

Fractal Tree

by George Peck, Lynbrook High School

edited by Geoff Schmit, Naperville North High School

Introduction

Imagine you were describing how to draw a tree. You might say:

1. Draw a vertical line
2. At the top of the line, draw two smaller lines ("branches") in a v shape
3. At the ends of each of those two branches, draw two even smaller branches
4. Keep repeating the process, drawing smaller and smaller branches until the branches are too small to draw

This process of repeating the same design at a continually decreasing scale is an example of a Fractal. Using fractals to draw trees can give some [interesting and beautiful patterns](#). In this assignment we will use a recursive branching function to create a fractal tree.

Procedure

Here is some sample code to get you started with the `Applet` subclass and the `Tree` class:

```
import java.awt.*;
import java.applet.*;

public class FractalTreeExample extends Applet
{
    Tree joyce;
    public void init()
    {
        setBackground(Color.black);
        joyce = new Tree();
    }

    public void paint(Graphics g)
    {
        joyce.draw(g);
    }
}

class Tree
{
    double fractionLength;
    int smallestBranch;
    double branchAngle;
    int startX, startY, endX, endY;
```

```

public Tree()
{
    fractionLength = .8;
    smallestBranch = 10;
    branchAngle = .2;
    startX = 400;
    startY = 700;
    endX = 400;
    endY = 600;
}

public void draw(Graphics graphics)
{
    graphics.setColor(Color.green);
    graphics.drawLine(startX, startY, endX, endY);

    // call recursive branch method...
}
}

```

Notice that the `Applet` subclass is pretty simple. We just create an instance of the `Tree` class and draw it to the screen.

The trunk of the tree needs four values: the X and Y of the starting point and the X and Y of the end point. We will also have three member variables that will control:

- how much smaller the branches are
- how small the branches will get
- the angle between the branches.

Adjusting these parameters will change the appearance of the tree.

Now add a `branch` method to the `Tree` class. The `branch` method will first draw two smaller branches off the end of the tree. It will then call itself recursively to draw two smaller branches off the ends of the previous branches.

Extensions

- add `Scrollbar` objects to adjust the parameters that affect the appearance of the tree
- modify the algorithm to:
 - add asymmetry
 - adjust angles
 - adjust thickness
 - adjust color