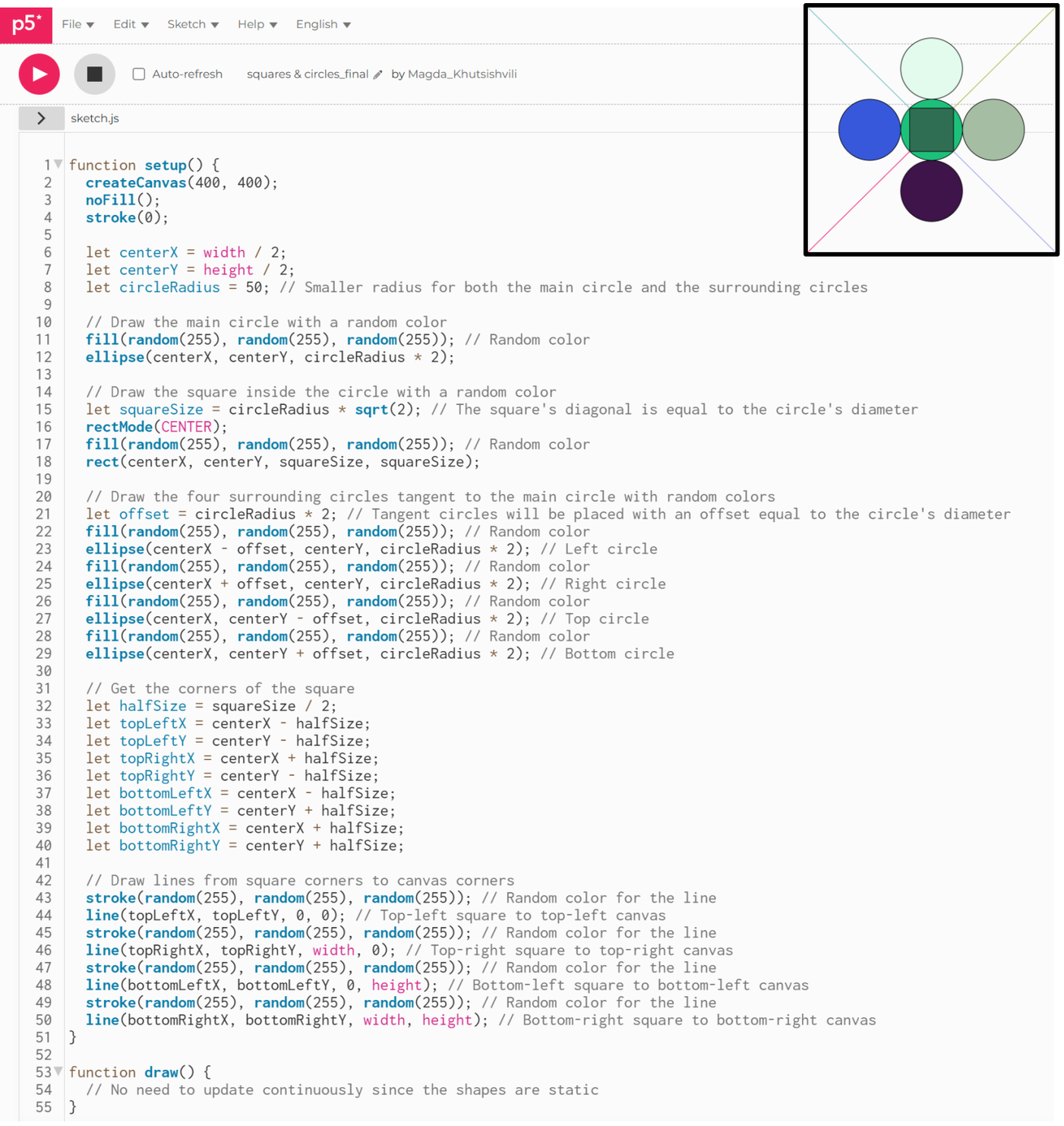
ΠΡΟΒΛΗΜΑ 1




```
function setup() {
  createCanvas(400, 400);
  noFill();
  stroke(0);

  let centerX = width / 2;
  let centerY = height / 2;
  let circleRadius = 50; // Smaller radius for both the main circle and the surrounding circles

  // Draw the main circle with a random color
  fill(random(255), random(255), random(255)); // Random color
  ellipse(centerX, centerY, circleRadius * 2);

  // Draw the square inside the circle with a random color
  let squareSize = circleRadius * sqrt(2); // The square's diagonal is equal to the circle's diameter
  rectMode(CENTER);
  fill(random(255), random(255), random(255)); // Random color
  rect(centerX, centerY, squareSize, squareSize);

  // Draw the four surrounding circles tangent to the main circle with random colors
  let offset = circleRadius * 2; // Tangent circles will be placed with an offset equal to the circle's diameter
  fill(random(255), random(255), random(255)); // Random color
  ellipse(centerX - offset, centerY, circleRadius * 2); // Left circle
  fill(random(255), random(255), random(255)); // Random color
  ellipse(centerX + offset, centerY, circleRadius * 2); // Right circle
  fill(random(255), random(255), random(255)); // Random color
  ellipse(centerX, centerY - offset, circleRadius * 2); // Top circle
  fill(random(255), random(255), random(255)); // Random color
  ellipse(centerX, centerY + offset, circleRadius * 2); // Bottom circle

  // Get the corners of the square
  let halfSize = squareSize / 2;
  let topLeftX = centerX - halfSize;
  let topLeftY = centerY - halfSize;
  let topRightX = centerX + halfSize;
  let topRightY = centerY - halfSize;
  let bottomLeftX = centerX - halfSize;
  let bottomLeftY = centerY + halfSize;
  let bottomRightX = centerX + halfSize;
  let bottomRightY = centerY + halfSize;

  // Draw lines from square corners to canvas corners
  stroke(random(255), random(255), random(255)); // Random color for the line
  line(topLeftX, topLeftY, 0, 0); // Top-left square to top-left canvas
  stroke(random(255), random(255), random(255)); // Random color for the line
  line(topRightX, topRightY, width, 0); // Top-right square to top-right canvas
  stroke(random(255), random(255), random(255)); // Random color for the line
  line(bottomLeftX, bottomLeftY, 0, height); // Bottom-left square to bottom-left canvas
  stroke(random(255), random(255), random(255)); // Random color for the line
  line(bottomRightX, bottomRightY, width, height); // Bottom-right square to bottom-right canvas
}

function draw() {
  // No need to update continuously since the shapes are static
}
```

Link:https://editor.p5js.org/Magda_Khutsishvili/sketches/zsO127sDn

START

1. Set up canvas with size 400x400

2. Define the center coordinates of the canvas (centerX, centerY)
3. Define the radius for the main circle (circleRadius)

4. Draw the main circle with the defined radius at (centerX, centerY)

5. Calculate the square's size:
   - Square's diagonal equals the circle's diameter (circleRadius * sqrt(2))

6. Draw the square at the center with the calculated size

7. Define the offset for surrounding circles (equal to the circle's diameter)

8. Draw 4 surrounding circles at the following positions:
   - Left: (centerX - offset, centerY)
   - Right: (centerX + offset, centerY)
   - Top: (centerX, centerY - offset)
   - Bottom: (centerX, centerY + offset)

9. Get the corners of the square:
   - Top-left corner: (topLeftX, topLeftY)
   - Top-right corner: (topRightX, topRightY)
   - Bottom-left corner: (bottomLeftX, bottomLeftY)
   - Bottom-right corner: (bottomRightX, bottomRightY)

10. Draw lines connecting the corners of the square to the corresponding corners of the canvas:
   - From top-left square to top-left canvas
   - From top-right square to top-right canvas
   - From bottom-left square to bottom-left canvas
   - From bottom-right square to bottom-right canvas

11. Set random colors for the shapes (circle, square, surrounding circles)
   - Use random values for RGB components

12. Set random colors for the lines connecting the square corners to the canvas corners

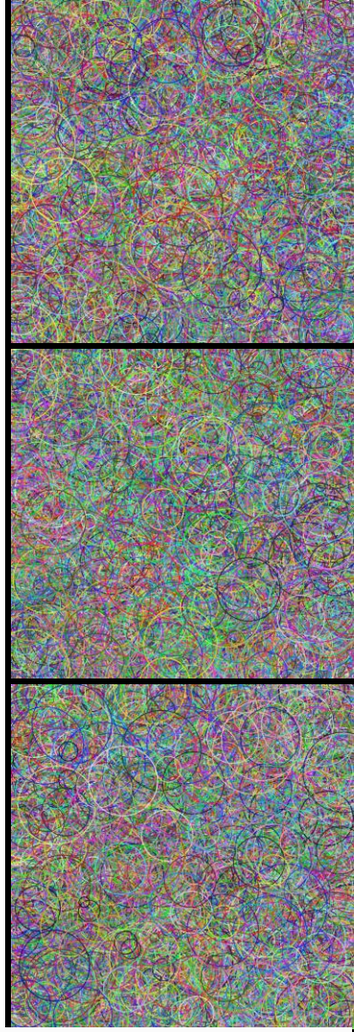
END

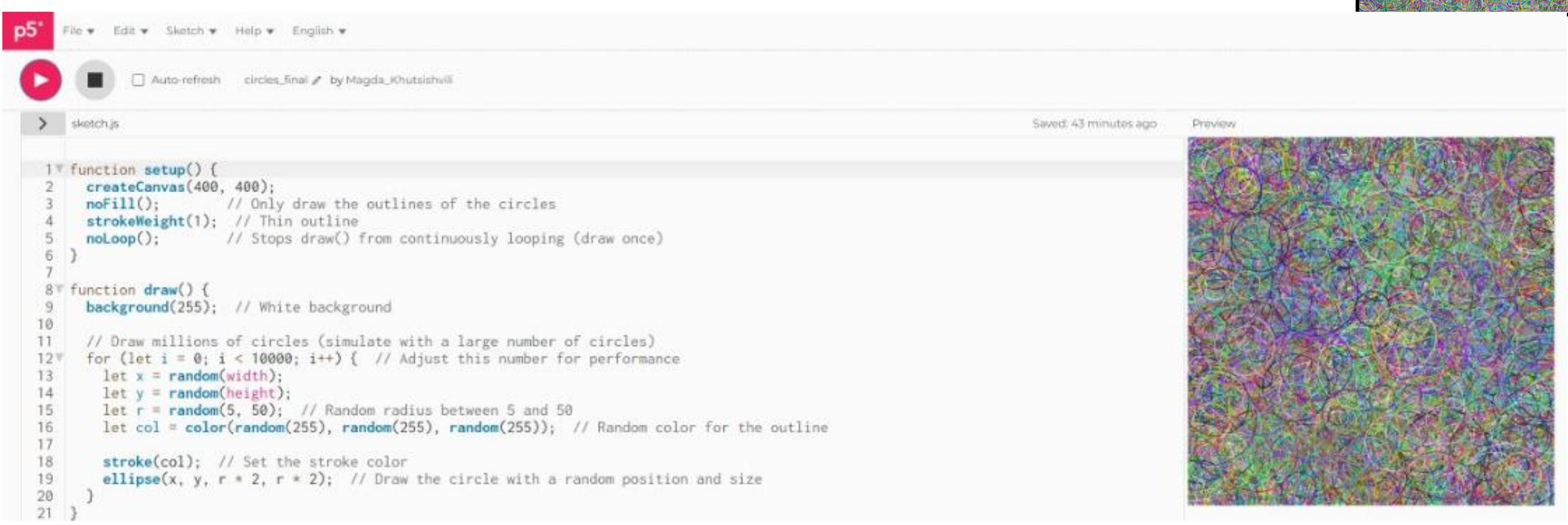
ΠΡΟΒΛΗΜΑ 3

START

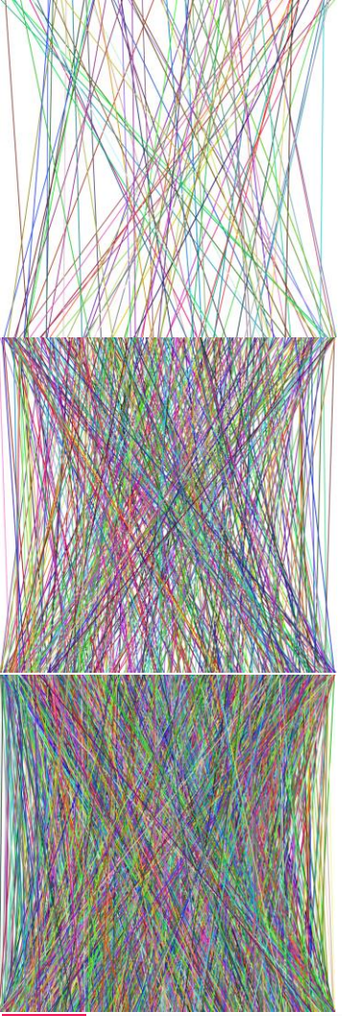
FUNCTION setup()
---CREATE canvas of size 400x400
---SET noFill() // Only outlines for circles
---SET strokeWeight(1) // Thin outline
---CALL noLoop() // Stop continuous looping
FUNCTION draw()
---SET background color to white
---FOR i FROM 0 TO 9999 DO
------SET x = random value between 0 and canvas width
------SET y = random value between 0 and canvas height
------SET r = random value between 5 and 50
------SET col = random color (R, G, B)
------SET stroke color to col
------DRAW circle at (x, y) with diameter (r * 2)

END





Link:https://editor.p5js.org/Magda_Khutsishvili/sketches/69OQik0Kz



BEGIN

---SET canvas size to 400x400
---SET background color to white
---FOR each iteration from 1 to 50 (you can adjust the number of lines)
-------RANDOMLY generate x1 (top x-coordinate)
-------SET y1 to 0 (top y-coordinate)
-------RANDOMLY generate x2 (bottom x-coordinate)
-------SET y2 to canvas height (bottom y-coordinate)
-------RANDOMLY generate color for stroke (red, green, blue)
-------DRAW line from (x1, y1) to (x2, y2) with random color

END

ΠΡΟΒΛΗΜΑ 4

p5* File Edit Sketch Help English

Auto-refresh lines_final by Magda_Khutsishvili

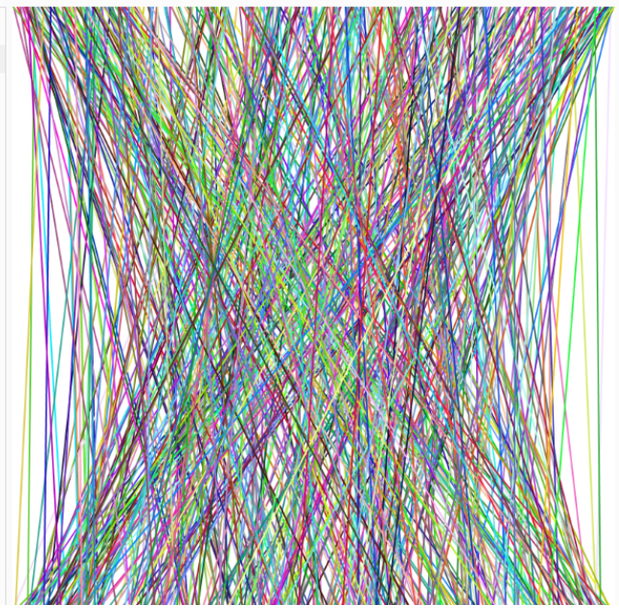
sketch.js Saved: 35 minutes ago Preview

```
function setup() {
  createCanvas(400, 400);
  noLoop();  // Prevents continuous drawing
}

function draw() {
  background(255);  // Set background to white

  // Draw random lines connecting top to bottom
  for (let i = 0; i < 500; i++) {  // You can change 50 to any number for more or fewer lines
    let x1 = random(width);  // Random x position on top
    let y1 = 0;              // Top of the canvas
    let x2 = random(width);  // Random x position on bottom
    let y2 = height;         // Bottom of the canvas

    stroke(random(255), random(255), random(255));  // Random color
    line(x1, y1, x2, y2);  // Draw the line
  }
}
```



Link:https://editor.p5js.org/Magda_Khutsishvili/sketches/DwplHQ9MS

ПРОВЛНМА SOL LEWITT

START

FUNCTION setup()
  CREATE CANVAS (400, 400)

FUNCTION draw()
  SET background color to white (255)

  // Calculate trapezoid coordinates based on mouse position
  SET x1 = mouseX - 50
  SET x2 = mouseX + 50
  SET y1 = mouseY - 50
  SET y2 = mouseY + 50

  // Map mouse position to color values
  SET r = MAP(mouseX, 0, width, 0, 255)
  SET g = MAP(mouseY, 0, height, 0, 255)
  SET b = 150 // Constant blue value

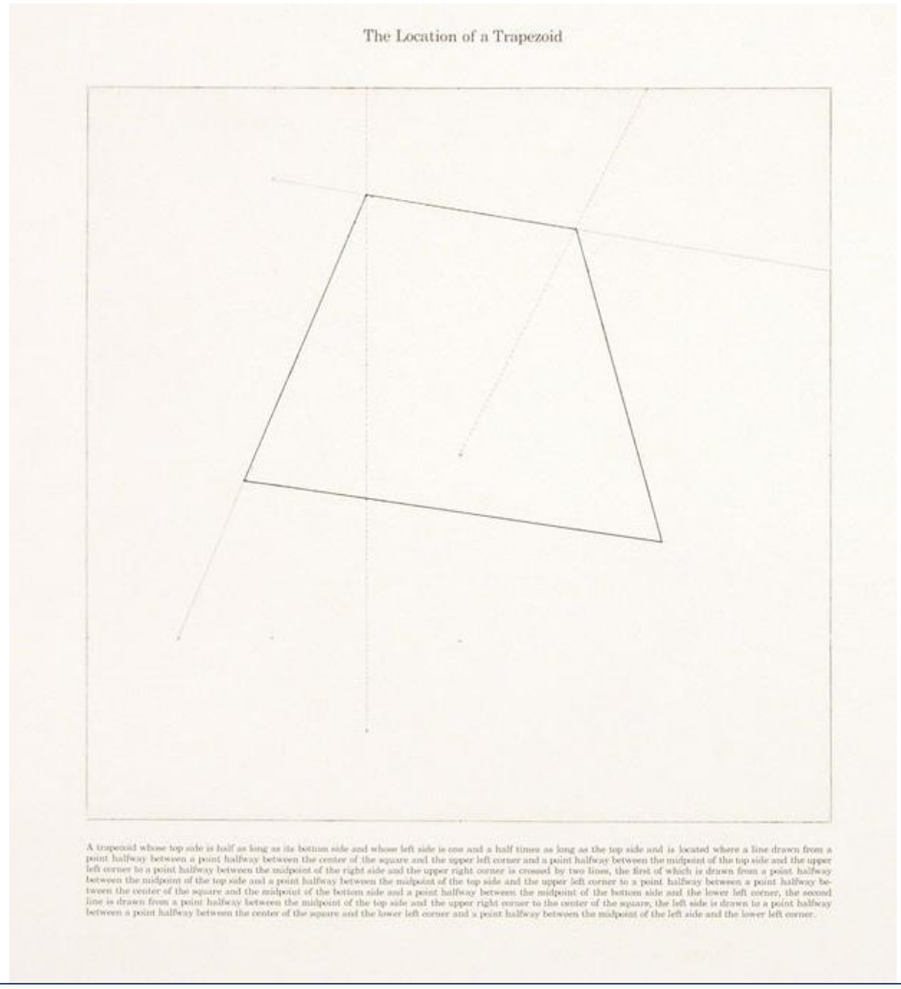
  // Set fill color for trapezoid
  SET fill color to (r, g, b)

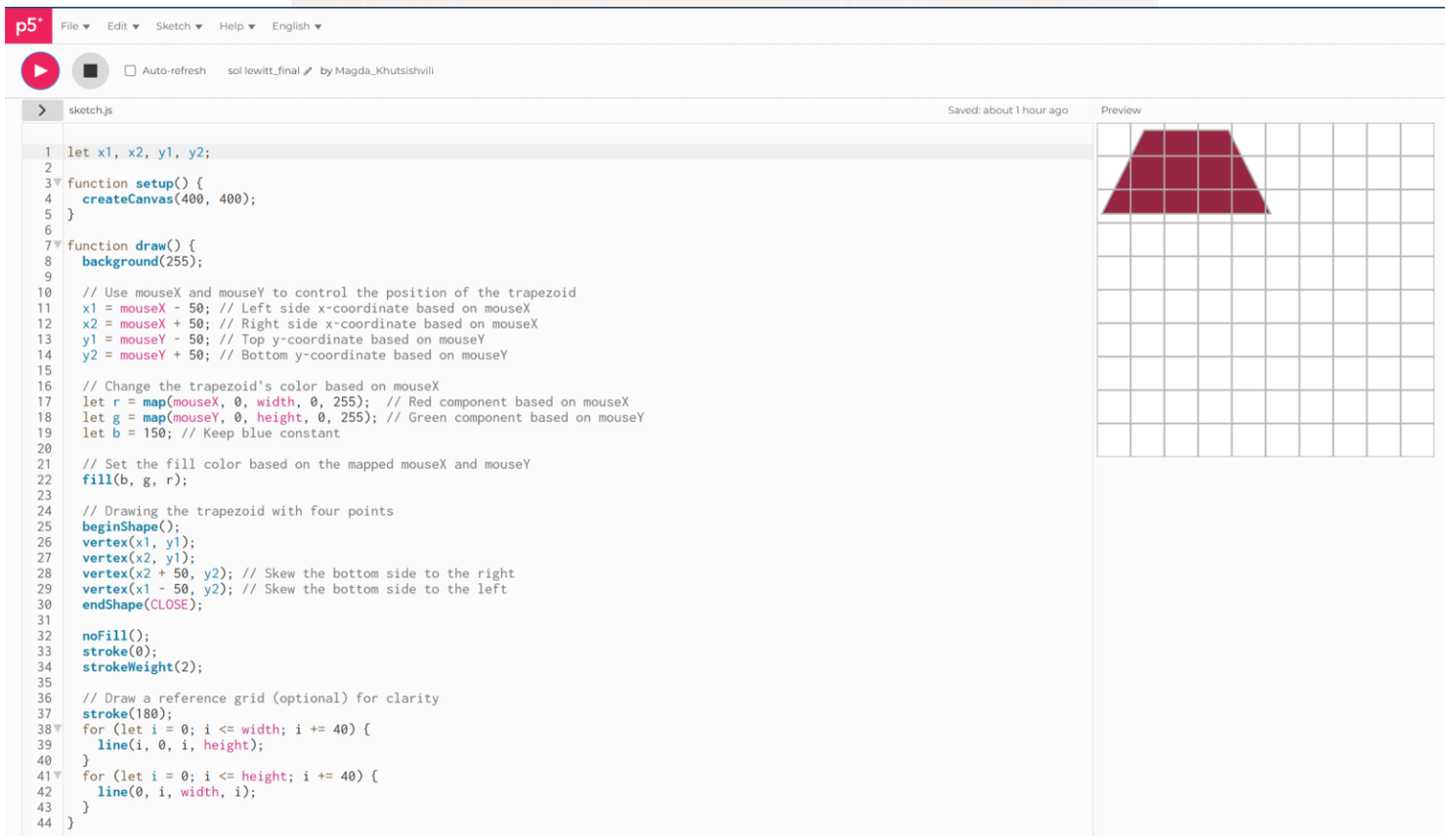
  // Draw trapezoid using four vertices
  BEGIN SHAPE
    VERTEX(x1, y1)
    VERTEX(x2, y1)
    VERTEX(x2 + 50, y2) // Bottom right skew
    VERTEX(x1 - 50, y2) // Bottom left skew
  END SHAPE (CLOSE)

  // Draw grid lines for reference
  SET stroke color to gray (180)
  FOR i FROM 0 TO width STEP 40
    DRAW vertical line at x = i
  END FOR
  FOR i FROM 0 TO height STEP 40
    DRAW horizontal line at y = i
  END FOR

END FUNCTION

LOOP draw() CONTINUOUSLY





Link:https://editor.p5js.org/Magda_Khutsishvili/sketches/8J2dcrymn