

Error Log 1

Successes: Reflection on Project Strengths

Reflecting on Project 1, the strengths or successes of my project are:

- **Flower Patter Generation:** One of the major successes was the creation of flower designs where I created a function to create a circular pattern for the petals and users can adjust the number of petals, the radius of the inner circle as well as the size of the petal when calling the function by changing the appropriate parameters. I think the petals came out really nice and resembles the original inspiration picture really well. Having the adjustable parameters also allowed me to experiment with the size of the flower and see which size and number of petals worked cohesively with the other elements in the tile. The picture below shows the structure of the **drawPetal()** function I created:

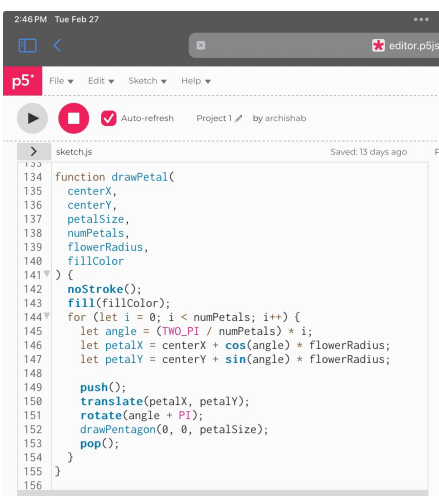


Figure 1: `drawPetal()` function used in my project

- **Attention to Detail:** Adding pentagon shapes to the flowers significantly improved their resemblance to the original tile pattern. This small but impactful change required careful observation and an understanding of geometric shapes. It shows my ability to pay attention to fine details, which enhanced the overall look and feel of the project. It was a bit tricky to figure out the shape and ensure that the flat edge of each of the pentagon shaped flower lined with the center of the flower instead of the points. I had to play around with the rotate value and most of the resources I found had circular petals, hence I had to figure out my own code for the functions to maintain the resemblance with the original picture. The image below shows the resulting the flower shape as well as a the code used to create the hexagon shape:

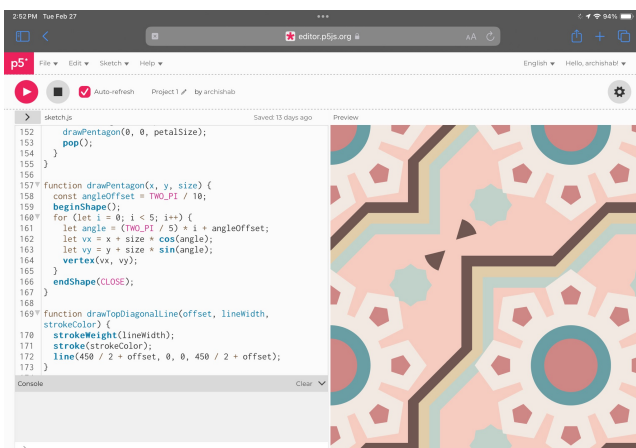


Figure 2: `drawPentagon()` function used in my project and the resulting flower pattern

- **User Interaction:** Including interactive elements, such as allowing users to change the color of lines and providing a reset option without refreshing the page, made the sketch more engaging to the users. It added an element of fun and made the user play around with the sketch more thus retaining their attention longer. It was a learning curve to implement these features, but I think it greatly improved the overall user experience. The image below shows the code snippet I used in my project to add the key and mouse interactions:

```

44 function keyPressed() {
45   loop();
46   for (let i = 3; i < colors.length-1; i++) {
47     colors[i] = color(
48       128 + 128 * sin(frameCount * 0.01 + i),
49       128 + 128 * sin(frameCount * 0.02 + i),
50       128 + 128 * sin(frameCount * 0.03 + i)
51     );
52   }
53 }
54
55 function mousePressed() {
56   colors = [
57     color("#F6CCC0"), // light pink
58     color("#CE8785"), // dark pink
59     color("#F2E9E2"), // antique white
60     color("#E1CDAA"), // mustard
61     color("#6F5651"), // brown
62     color("#C1D2CA"), // light green
63     color("#699EA4"), // turquoise
64   ];
65 }
66

```

Figure 3: keyPressed() and mousePressed() function used in my project

Challenges: Overcoming Obstacles

The following are the major challenges I came across during the project development:

- **Colour Change Feature:** For the keyPressed() feature, my initial plan was for the colours of the diagonals to keep on changing continuously as long as the user presses a key on their keyboard. However, I struggled with the technical aspects and could not fully implement this feature as intended. I tried using a while-loop to implement it the following way:

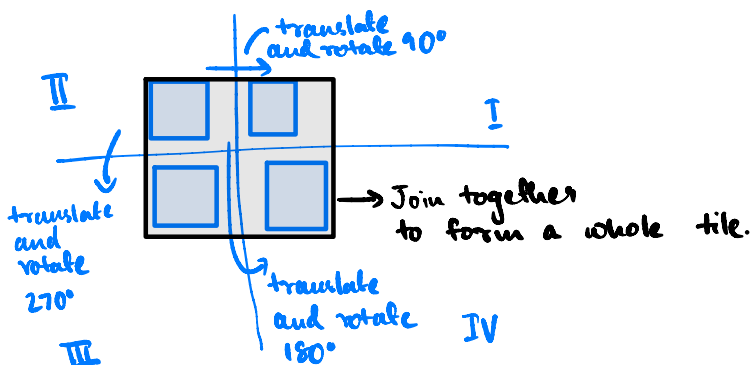
```

while (keyPressed) {
  //function to change colour
}

```

However, this approach gave me the same result as the one I currently have (code snippet in previous section) where the user has to repeatedly press a key to get the colour to change, so I got rid of the while loop.

- **Translate and rotate functions:** Using the translate and rotate function was a bit tricky at first as I could not really wrap my head around how to use them together to create a whole tile. My initial plan was to create a quarter tile in quadrant II and then use the translate and rotate functions to repeat the pattern for quadrants I, IV and III to complete the whole tile



However, it became too complicated for me to manage and instead I took the approach of simply creating a whole tile and just translating it for all of the rows and columns to create a repeating pattern. The code snippet below shows that i created the pattern for a singular tile in a function and then repeated it by calling the function in a nested for loop for rows and columns in the draw function:

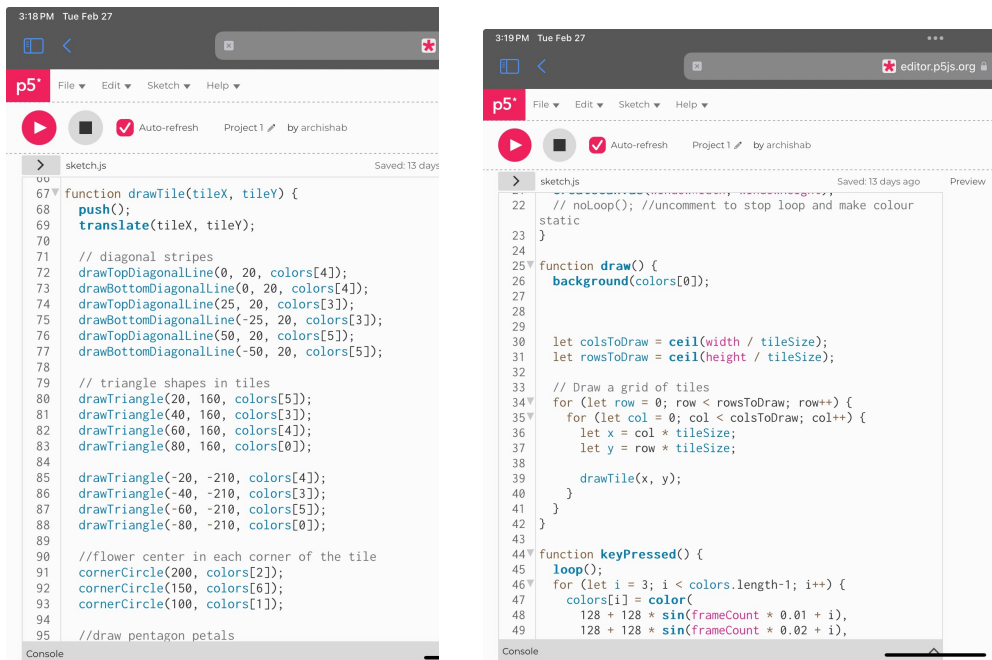


Figure 3: draw tile() function used in my project and nested for loops in draw() function to repeat the tile pattern

Growth: Future Scope

A key learning was exploring user interaction for this project by using the keyPressed() and mousePressed() functions. I enjoyed adding elements that engage users and plan to continue this in future projects. I am interested in interactive and engaging UI/UX design for web development and I think p5.js has a lot of tools and features that can be potentially used to create an interesting and appealing front-end designs. Hence I plan to continue to explore the various tools of p5.js and potentially use them to create fun and interactive prototypes.

Grade: Self-Assessment

Given the above reflections, I would assign myself a grade of B. The project was a success in terms of closely following the original inspiration and demonstrating my technical and creative abilities. However, there were areas that could have been improved, such as adding more variety to the flower designs, not being able to implement my original design idea and the continuous colour changing features and adding comments in my code for better readability to viewers who might not be as familiar with p5.js or JavaScript. Recognizing these shortcomings, I see potentials areas for improvement and growth in future.