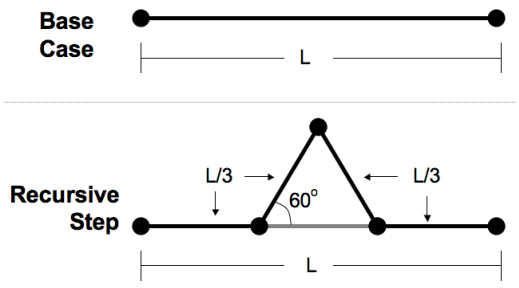
Computer Graphics Project 4 Report (Taekyu Shin at shin7@umbc.edu)

All the model generations are in each Folder(Daisy, Snowflakes, GameWithNFF, Platonic, Seashell)

1. Daisy
    a. Usage > daisy –p <petal_No> ➔ I designed in a way that petal_no maximum is 8. Over 9 just looks kind of weird, not looking like daisy.
    b. Method : I made an affine transformation class. By simply rotation and translation of the basic petal shape. It generates daisy flowers according to it.
        i. Ex > Class Affine af; af.RotateX(90);
        ii. af * daisyFlowerPetalPoints // Affine Transformation occurs.
    c. The important feature is that I apply a same affine transformation to different situation to achieve a natural looking petal, which is naturally tilted and have different lengths in a nicely symmetrical way.
2. Platonic
    a. Usage > platonic <platonic Number>
    b. Method : Depending on the platonic number It generates:

        1: "tetrahedron.nff";
        2: "cube.nff";
        3: "octahedron.nff";
        4: "dodecahedron.nff";
        5: "icosahedron.nff";

    c. Reference: http://www.csee.umbc.edu/~squire/reference/polyhedra.shtml
    d. It uses the same technique as openGL draws a sphere. The basic idea is that we find a vertex corresponding in the surrounding circle.
    e. For your information, you can generate a bigger platonic solid objects by changing the radius. Just change the parameter in the function head. The default is 1.
        i. Radius smaller than 1 does not work in the source code.
3. Snowflakes
    a. Usage > snowflakes <recursive_number>
    b. Reference: answers.oreilly.com/topic/1383-create-a-monster-fractal-snowflake-using-processing/
    c. I use Koch snowflakes. That should explain a lot. Below is the basis and recursive degree.



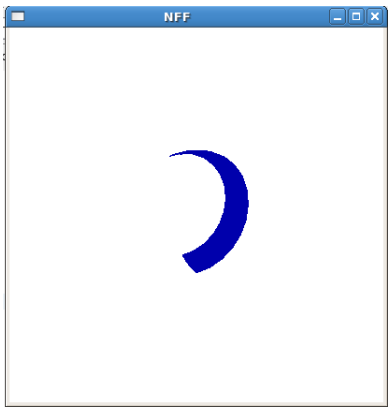
4. Swisscheese
    a. Usage > Swisscheese <number_of_holes_on_each_side_of_swisscheese>

b. Of course, when it collides with other holes, it generates another position to place the hole. If you put more than 13 holes, it won't be able to find space to fill in a hole, which means that it will endless look for space. Therefore, I blocked more than 13 holes.

5. Seashell
    a. Usage > seashell <level_number>
    b. Level number determines the number of quarter spheres drawn to form a seashell.
    c. Reference> http://www.bsu.edu/web/math/exchange/01-01/allen.pdf
    d. Method>
        i. It uses Fibonacci series as the pattern of seashell generation
        ii. I modified the implementation of spheres from my undergraduate class. It can draw any part of spheres. Using that, it can draw something like Figure 1.

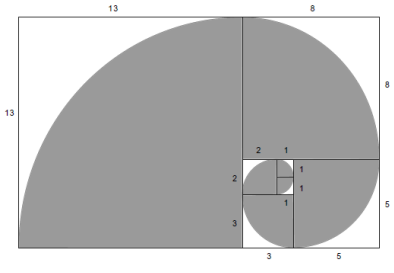
    <Please see the picture below.>

<Below – Figure 1>

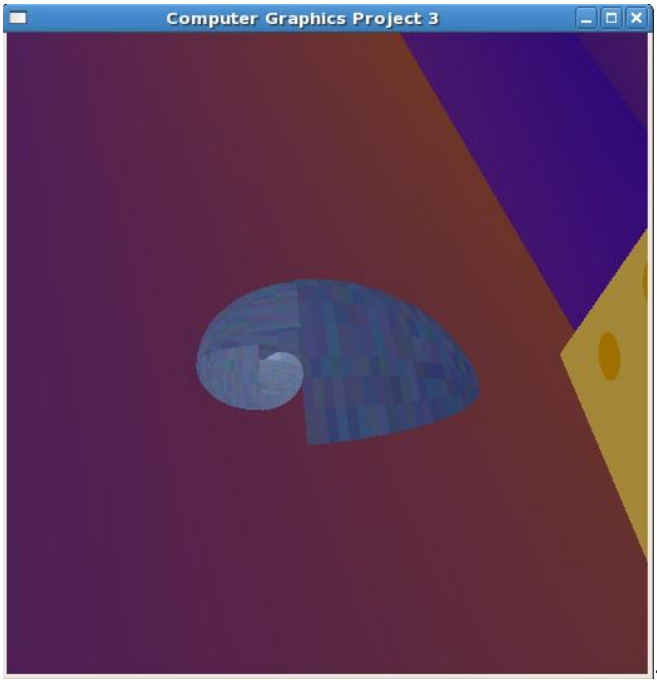


Using many of those, I can draw a little part of the whole seashell.

In a recursive way with Fibonacci series, it can generate all the polygons for it. Please see Figure 2.

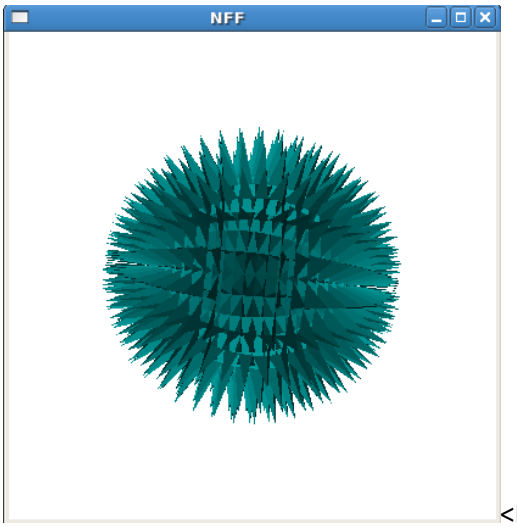


<Figure 2>

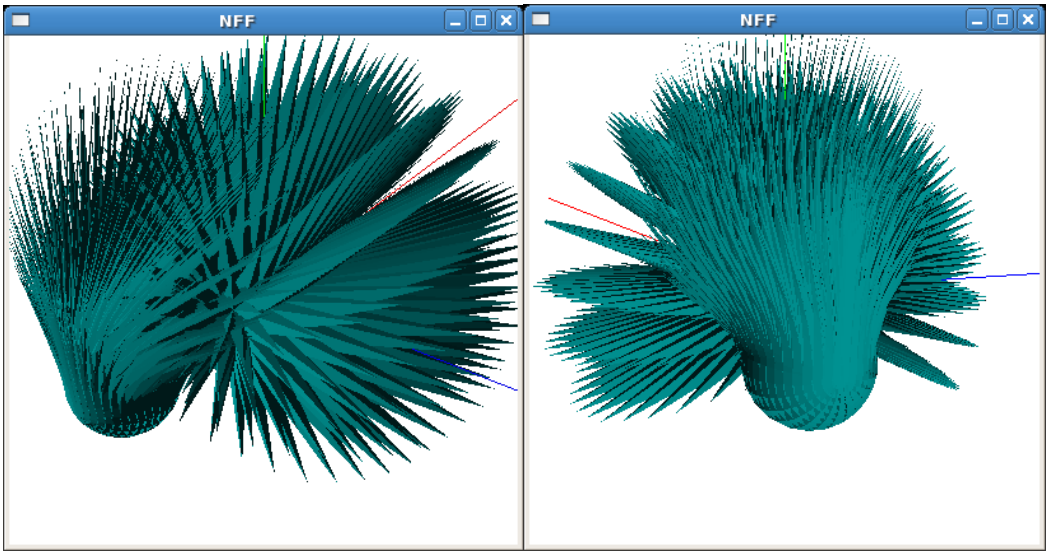


<Generated Model using the technique>

6. Porcupine
    a. <Reference> - None
    b. <Method> - I drew a sphere on my own and modified it. It is all automatic, which means that we can modify the radius, needle length, partial sphere fraction, and etc.
    c. <Information> - I did not get Dr. Rheingans's approval so I do not take credit for it. I just felt really bad because I put a lot of effort into this.
    d. Picture



<Figure – Porcupine Basis>



<Figure – Porcupine. Variable<Radius- 1 , Needle Length – 2.5, Translation 0.3> >

Also, notice that it has non-linear fur.