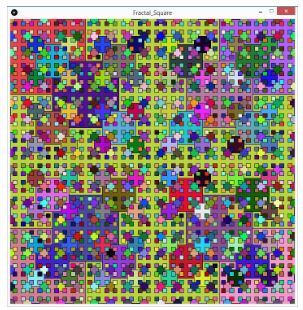
**Recursion Project: Create-a-Fractal**

As a class, we've worked through several fractal examples; images that are created **recursively**by applying a set of **production rules.**

Remember the production rules we created and followed for our square fractal:

1. Draw a square  
2. Determine the location of the four corners  
3. Draw a smaller square at each corner



**Something fun to try before starting your own fractal:**

We made our rectangle fractal a bit interactive, by *mapping the recursion depth to the value of mouseX.*However, the fractal itself was still pretty static. What if we rotated each rectangle each frame? It only takes a bit of code, wrapped around the rect() function call:

* create a new coordinate system - pushMatrix()
* translate by x,y so that the new origin (0,0) is where the rectangle should be drawn
* rotate by **frameCount** degrees  (use the radians() function to convert to radians)
* draw the rectangle (but at 0,0 now since the origin has been moved)
* popMatrix() to discard the current coordinate system transformations.

This would give you something like the video below. Tweak and modify to create something amazing!

<https://youtu.be/77U0mbYHErU>

 This small project is for you to try developing your own fractal image:

1. Begin by constructing a set of *production rules*. This should initially be done without any programming. Try drawing something out on paper.
2. Once you've devised your production rules, translate them into a recursive function.
3. Have fun with color and stroke - try to make the most interesting generative image possible!

Project Requirements

* In your comment header, you must also include your **production rules.**
* Your creative fractal must be generated recursively.

Once complete, ZIP the sketch folder, and submit the compressed file.

Extra for Experts

If we haven't yet, we will shortly go through the basics of working in a 3-D Cartesian coordinate system. As an extra challenge, create a fractal which is built in three dimensions.

