PROFESSIONAL

PLINKO WITH SCORES

INSTRUCTIONS:

Goal of the Project

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

In this project, you will have to practice and apply what you have learnt in the class and develop the Plinko Game with a Score Card.

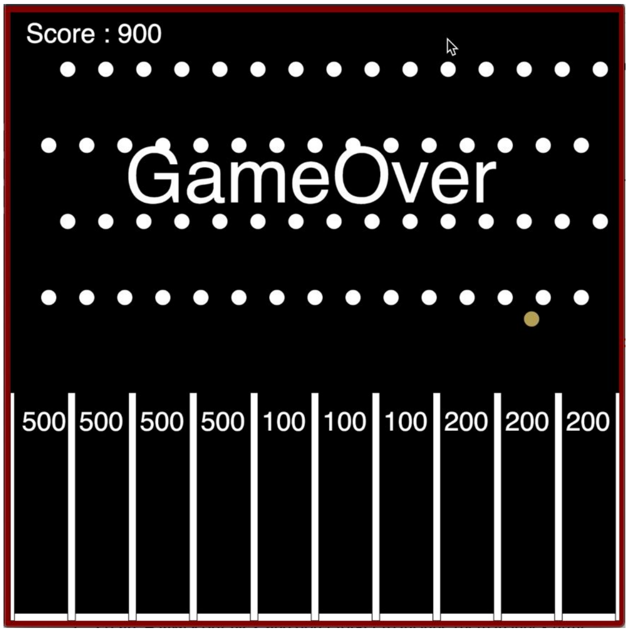
Story:

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.**

PROFESSIONAL

PLINKO WITH SCORES

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 complete the Project:

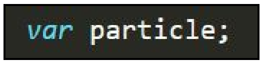
There are two activities that you have to do in this project.

Activity 1: Debugging:

1. The project template provided to you has a few bugs in it.
2. Resolve them before you do Activity 2.

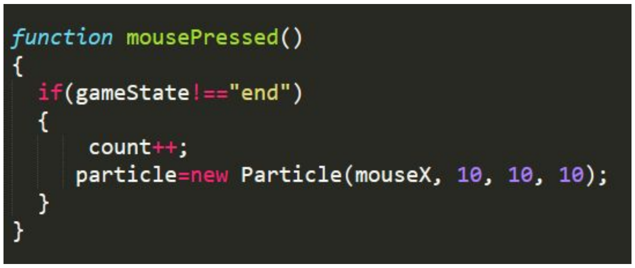
Activity 2: Adding Scores in Plinko

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).



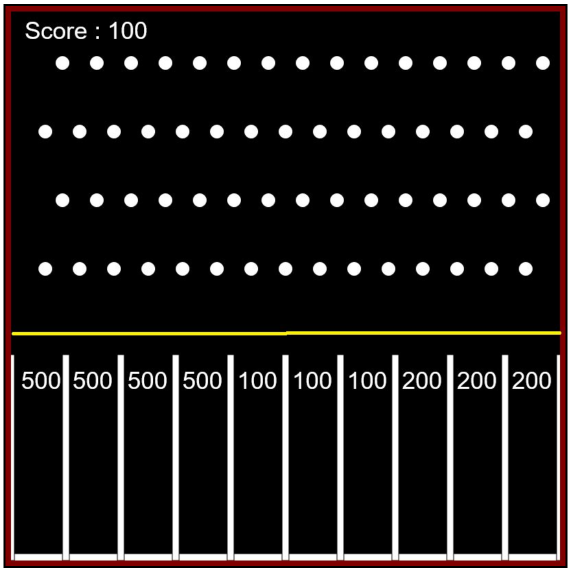
```
var particle;
```

3. Create a **variable turn** 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.



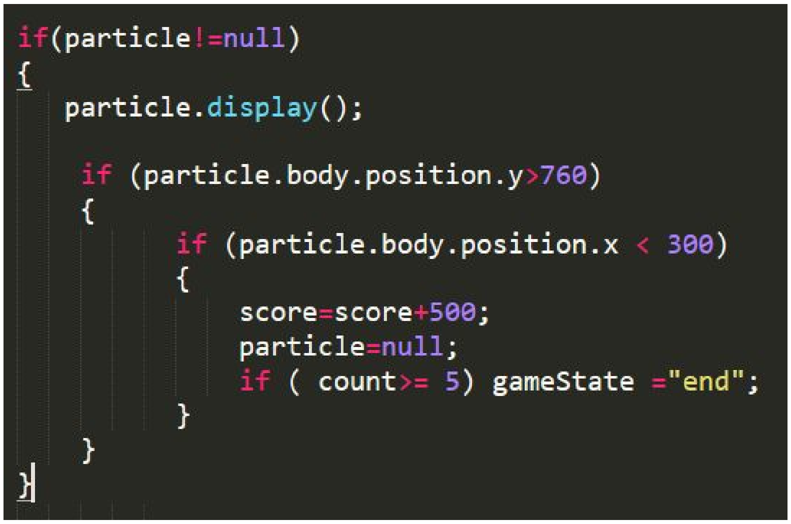
```
function mousePressed()
{
  if(gameState!=="end")
  {
      count++;
     particle=new Particle(mouseX, 10, 10, 10);
  }
}
```

8. 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.



9. 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.

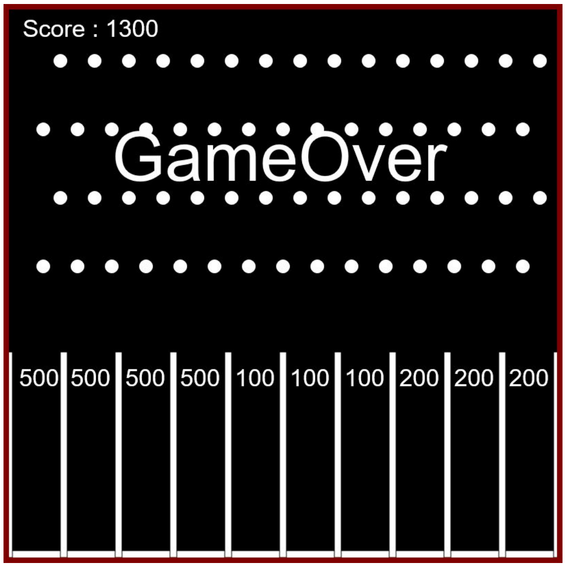
     Here you see a sample code for the first condition.



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

    if (particle.body.position.y>760)
    {
          if (particle.body.position.x < 300)
          {
              score=score+500;
              particle=null;
              if ( count>= 5) gameState ="end";
          }
    }
}
```

10. Once the score is calculated, make sure you set the **particle variable** to **null**.
11. For every turn played, **increase** the **turn variable** by **1**.
12. If the player has played 5 times:
    - The game is over.
    - **gameState** is END.
    - Show that the game has ended.



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.

Additional Challenging Activity:

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.

PROFESSIONAL

PLINKO WITH SCORES

Hints :

1. Debugging Hints:
   Check to see if:
   - 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.

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 ———