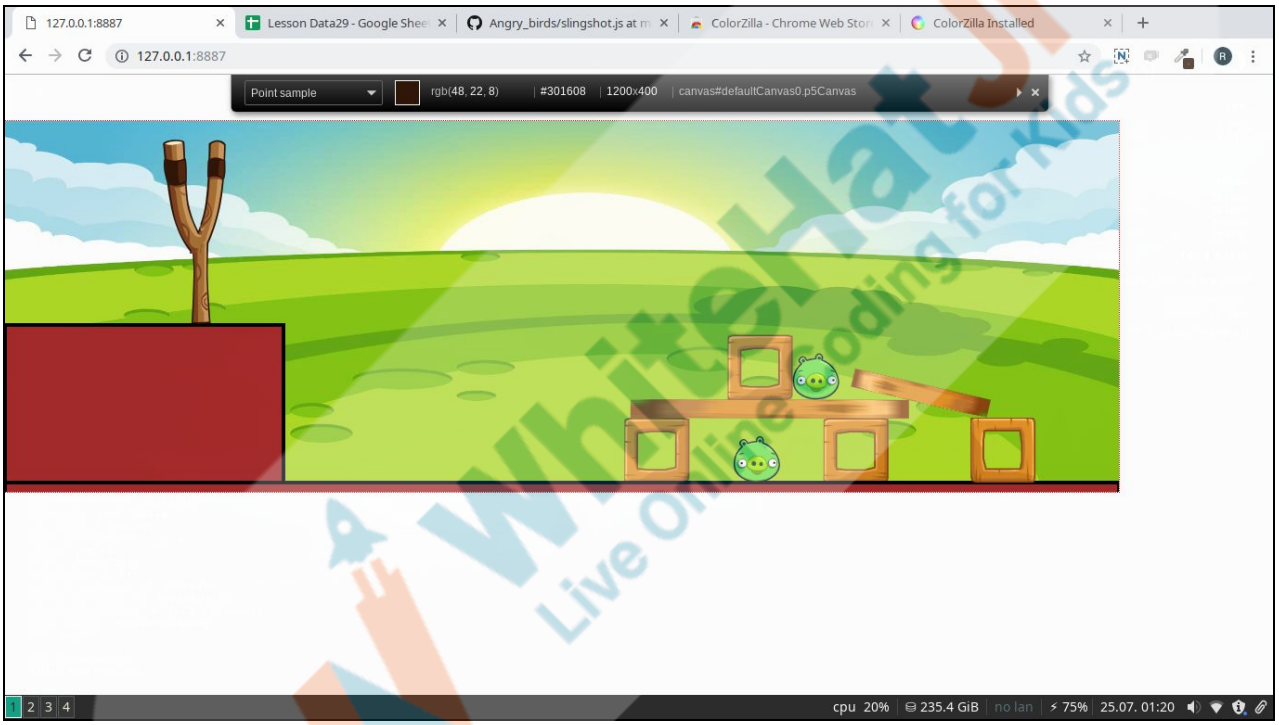
	Ok, we seem to have the catapult in the right position. Our bird needs to be higher.	*The student observes and learns.*

	We will have to change the position for both the bird and the point to which it is anchored. *Teacher modifies script.js to change the position of the bird and the point to which it is anchored.* *Teacher uses trial and error to find the positions.* *Teacher runs the code and shows the output.*	

		
	Wow, we have the catapult and the bird in between the two ends. But we don't want the line from the anchor point. *Teacher comments the line() in the display function and runs the code.*	*The student observes and learns.*



	But we do want two rubber bands from either end of the catapult. How will we get that? What color would be these rubber bands / lines?	**ESR:** We can draw two lines from the two ends of the catapult behind the bird. **ESR:** The same as the rubber color on the catapult image.
	Any ideas on how we can identify the exact color and draw the line using the same color? All colors are made up of red, green and blue. There are thousands of colors by combining different amounts of red, green and blue. Fortunately,	*Student installs the Chrome Plugin **(Student Activity 3)**.*

	we have color pickers which can pick up the color from the screen and tell us the right amount of red, green and blue needed to make the color. Teacher guides the student on how to install and use the ColorZilla chrome extension (**Student Activity 3**) and use it to identify the color of the rubber band.	
		
	Alright, we have the exact color now. Can you now draw the lines from the two ends of the catapult?	-
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

- **Draw the lines of the rubber band from the two ends of the catapult.**
- **Use conditional programming to give different strokes and position for the rubber band when the ball is ahead and behind the catapult.**

Step 3: Student-Led Activity (15 min)	*Guide the student to use a color picker from the rubber band in the image.*	*The student chooses the color of the rubber band using the ColorZilla chrome plugin's color picker tool.* *The student uses the color as the stroke color.*

AngryBirdsStage3 ▸ Slingshot.js ▸ SlingShot ▸ display

```
class SlingShot{
    constructor(bodyA, pointB){
        var options = {
            bodyA: bodyA,
            pointB: pointB,
            stiffness: 0.04,
            length: 10
        }
        this.sling1 = loadImage('sprites/sling1.png');
        this.sling2 = loadImage('sprites/sling2.png');
        this.sling3 = loadImage('sprites/sling3.png');
        this.pointB = pointB
        this.sling = Constraint.create(options);
        World.add(world, this.sling);
    }

    fly(){
        this.sling.bodyA = null;
    }

    display(){
        image(this.sling1,200,20);
        image(this.sling2,170,20);
        if(this.sling.bodyA){
            var pointA = this.sling.bodyA.position;
            var pointB = this.pointB;
            push();
            strokeWeight(4);
            stroke(rgb(48,22,8));
            line(pointA.x - 25, pointA.y, pointB.x, pointB.y);
            pop();
        }
    }
```

	Guide the student to draw the lines.	*Student draws line 1 from one end of the catapult to behind the bird.* *Student runs code and checks the output.*


AngryBirdsStage3 ▸ Slingshot.js ▸ SlingShot ▸ display
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.sling1 = loadImage('sprites/sling1.png');
10 this.sling2 = loadImage('sprites/sling2.png');
11 this.sling3 = loadImage('sprites/sling3.png');
12 this.pointB = pointB
13 this.sling = Constraint.create(options);
14 World.add(world, this.sling);
15 }
16
17 fly(){
18 this.sling.bodyA = null;
19 }
20
21 display(){
22 image(this.sling1,200,20);
23 image(this.sling2,170,20);
24 if(this.sling.bodyA){
25 var pointA = this.sling.bodyA.position;
26 var pointB = this.pointB;
27 push();
28 strokeWeight(4);
29 stroke(rgb(48,22,8));
30 line(pointA.x - 25, pointA.y, pointB.x, pointB.y);
31 pop();
32 }
33 }		

	Guide the student to draw the lines.	*Student draws line 2 from the other end of the catapult to behind the bird.* *Student runs code and checks the output.*

	Guide the student to load and position the image.	*The student loads - sling3.png - and positions it behind the bird as the base of the rubber band.* *Student runs code and checks the output.*

AngryBirdsStage3 › SlingshotJs › SlingShot › display

```
class SlingShot{
    constructor(bodyA, pointB){
        var options = {
            bodyA: bodyA,
            pointB: pointB,
            stiffness: 0.04,
            length: 10
        }
        this.sling1 = loadImage('sprites/sling1.png');
        this.sling2 = loadImage('sprites/sling2.png');
        this.sling3 = loadImage('sprites/sling3.png');
        this.pointB = pointB
        this.sling = Constraint.create(options);
        World.add(world, this.sling);
    }

    fly(){
        this.sling.bodyA = null;
    }

    display(){
        image(this.sling1,200,20);
        image(this.sling2,170,20);
        if(this.sling.bodyA){
            var pointA = this.sling.bodyA.position;
            var pointB = this.pointB;
            push();
            strokeWeight(5);
            stroke(48,22,8);
            line(pointA.x - 20, pointA.y, pointB.x -10, pointB.y);
            line(pointA.x - 20, pointA.y, pointB.x + 30, pointB.y - 3);
            image(this.sling3,pointA.x -30, pointA.y -10,15,30);
            pop();
```

	Great! We have the rubber band effect of the catapult ready. Now, there is only one small problem. Observe what happens when the rubber band is pulled ahead of the catapult. What can we do about this?	*Student observes how the base of the rubber band is still behind the bird even when the bird is pulled forward to the catapult.* **ESR:** We can use conditional programming to draw different lines at different end points depending on the position of the bird with respect to the catapult. *Student runs code and checks the output.*
		
	Awesome. Let's do this. We can also adjust the strokeWeights so that the rubber band appears thinner when the bird is pulled forward. *Guide the student to write the conditional code.*	*The student writes the code to draw the lines at different end points.* *Student runs the code to see the output.*

AngryBirdsStage3 › Slingshot.js › SlingShot › display

```
    fly(){
        this.sling.bodyA = null;
    }

    display(){
        image(this.sling1,200,20);
        image(this.sling2,170,20);
        if(this.sling.bodyA){
            var pointA = this.sling.bodyA.position;
            var pointB = this.pointB;
            push();

            stroke(48,22,8);
            if(pointA.x < 220) {
                strokeWeight(7);
                line(pointA.x - 20, pointA.y, pointB.x -10, pointB.y);
                line(pointA.x - 20, pointA.y, pointB.x + 30, pointB.y - 3);
                image(this.sling3,pointA.x -30, pointA.y -10,15,30);
            }
            else{
                strokeWeight(3);
                line(pointA.x + 25, pointA.y, pointB.x -10, pointB.y);
                line(pointA.x + 25, pointA.y, pointB.x + 30, pointB.y - 3);
                image(this.sling3,pointA.x + 25, pointA.y -10,15,30);
            }


            pop();
        }
    }

}
```

	Wow! Great job. Our grand catapult is ready to fire at the pigs. It wasn't as hard as we thought, was it?	**ESR:** varied
Teacher Guides Student to Stop Screen Share		
FEEDBACK ● **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 mins)**	Let's wrap up today's class. Can you capture all that we learned today?	**ESR:** - How to place and position images in the game.

		- How to create a rubber band effect in the game.
	Great! There are still a few things left in our game. We want a way to make the pig disappear from the game after it has been hit. We might want to add the score and other small things which we will cover in our next class. You get a hats off. See you in the next class then.	*Make sure you have given at least 2 Hats Off during the class for:*   
Project Overview	## TOWER SIEGE - 1 **Goal of the Project:** Today, you have learnt the concept of adding rubber bands and adjusting the points for the slingshot using the concept of constraints. In this project, you will have to practice and apply what you have learnt in the class and create a Tower Siege Game using Constrained Bodies. **Story:** In the game design competition in your school, you are asked to make a game related to knocking down objects. Create a Tower Siege Game where your friends can throw a rock at a	*Student engages engages with the teacher over the project.*

	group of stacked objects and crash them. I am very excited to see your project solution and I know you will do really well. Bye Bye!	
Teacher Clicks 		
Additional Activities	*Encourage the student to write reflection notes in their reflection journal using markdown.* Use these as guiding questions: ● 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?	*Student uses the markdown editor to write her/his reflection as a reflection journal.*

Activity	Activity Name	Links
Teacher Activity 1	Angry Birds Stage 4	https://github.com/whitehatjr/AngryBirdsStage4
Teacher Activity 2	image() p5.js documentation	https://p5js.org/reference/#/p5/image
Teacher Activity 3	Color picker extension	https://chrome.google.com/webstore/detail/colorzilla/bhlhnicpbhignbdhedgjhgdocnmhomnp?hl=en
Teacher Activity 4	Reference Link	https://github.com/whitehatjr/AngryBirdsStage5
Student Activity 1	Angry Birds Stage 4	https://github.com/whitehatjr/AngryBirdsStage4
Student Activity 2	image() p5.js documentation	https://p5js.org/reference/#/p5/image
Student Activity 3	Color picker extension	https://chrome.google.com/webstore/detail/colorzilla/bhlhnicpbhignbdhedgjhgdocnmhomnp?hl=en