CSC 210 Program #1

Recursive Trees
Jan 7, 2008

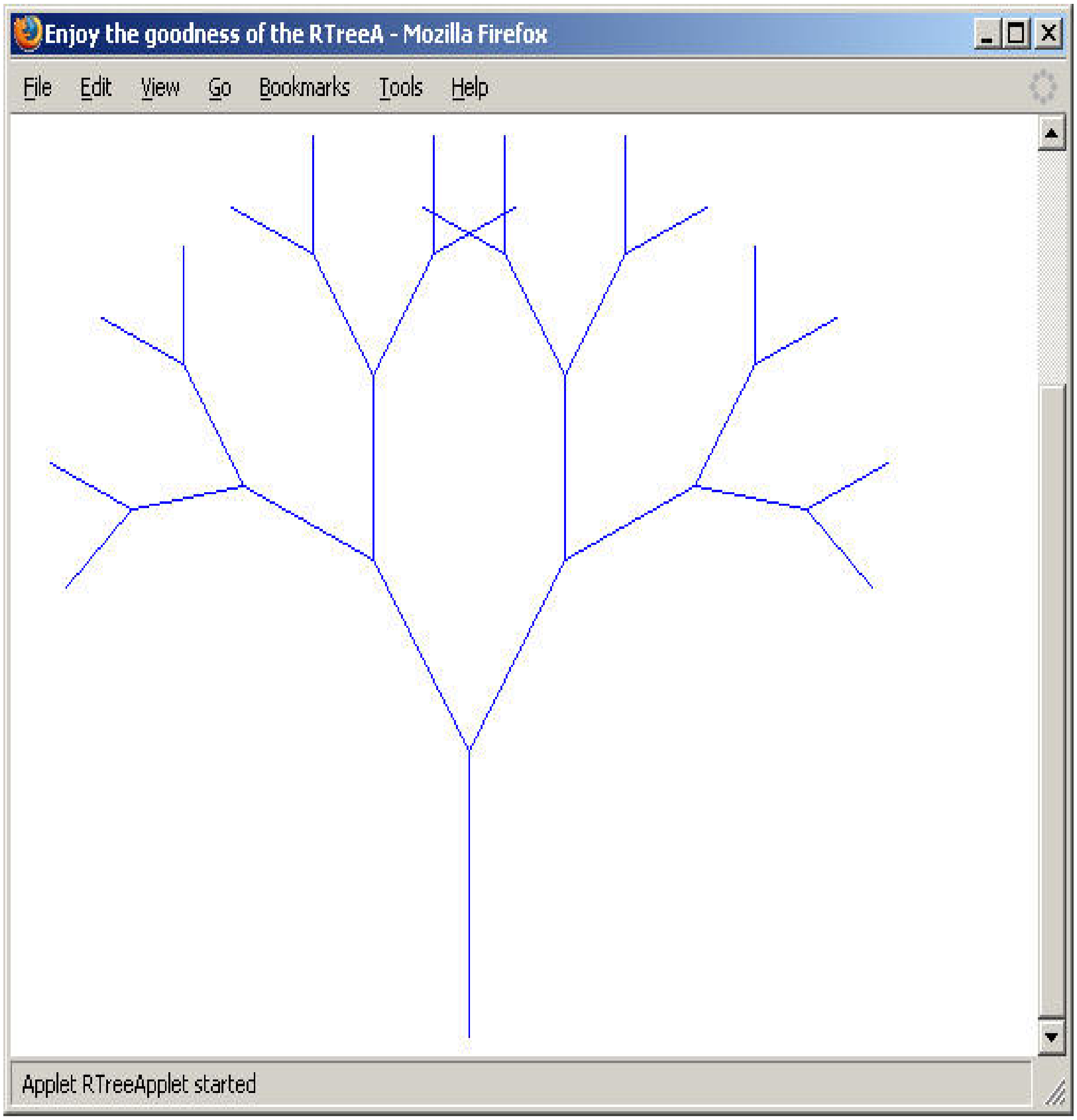
Logistics

Program #1 is:

- **Worth** 5 points, or 5% of your grade
- **Due** Wednesday Jan 16, 2008
- **Covers** recursion and Java GUI programming

Description

In program #1, we'll draw some fun recursive trees, like this one:


Enjoy the goodness of the RTreeA - Mozilla Firefox
File
Edit
View
Go
Bookmarks
Tools
Help
Applet RTreeApplet started

The recursion starts with the trunk. Each branch recursively creates two branches just like itself, but with the following changes:

- Each new branch starts at a new (x, y) coordinate at the end of the previous branch.
- The length of the new branch is changed (usually reduced) by some amount… a multiplicative factor we'll call `lengthFactor`.
- The angle of the new branches is changed by another factor, let's call it `angleDelta`.

The tree example on the previous page has a recursive depth of 5 (with it 5 levels of branches), `lengthFactor` of 0.80 (each successive branch is 80% of the length of the previous level), and `angleDelta` of 33 degrees.

Pseudo-code for the recursive tree algorithm is:

```
// recursively draw tree with depth levels, starting at (x,y)
// length pixels long, with 2 branches at angle
rtree( depth, x, y, length, angle)
  if depth is 0
    then return
  Else
    dx = length * sin( angle)
    dy = length * cos( angle)
    draw line from (x,y) to (x+dx,y-dy)

    // for the left branch
    rtree( depth-1, x+dx, y-dy, length*factor, angle+angleDelta)
    // for the right branch
    rtree( depth-1, x+dx, y-dy, length*factor, angle-angleDelta)
```

Well, how the heck are we going to do all this? There's a lot going on here. We'll talk about this in lecture, and I have some examples here to help you get started:

- You'll want to create a `JPanel` to do the fun things you want to do. I have created an example for you to study: `BlueScreenPanel.java`
- As a final step (probably), you'll want to run your program as an applet. Here's my example: `BlueScreenApplet.java`

An essential source for you will be the Javadoc pages for the Java API:

http://java.sun.com/j2se/1.5.0/docs/api/index.html

For example, you'll do most of your actual drawing in your `JPanel` using the `Graphics` class in Java. Its methods are described here:
http://java.sun.com/j2se/1.5.0/docs/api/java/awt/Graphics.html

I also ask that you add your own creative flair to your program. You can change colors of branches, width of branches, add berries to the end of your branches, add controls to your GUI to change your tree interactively, whatever… something that makes your trees your own. BTW, the `Random` class is a fun way to tweak programs like this.

Grading

By the due date, please place your work for Program #1 in your folder on the k: drive. I'll be looking for:

1. **Your README file** – describing the state of your program... what works, what doesn't, what your “flair” is, etc.
2. **Your Net Beans folder** – including your Java source code, class files, etc.
3. **Your Javadoc** – generate using the “Build/Generate Javadoc” menu
4. **Your applet** – create a web page at w:/index.htm with an Applet of one of your favorite trees
5. **Your printout** – Please print one source file… your most important one, so that I have someplace where I can scratch my comments.

A couple analogies for you… art majors complete beautiful art projects. Architecture majors complete beautiful models. English majors complete beautiful prose.

As computer science majors, your code should be beautiful. It should be well-formatted, commented and follow the class coding guidelines. Code that does not meet this metric will be served a harsh brand of grading justice.

Finally, on the night of the due date, we’ll show everyone’s recursive tree applets in class on the due date. Woot!

good luck… yow, bill

Additional Notes

I have placed my code in the `common_area` folder on the k: drive. There is a method called `degreesToRadians()` in the `Utilities` class which you’ll want to use with the `sin()` and `cos()` methods in the Java’s `Math` class.