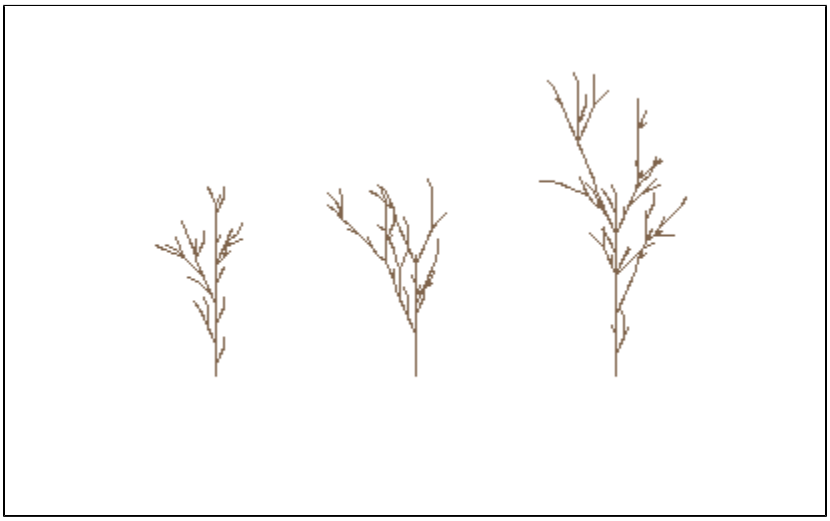
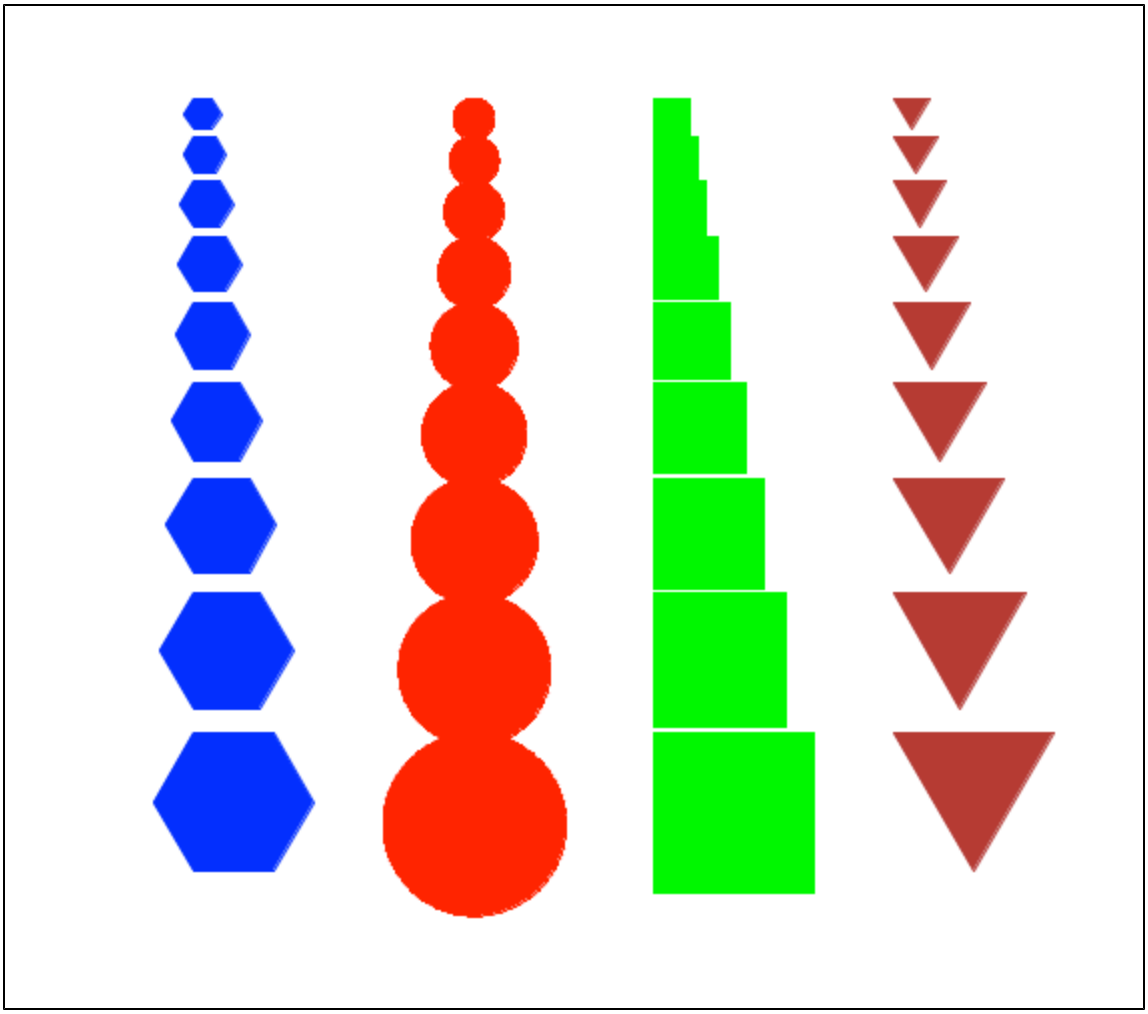
project 9 Netzer

In this project we learned about inheritance. Inheritance is useful because it allows you to define a class and then make new classes similar to that original class with minimal effort. In this project our basic class was one defining a shape, with methods to draw that shape and change its color, and our subsequent child classes made that shape more specific, eg a triangle or square.

We also dealt more with L-systems. We introduced a new feature where multiple rules could occupy one line of an L-system file, and the interpreter chooses one at random to replace the base string. This way we could build trees with natural looking variation. see pic 1:

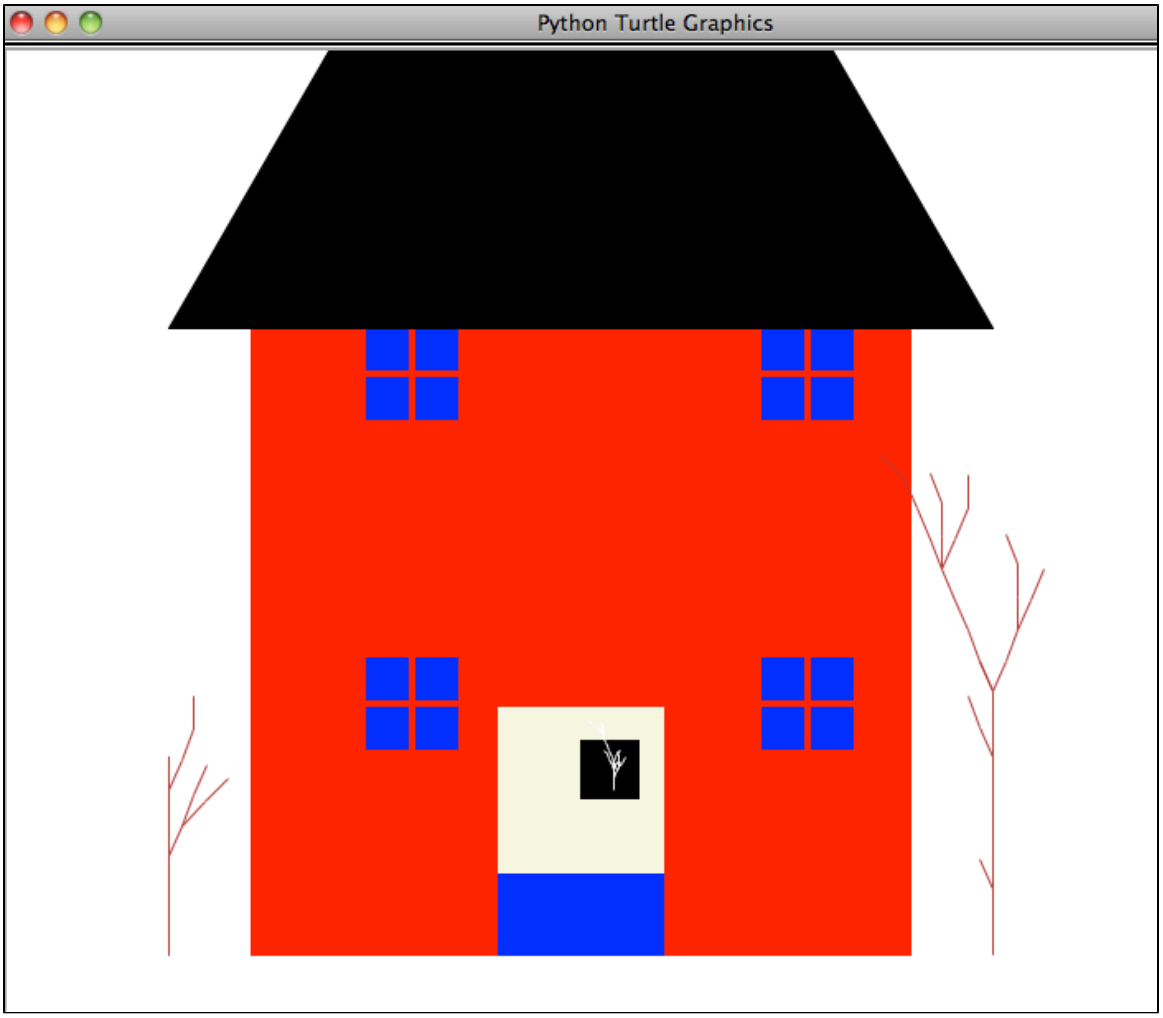


Next I made several child classes off of the basic, "Shape" parent class. I took a pic of my resulting shapes:

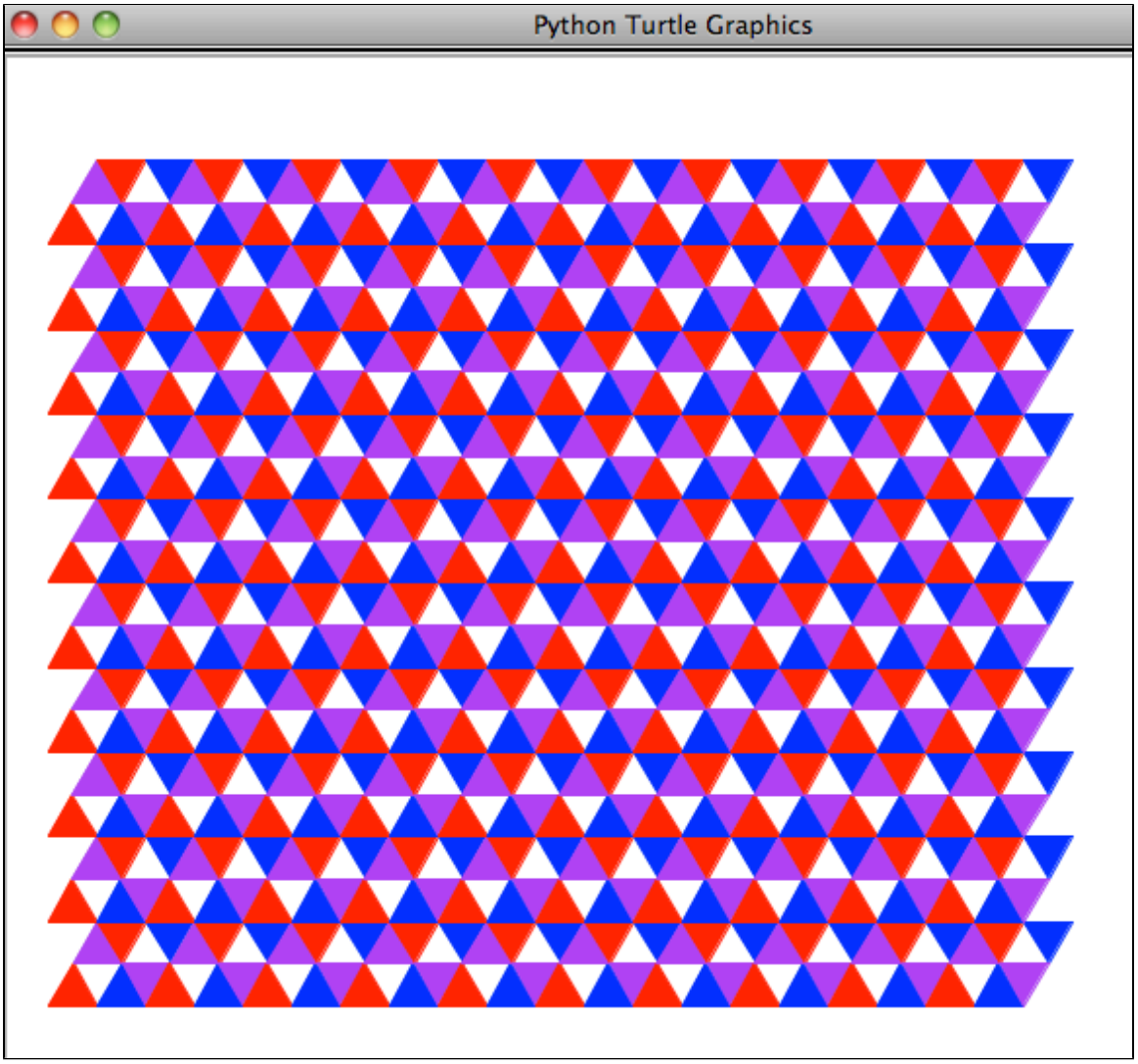


The circle is made with an l-string with 360 "F-", so its really a 360-gon. But its a circle for all purposes here.

Next I drew a house using a combination of my shape objects and draw and setColor methods:



Next I made a mosaic by putting several triangle objects into a function and then iterating that function several times. I made my tile shape a triangle, so I had to include a variable in my tile function to keep track of whether the tile is facing up or down. Using for loops I could draw a tiling of any size across the computer screen.



In this project I learned about objects and how they can be used to simplify code structures. I only had to define one method to draw a scaled, oriented shape that I could use for any shape object I wanted because every new shape class inherits that method. I also learned how randomization can create more naturalistic images.