Assignment 1

Containment Layout of Phyllotactic Pattern

Images by: Tobias Isenberg, Tillman Steinbrecher
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

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

\[
\alpha \cdot \phi = n_{\text{nn}} \Rightarrow \n = 0, 1, 2, \ldots, n_{\text{max}}
\]

\[
\phi = n \cdot \alpha \quad r = c \cdot \sqrt{n}
\]
Excursion: Polar Coordinates

• Described as a pair: \((r, \phi)\)

• From polar to Cartesian

\[
x = r \cos(\theta)
\]
\[
y = r \sin(\theta)
\]
Phyllotactic Pattern Creation

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

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

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

2 \times 137.5^\circ

2, 1, 0
Changing $\alpha$

$\phi = n \cdot \alpha$

$\alpha = 10.0^\circ, \ 39.5^\circ, \ 137.5^\circ$
Changing $n$

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

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

Angular constant: 137.5
Changing c

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

Arrange nodes in phyllotactic pattern at each level in each branch
Possible Layouts
Tree Layouts

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

```java
void setup(){
    size(660,600);

    File file = new File("D:\programming\Processing\TutorialIIExample\wimbledon2004_
    tree = NaryTreeLoader.loadTree(file);
    root = (NaryTreeNode) tree.getRoot();
    root.setSize(600,50);
    root.setPosition(30,30);

```
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()
    - setNodeSize(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
void setup( ) {
    [..]
    tree = NAryTreeLoader.loadTree(file);

    root = (NAryTreeNode) tree.getRoot();
    root.setNodeSize(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)
```
Excursion: Postorder Traversal

Algorithm:

```
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

```c
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.setSize(width,height); parent.getWidth(); parent.getHeight();`
- `node.setPosition(x,y); parent.getXPosition(); parent.getYPosition();`

```java
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()) / (float) nrSiblings;
        node.setNodeSize(nodeWidth, parent.getHeight());
        int index = parent.getIndex(node);
        node.setPosition(index * nodeWidth + parent.getXPosition(),
                parent.getYPosition());
        // uncomment for icicle plot
        // node.setPosition(index * nodeWidth + parent.getXPosition(),
        //                 parent.getYPosition());
        // draw the node here: rect(...)
        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