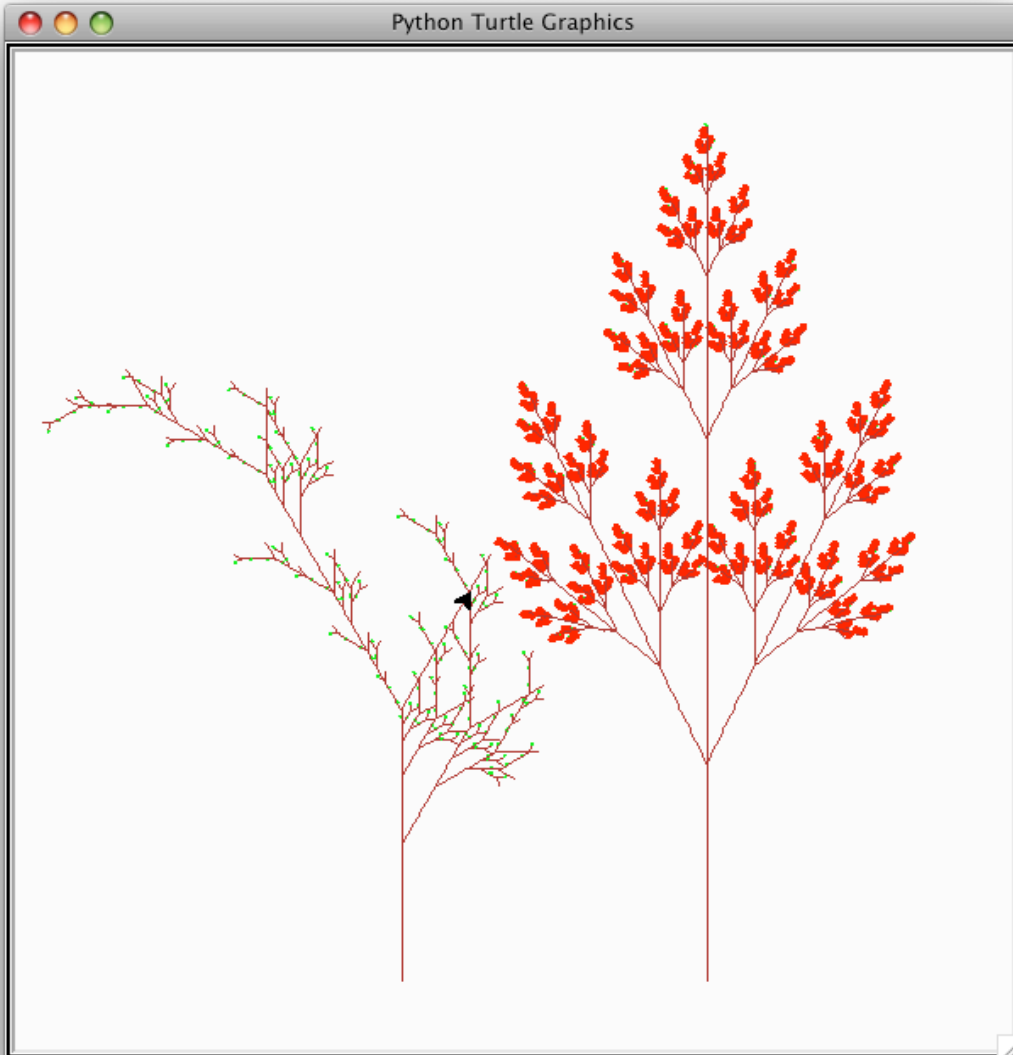


Project 8

The assignment was to use L-systems to create interesting trees. The trees become more realistic as I added leaves and berries to the drawString file.

Task one was to use 2 L-systems to make 2 different styles of trees. The trees used multirule L-systems like:

```
base X  
rule X F[+XLB][-XLB]FXL
```



```
rule F FF
```

Task two was to create a forest like scene with 3 different kinds of trees. To make the code more organized I set up 3 different L-systems for each tree.

```
lsys1 = lsystem.Lsystem( 'systemC' )
```

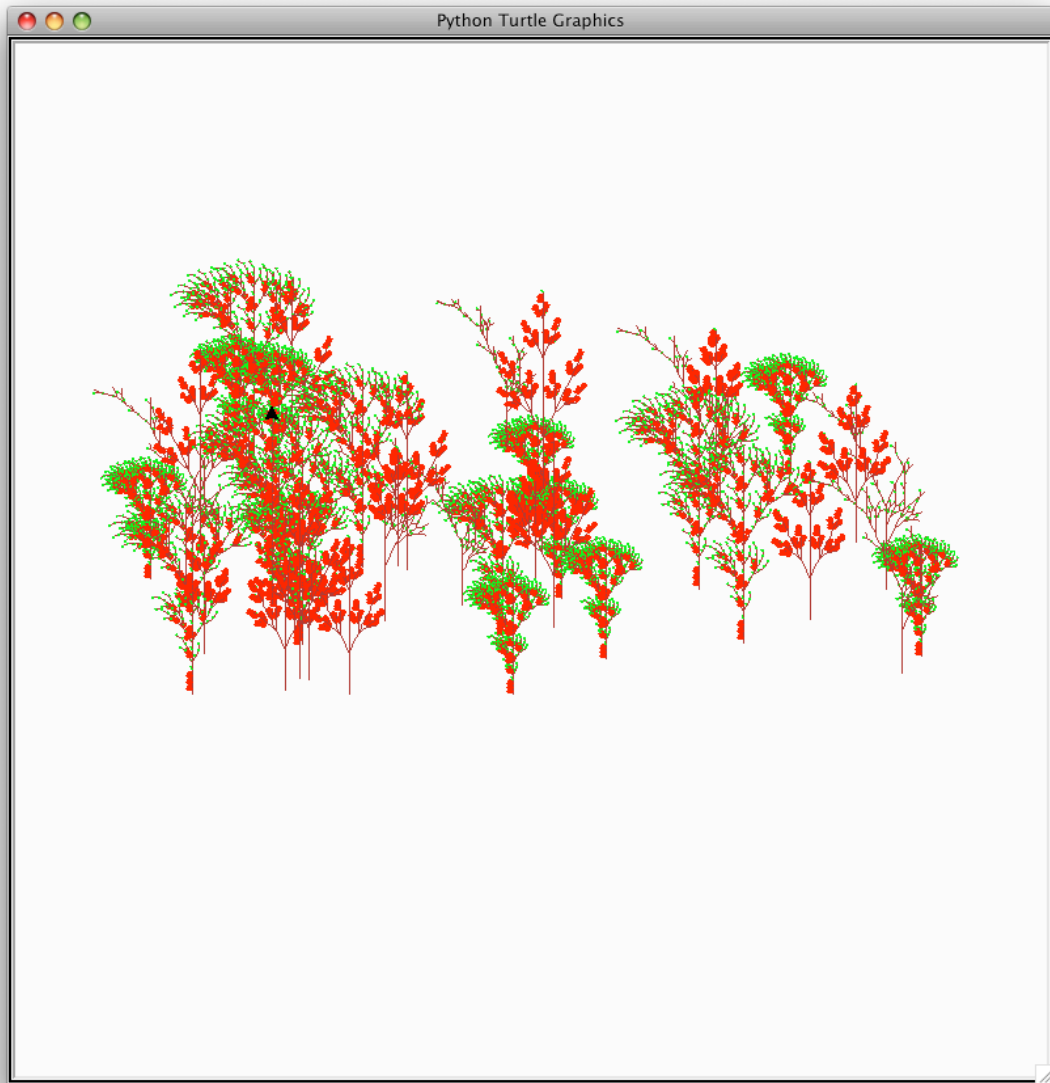
```
lsys2 = lsystem.Lsystem( 'systemD' )
```

```
lsys3 = lsystem.Lsystem( 'systemB' )
```

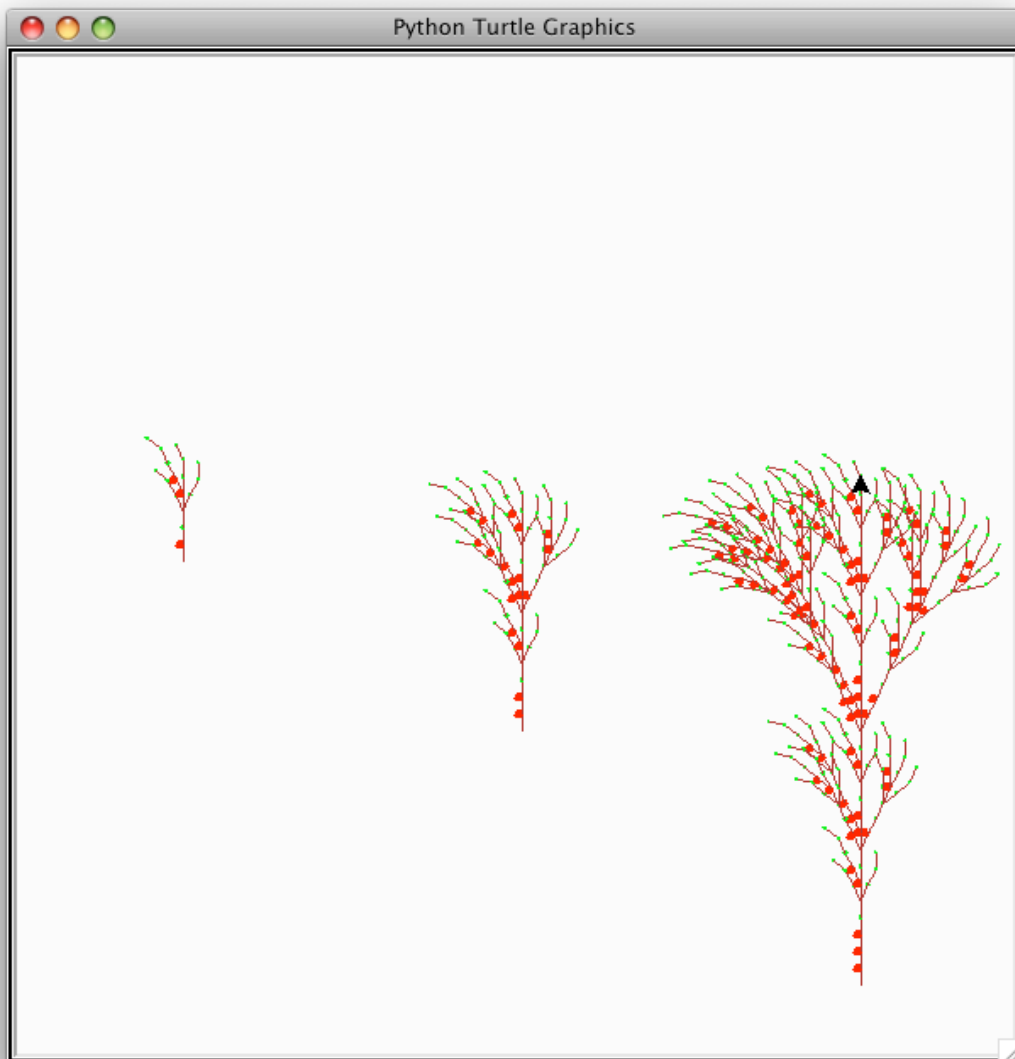
Then I used the random.choice() function to randomize the location, iterations and distances of the trees. Also I used duples to set the tree to a specific angle:

```
trees = [[tree1, 25.7], [tree2, 25], [tree3, 30]]
```

Finally I used a for loop to loop through the trees and place them in a random location. I had to set the x and y boundaries right so that the trees wouldn't go off the page or overlap too much.



Task three was to create my own L-system. I made the original systemB into a multirule L-system and added leaves and berries. Then I set each tree to have different iterations; 3, 4, 5.



The extension I did was to create berries, leaves, and apples for my trees. The code for this comes from the interpreter.py file called by drawString. An example of this code looks like:

```
elif c == 'X':  
    #leaves  
    turtle.color( 'green' )  
    turtle.fill(True)  
    turtle.circle(.5)  
    turtle.fill(False)  
    turtle.color( 'black' )
```

By doing this project I learned how to use to create an L-system, how to make my trees look more realistic by adding elif statements to drawString, and how to organize code.