

## INSTRUCTIONS:

---

### Goal of the Project:

In Class 29, you have learnt the concept of adding rubber bands and adjusting the points for the slingshot using the concept of constraints.

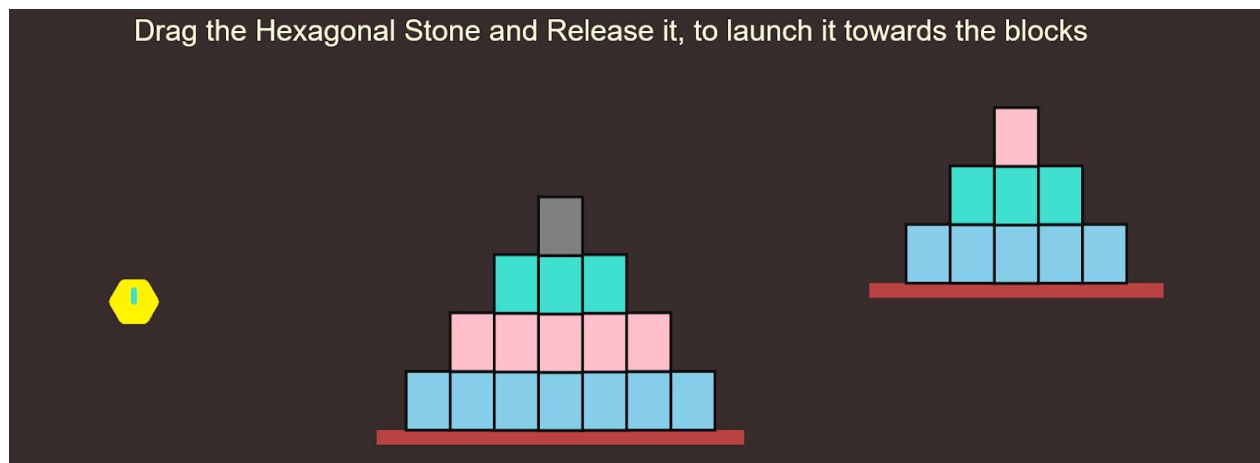
In this project, you will have to practice and apply what you have learnt in the class and create a Tower Siege Game using Constrained Bodies.

### 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 group of stacked objects and crash them.

See a video of this in action [here](#).



**\*This is just for your reference. We expect you to apply your own creativity in the project.**

### Getting Started:

1. Use the template on github, available for download on the following link:  
<https://github.com/rupinwhitehatjr/TowerSiege-1>
2. **Unzip** this folder.
3. Rename the unzipped folder as **Project 29**.
4. **Import** this folder into **VS Code**.
5. Start editing your code in **sketch.js**.

### Specific Tasks to complete the Project:

1. Make a copy of the following files created in the Angry birds game in the class:
  - a. Ground JavaScript file
  - b. Box JavaScript file ,
  - c. Slingshot JavaScript
2. Add these files to your project and add these file names to index.html.
3. In sketch.js, create an object using **Ground Class**.
4. Create an **object for the Stand** (which is the base to hold the blocks) **using the same Ground Class**.
5. Create multiple objects using the **Box Class** and stack them in a way that the structure looks like a pyramid.
  - Here is an example of of the code snippet for creating this structure:

```
//level two
block8 = new Block(330,235,30,40);
block9 = new Block(360,235,30,40);
block10 = new Block(390,235,30,40);
block11 = new Block(420,235,30,40);
block12 = new Block(450,235,30,40);
//level three
block13 = new Block(360,195,30,40);
block14 = new Block(390,195,30,40);
block15 = new Block(420,195,30,40);
//top
block16 = new Block(390,155,30,40);
```

6. Create an **object called Polygon using Bodies** and add it to the world.
7. Add an image to the polygon object using the concepts taught in class C25. (See **Hints on how to do this.**)
8. **Link the Polygon (bodyA) to the Slingshot Class** with a **pointB**.
9. Adjust the position of the Polygon such that it is aimed at the pyramid at a certain distance from the stand.

10. Add the **mouseDragged** and **mouseReleased** events in the sketch.js to launch the Polygon on the pyramid.
11. Update the **fly()** method to set the **bodyA** to **null**.
12. Make sure the project works before you submit it.

\*Refer to the images given above for reference.

### Submitting the Project:

1. Upload your completed project to your own github account.
2. Create a new repository named "**Project 29**"
3. **Upload** working code to this github repository.
4. Enable Github pages for the repository.
5. Copy the link to the github pages link in the Student Dashboard.

**Hints:**

1. Create the Polygon using Bodies.
  - There are many types of Bodies to give shape.
  - Refer to the link: <https://brm.io/matter-js/docs/classes/Bodies.html>Here is one example for you:

```
//polygon holder with slings  
polygon = Bodies.circle(50,200,20);  
World.add(world,polygon);  
  
slingShot = new Slingshot(this.polygon,{x:100,y:200});
```

2. When adding images, ensure your imageMode is CENTER.
  - For reference: <https://p5js.org/reference/#/p5/image>

```
imageMode(CENTER)  
image(polygon_img ,polygon.position.x,polygon.position.y,40,40);
```

**\*Note:** **mouseDragged** and **mouseReleased** are the default, they need not be called so feel free to experiment with its functionality.

Links for reference:

1. <https://p5js.org/reference/#/p5/mousePressed>
2. <https://p5js.org/reference/#/p5/mouseDragged>

**REMEMBER...** Try your best, that's more important than being correct.

After submitting your project your teacher will send you feedback on your work.

\_\_\_\_\_ **xxx** \_\_\_\_\_ **xxx** \_\_\_\_\_ **xxx** \_\_\_\_\_ **xxx** \_\_\_\_\_ **xxx** \_\_\_\_\_