Tree Coding
With Processing
An Introduction
Petra Isenberg
CPSC 599.28, 2008
Assignment 1

Containment Layout of Phyllotactic Pattern
Phyllotactic Pattern Creation

Diagram from: The Algorithmic Beauty of Plants (Prusinkiewicz, Lindenmayer)

Formula from: A Better Way to Construct the Sunflower Head (Vogel, 1978)

Image by: Tillman Steinbrecher
Phyllotactic Pattern Creation

$n = 0, 1, 2, ..., n_{\text{max}}$

$\phi = n \cdot \alpha$

$r = c \cdot \sqrt{n}$
Excursion: Polar Coordinates

- Described as a pair: $(r, \phi)$
- From polar to Cartesian:
  - $x = r \cos(\theta)$
  - $y = r \sin(\theta)$
Phyllotactic Pattern Creation

\[ n = 0, 1, 2, \ldots, n_{\text{max}} \]

\[ r = c \cdot \sqrt{n} \]

\[ \phi = n \cdot \alpha \]

\[ m_{\max} = 2, 1, 0 \]

\[ \binom{ncr}{\alpha} = n \]
Changing $\alpha$

$\alpha = 10.0^\circ \quad \alpha = 39.5^\circ \quad \alpha = 137.5^\circ$

$\phi = n \cdot \alpha$
Changing $n$

$$r = c \cdot \sqrt{n}$$

$$\phi = n \cdot \alpha$$

angular constant: 137.5
Changing c

\[ r = c \cdot \sqrt{n} \]
Arrange nodes in phyllotactic pattern at each level in each branch.
Tree Layouts

- Two major types:
  - Node-Link
  - Containment
Containment Layout

- Children part of the display space of the parent
Examples – Voronoi Treemap

• Original Treemap in your book

[Balzer & Deussen, 2005]
Examples - Sunburst

[Stasko & Zhang, 2000]
Example – Software Landscapes

[Balzer et al., 2004]
Hands On

• Open Processing
• Create the initial program structure
• Save your sketch

```java
void setup( ) {
    size(660,600);
    noLoop();  //we don’t have to redraw continuously;
}

void draw( ) {
}
```
Adding external libraries

http://innovis.cpsc.ucalgary.ca/Courses/InformationVisualizationDetails

– Material for Coding Trees
– Download:
  • JTreeLib.jar
  • Crimson.jar
  • Tree Data Sets -> extract and remember where to
Adding libraries in Processing

- Add crimson.jar and JTreeLib.jar
Using external libraries

```java
import ca.ucalgary.innovis.*;
import java.io.File;

NAryTree tree;
NAryTreeNode root;

void setup() {
    size(660,600);
    noLoop(); //we don’t have to redraw continuously;
}
void draw() {
} 
```
JTreeLib

NAryTree
- root
NAryTreeNode
- getChildCount()
- getChildAt(index)
- getParent()
- setParent()
- getNodeSize(w,h), getWidth(), getHeight()
- setPosition(), getXPosition(), getYPosition()
- getIndex(child)
void setup( ) {
    size(660,600);
    noLoop();
    File file = new File("your path\smallTreeTest.tree");
    //use a different file separator if !Windows (File.separator)
    tree = NAryTreeLoader.loadTree(file);
}
Coding a Tree Layout

• Today: 1D TreeMap & Icicle Plot

[Diagram of 1D TreeMap and Icicle Plot]
Drawing the first node

```java
void setup() {
    //initial setup code
    tree = NaryTreeLoader.loadTree(file);

    root = (NaryTreeNode) tree.getRoot();
    root.setSize(600, 50);
    root.setPosition(30, 30);
}

void draw() {
    //draw node in here
}
```
void draw(){
    rect((float)node.getXPosition(),
         (float)node.getYPosition(),
         (float)node.getWidth(),
         (float)node.getHeight());
}
What about all the other nodes?

• Think about it!
Solution

- The layout of every node (!root) depends on:
  - The size of its parent
  - The position of its parent
  - Its position among its siblings
Excursion: Tree traversal

• How to visit each node of the tree
  – Exactly once
  – In a systematic way

• Several methods
  – Classified by order in which nodes are visited
  – Most easily described through recursion
Excursion: Preorder Traversal

- Also called Depth-First

Algorithm:

```python
preorder(node)
    print node.value
    (or do something else with the node)
    for(all the node’s children)
        preorder(child)
```

A H G I F E B C D
Excursion: Postorder Traversal

Algorithm:

```python
postorder(node)
    for (all the node’s children)
        preorder(child)

print node.value (or do something else with the node)
```

G F E I H D C B A
Which traversal do we need?

- The layout of every node (!root) depends on:
  - The size of its parent
  - The position of its parent
  - Its position among its siblings
Preorder

- Implement a function we can call recursively
  - Function should be called from draw()
  - Move drawing of node into separate function

```java
void draw() { drawNode(root); }

void drawNode(NAryTreeNode node) {
    calculate node position, size here
    draw node

    for all children of node: drawNode(child)
}
```
Some hints

• NAryTreeNode parent = node.getParent();
• int nrSiblings = parent.getChildCount();
• Do things differently for the root (parent==null);
• int index = parent.getIndex(node); // Find position of node among siblings
• node.setNodeSize(width,height); parent.getWidth(); parent.getHeight();
• node.setPosition(x,y); parent.getXPosition(); parent.getYPosition();

void draw(){drawNode(root);}
void drawNode(NAryTreeNode node){
    calculate node position, size here
draw node

    for all children of node: drawNode(child)
}
}
void drawNode(NAryTreeNode node) {
    int nrChildren = node.getChildCount();
    NAryTreeNode parent = (NAryTreeNode) node.getParent();
    if (parent != null) {
        int nrSiblings = parent.getChildCount();
        float nodeWidth = (float) (parent.getWidth() / nrSiblings);
        node.setNodeSize(nodeWidth, parent.getHeight());
        int index = parent.getIndex(node);
        // uncomment for icicle plot
        // node.setPosition(index * nodeWidth + parent.getXPosition(),
        //                 parent.getHeight() + parent.getYPosition());
        // node.setPosition(index * nodeWidth + parent.getXPosition(),
        //                 parent.getYPosition());
    }
    drawNode(node);
    for (int i = 0; i < nrChildren; i++) {
        drawNode((NAryTreeNode) node.getChildAt(i));
    }
}
Assignment 1

- Figure out design of containment phyllotree
- Apply knowledge about tree traversal, drawing
- Draw using examples