PROBLEM 3 - ▲■●

(LINK)

The program uses p5.js to create an interactive and dynamic composition of randomly generated shapes (circles, squares, and triangles) on the screen. The background is set to white only once inside the `setup()` function (`background(255);`), so all new shapes are drawn on top of previous ones and accumulate over time, creating an abstract visual effect.

In the `setup()` block, the code initializes the canvas to fill the browser window (`createCanvas(windowWidth, windowHeight);`) and ensures that rectangles and ellipses are drawn from their center (`rectMode(CENTER);` and `ellipseMode(CENTER);`). The default stroke is removed with `noStroke();` so shapes do not have a border by default.

The core drawing logic happens in the `draw()` function, which is called repeatedly by p5.js (about 60 times per second). Each time `draw()` executes, the code first chooses a **random position** on the canvas by generating values for `x` and `y` using `random(width)` and `random(height)`. It then determines a **random size** for the next shape with `let size = random(30, 120);`, so each shape appears in a variety of dimensions.

Next, the program sets a **random fill color** with some transparency: `fill(random(255), random(255), random(255), random(70, 170));`. The four arguments correspond to red, green, blue, and alpha (opacity), with alpha ensuring that overlapping shapes create interesting blends.

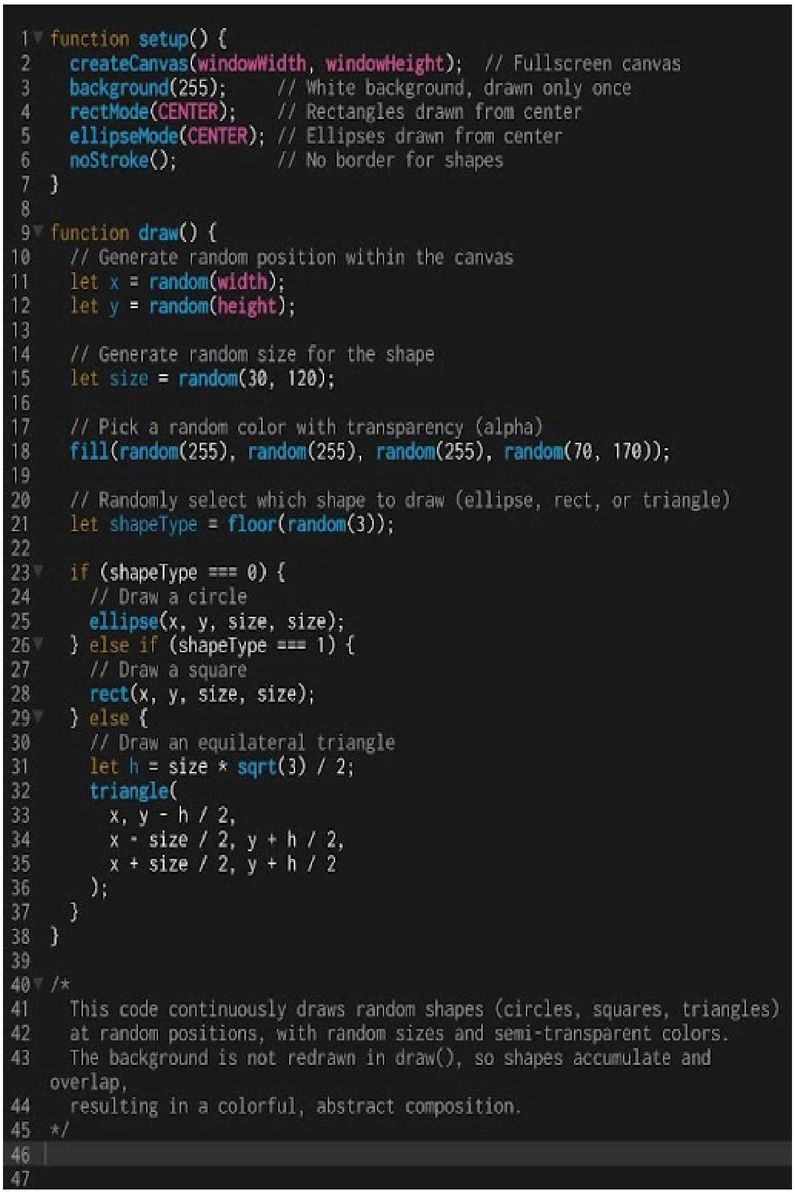
The type of shape to draw is decided randomly as well: `let shapeType = floor(random(3));`. This produces either 0 (ellipse), 1 (rectangle), or 2 (triangle).

- If `shapeType === 0`, the code draws a **circle** at the chosen position and size with `ellipse(x, y, size, size);`.

- If `shapeType === 1`, it draws a **square** using `rect(x, y, size, size);`, centered at `(x, y)`.

- If `shapeType === 2`, the code calculates the height for an equilateral triangle and draws the **triangle** using the `triangle()` function. The coordinates are calculated so that the triangle is centered at `(x, y)`.

Because the background is only set in `setup()` and not redrawn in `draw()`, shapes **accumulate** and overlap, resulting in a lively, evolving, and colorful abstract image. Transparency and random choices for position, size, and color make every composition unique and unpredictable, giving the sketch a playful and artistic character.



```
function setup() {
  createCanvas(windowWidth, windowHeight);  // Fullscreen canvas
  background(255);     // White background, drawn only once
  rectMode(CENTER);    // Rectangles drawn from center
  ellipseMode(CENTER); // Ellipses drawn from center
  noStroke();          // No border for shapes
}

function draw() {
  // Generate random position within the canvas
  let x = random(width);
  let y = random(height);

  // Generate random size for the shape
  let size = random(30, 120);

  // Pick a random color with transparency (alpha)
  fill(random(255), random(255), random(255), random(70, 170));

  // Randomly select which shape to draw (ellipse, rect, or triangle)
  let shapeType = floor(random(3));

  if (shapeType === 0) {
    // Draw a circle
    ellipse(x, y, size, size);
  } else if (shapeType === 1) {
    // Draw a square
    rect(x, y, size, size);
  } else {
    // Draw an equilateral triangle
    let h = size * sqrt(3) / 2;
    triangle(
      x, y - h / 2,
      x - size / 2, y + h / 2,
      x + size / 2, y + h / 2
    );
  }
}

/*
  This code continuously draws random shapes (circles, squares, triangles)
  at random positions, with random sizes and semi-transparent colors.
  The background is not redrawn in draw(), so shapes accumulate and overlap,
  resulting in a colorful, abstract composition.
*/

```