

## INSTRUCTIONS:

---

### Goal of the Project

In Class 33, you have learned the concept of debugging.

In this project, you will apply what you have learned in the class to achieve the following goals.

<b>Main Goal</b>	<ul style="list-style-type: none"><li>• Debug the code and get the output.</li></ul>
<b>Additional Goal 1</b>	<ul style="list-style-type: none"><li>• Design the game design by adding scoring system text.</li></ul>
<b>Additional Goal 2</b>	<ul style="list-style-type: none"><li>• Write code to increment the score.</li></ul>
<b>Additional Goal 3</b>	<ul style="list-style-type: none"><li>• Add the game end functionality after some turns.</li></ul>
<b>Additional Goal 4</b>	<ul style="list-style-type: none"><li>• Write code to increment the scores randomly.</li></ul>

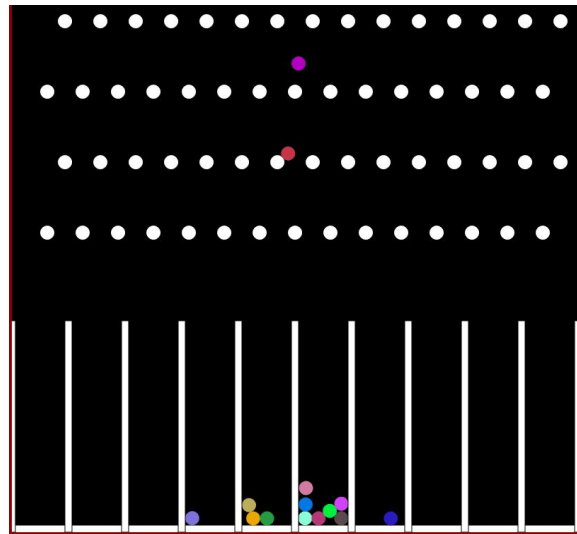
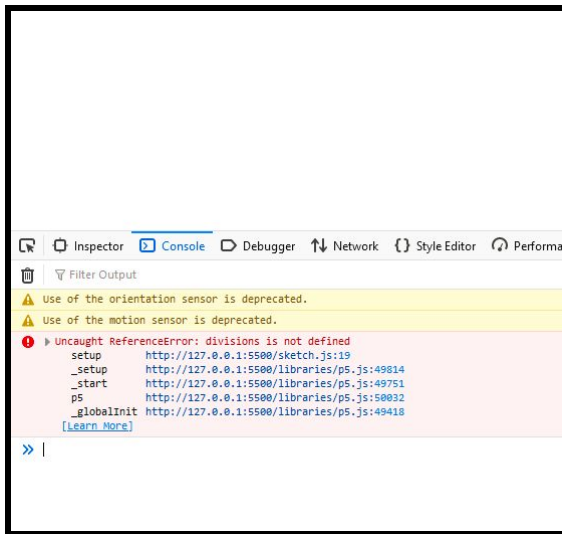
### Story:

Honey has seen the plinko game played by various contestants on the TV.

Plinko is a game played on a nearly vertical board populated with offset rows of pegs. The player chooses one of five slots in the top of the board, drops the chip into it and watches as the chip bounces down the board. Each time the chip encounters a peg, it will either bounce left or right.

Now Honey wants to make this game for herself so she can play anytime.

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 [here](#).
2. **Unzip** this folder, rename the unzipped folder as **Project-33**.
3. **Import** this folder into **VS Code**.
  - Click on file -> Open Folder -> Select the folder that we renamed in the correct location.
4. Start editing your code in **sketch.js**.

### Specific Tasks to Achieve the Main Goal:

1. The project template provided to you has a few bugs in it.
2. Make sure the project works before you submit it.

### Submitting the Project:

1. Upload your completed project to your own GitHub account.
2. Create a New Repository named **"Project 33"**.
3. **Upload** working code to this GitHub repository.
4. Enable GitHub pages for your repository.
5. Copy the link to the GitHub pages link in the Student Dashboard.

### Hints for the Main Goal:

1. Debugging Hints:  
Check to see if:

PROFESSIONAL

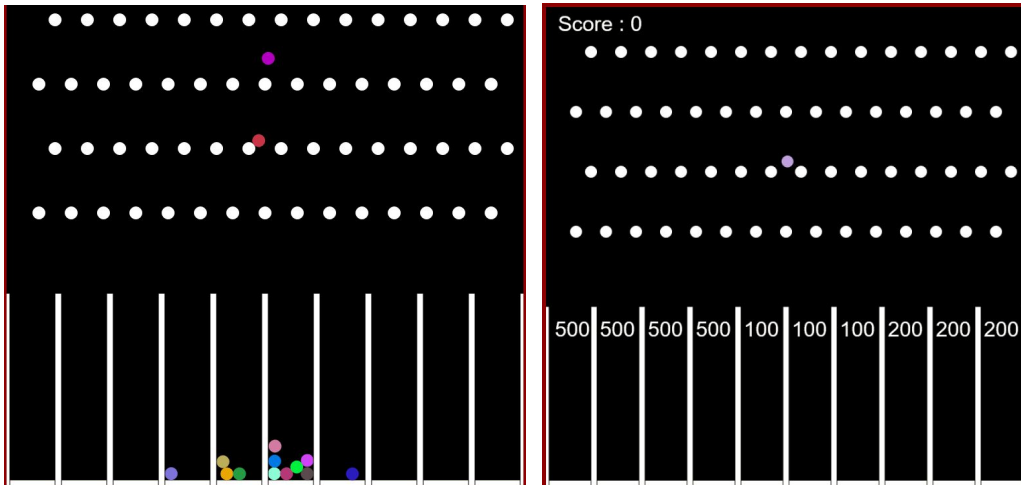
## PLINKO WITH SCORES



- files are not included in index.html.
- variables are not created.
- class is called in a wrong manner.
- the function of a class is not called.

### Additional Goal 1:

Now add score text in the game and create particles on mouse click.



### Specific Tasks to Achieve Additional Goal 1:

1. Create a variable score and initialize the score to 0.
2. Because there will always be one particle while calculating the score, we need to create a particle variable (not an array).

```
var particle;
```

3. Create a variable **count** and initialize the turn to 0.
  - The users should get 5 turns to maximise their score.
4. **Display the score** at a desired position using text.
5. Specify the points in between the divisions using text.
  - The score should get updated with the number of points specified in the division where the ball falls.
6. Create a **gamestate** as start or play.

7. Use a **mousePressed()** function to create a new particle and assign it to the “particle” variable.

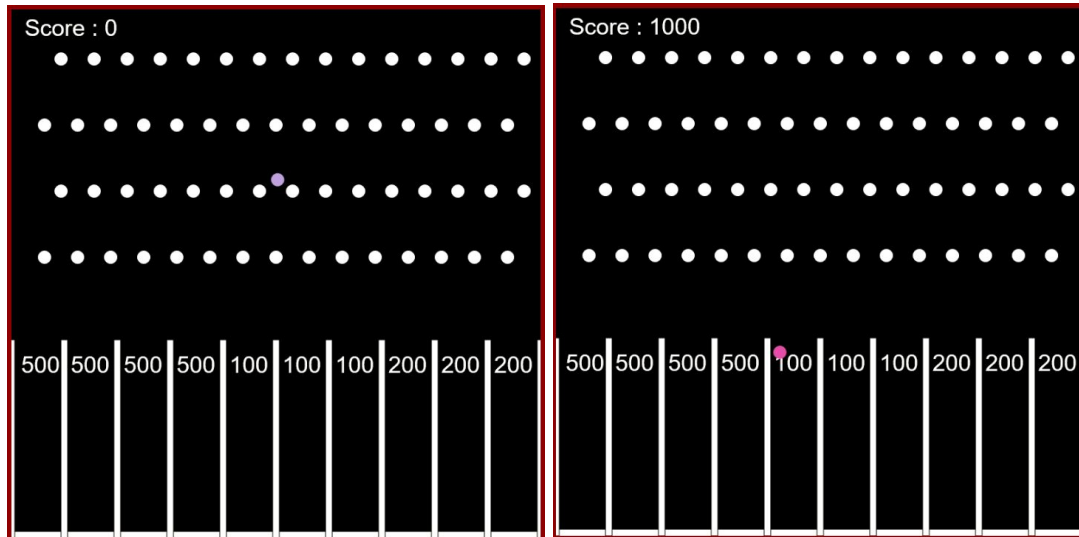
```
function mousePressed()
{
  if(gameState!="end")
  {
    particle=new Particle(mouseX, 10, 10, 10);
  }
}
```

8. Make sure the project works before you submit it.

**\*SAVE** all the changes made to the project and **SUBMIT** the shareable link in the Student Dashboard Projects panel against the correct class number.

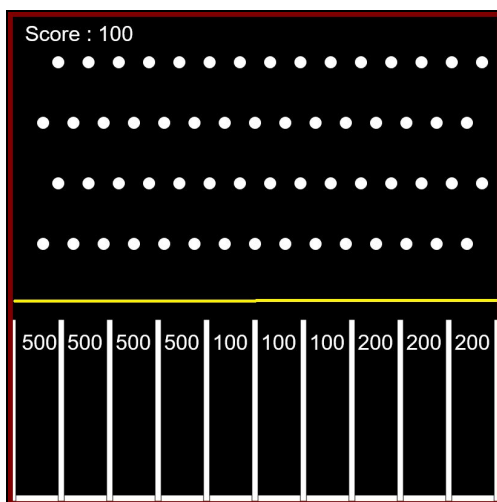
### Additional Goal 2:

Here, you have to help Honey in incrementing the scores.



### Specific Tasks to Achieve Additional Goal 2:

1. If the particle has crossed the yellow line (shown in the image below), you will know that the particle will now fall inside one of the buckets.



2. Once the particle crosses the yellow line, to determine in which division the particle is, add the code for the following logic:
  - If the particle's **x position is less than 300** then the **score is 500** points.

- If the particle's **x position is more than 301 and less than 600** then the **score is 100** points.
- If the particle's **x position is more than 601 and less than 900** then the **score is 200** points.
  - Here you see a sample code for the first condition.

```
if (particle != null)
{
    particle.display();

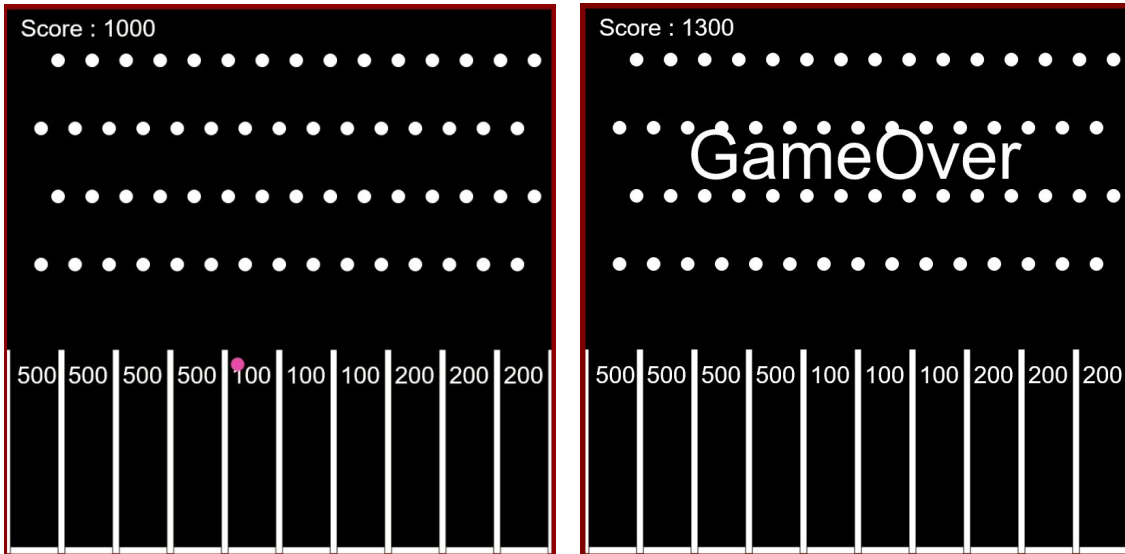
    if (particle.body.position.y > 760)
    {
        if (particle.body.position.x < 300)
        {
            score = score + 500;
            particle = null;
        }
    }
}
```

3. Once the score is calculated, make sure you set the **particle variable** to **null**.
4. Make sure the project works before you submit it.

**\*SAVE** all the changes made to the project and **SUBMIT** the shareable link in the Student Dashboard Projects panel against the correct class number.

**Additional Goal 3:**

Now, add the game end functionality, when the player's turn crosses 5 count.

**Specific Tasks to Achieve Additional Goal 3:**

1. For every turn played, **increase** the **count** variable by **1**.
2. If the player has played 5 times:
  - The game is over.
  - **gameState** is END.
  - Show that the game has ended.

```
if ( count >= 5) gameState = "end";
```

3. Make sure the project works before you submit it.

**\*SAVE** all the changes made to the project and **SUBMIT** the shareable link in the Student Dashboard Projects panel against the correct class number.



#### Additional Goal 4:

You can add the code to increment the score randomly.

#### Specific Tasks to Achieve Additional Goal 3:

1. Randomly assign scoring values to the divisions in multiples of 50.
2. Use an array to keep track of each division's X and Y values.
3. Use the array and the score value of a division, to calculate the user's score value.
4. Make sure the project works before you submit it.

**\*SAVE** all the changes made to the project and **SUBMIT** the shareable link in the Student Dashboard Projects panel against the correct class number.

**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 \_\_\_\_\_