Assignment 2: Animation Transitions

Due: Oct 6 **Mon**, 11:59pm, 2014 (midnight)

Overview

One of the things that make a visualization look polished is to add animation (animated transition) between each user interactions. In this assignment, you will be implementing animated transitions in Processing between different visualization layouts. The goals of this assignment are to: (a) further reinforce event-based programming, in particular with the render loop for starting and stopping user-triggered animations, (b) introduce the general notion of equivalence in seemingly different visualization layouts, and (c) introduce a few new visualization designs, namely ThemeRiver, Nightingale's Rose, and Stacked Barchart.

Basic Requirements

1. **Dataset**

   You will be giving multi-dimensional datasets in csv format. The first row of the dataset is the title of each column. The first column of the dataset is the labels. The assumptions you can make about the dataset are,

   a) No missing data
   b) No dirty data (No abnormal data)
   c) No negative numbers
   d) The numbers in one column are not all the same
   e) The dataset is not too big (i.e. measured in tens, not hundreds of rows)
   f) There is a maximum number of dimensions (and rows). e.g. 100
   g) The minimum number of rows is 2

2. **Visualizations**

   The "basic" visualizations that you are required to implement are: Line graphs, Pie chart and Bar chart. You have implemented Barchart and Line graph. Pie chart is new.

   These three visualizations are bivariate visualizations. Therefore, even though you will be parsing a multi-dimensional (i.e., multi-column) dataset, your Barchart, Line graphs, and Pie Chart will only display the **first two** dimensions of the dataset. **Namely, use the first column as the labels and use the second column as the values.**

3. **Animation Transitions**

   Recall what you did in the first Lab (Barchart and Line Graph), when the user clicks a button, the visualization sudden shifts to another. This sudden change may appear abrupt and confusing, and it could be difficult for the user to tell that these two visualizations depict the same dataset.

In this assignment, you will use animations to transition between these two visualizations and the newly introduced Pie Chart. We expect these animations to connect the visual elements between the source and target visualizations in **a meaningful way**. Although the definition of "meaningful" may be vague and subjective, there are some clear examples of "not-meaningful" transitions. For example, fading out of one visualization and fade in another does not connect the visual elements between the two visualizations and will therefore not be acceptable. Similarly, most slide-transition animations in MS PowerPoint are not meaningful because they do not connect the individual visual elements.

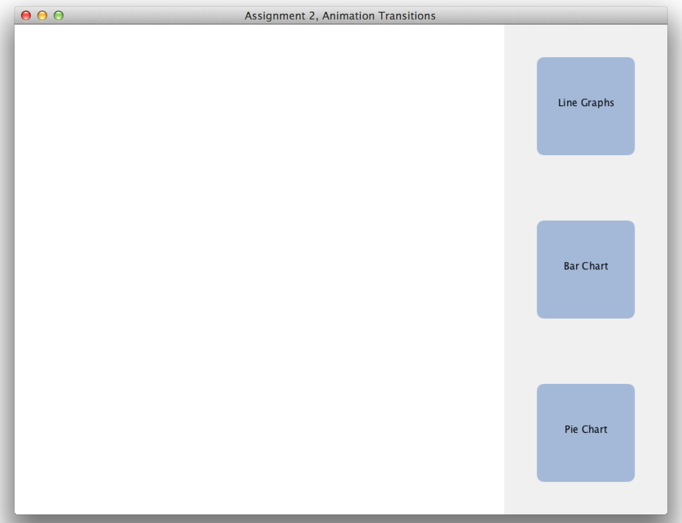
4. **Checklist of basic requirements**
   a. Implement pie chart
   b. In order to trigger transition, you should have three buttons, named as Line graphs, Pie chart and Barchart. Each button responds to one of the three visualizations. When the user clicks a button, the canvas will transit to the corresponding visualization.
   c. Add "direct" animation transitions between these three visualizations. Specifically, you are required to implement animation transitions for:
      Bar -> Pie, Pie -> Bar
      Line -> Bar, Bar -> Line
      Line -> Pie, Pie -> Line
   d. We define "direct transition" as
      i. You are not allowed to design a new "mediating" visualization where all the transitions will go through this mediator.
      ii. The transition between two listed visualizations cannot go through the other one. For instance, if you have implemented Line -> Bar and Bar - > Pie, then your Line -> Pie **cannot** go through Bar.
      iii. Direct transition does not always mean that the transition occurs in "one step". Often, the transition will need to go through a series of steps, some of which could be another visualization. For example, it is possible that your Bar -> Line might go from Bar -> Scatterplot -> Line.

5. **Other requirements/ not required**
   a. Both of your visualizations and animations should be resizable.
   b. You don't have to do hover during the transitions or even when they are end.

Notes for Basic Requirements

1. **Interface and interaction**
   Your interface should include three buttons. Clicking on a button transitions from the current visualization to the selected (new) visualization. The image

below is an example of your interface. Note that the specific position/size/shading of buttons in the image below are not required. You can design the interface however you want as long as there are three buttons.



2. **Guidelines for making animation transitions**
   a) The purpose of making animation transitions is to help your audience understand how the different visual mappings connect to each other. So you should consider how the different visual mapping changes. For instance, see the video Bar_Line_bad.mov. In this video, bars are directly transitioned to lines, which visually looks fine. However, the ***semantics*** of this animation is inappropriate. As a bar is meant to denote the magnitude of a data point, transitioning it into a connecting line makes no sense. So a good animation for Line -> Bar needs to take such mappings into account (See Bar_Line_good.mov for a better example).

   b) Following the above rationale, it can be reasoned that making any visual encoding disappear is not a good idea. For example, in a Bar chart, dropping all bars down to zero (i.e., stop mapping the second dimension of the data) and then have the new visual elements emerge from the x-axis is not acceptable.

   c) Your forward transition should consistent with the backward one. For instance, assume Lie -> Bar is the forward transition. Your backward transition Bar –> Line should be consistent with Lie -> Bar. **But this is not required!**

3. **Tips for making animation transitions**

a) There are almost always more than one way to animate the transition between any two visualizations. The examples used above and in the demo are not meant to be strict requirements. Be creative and have fun! Just make sure that the animations are meaningful semantically and visually.
b) Sometimes you may have to encoding information more than once.
c) The function lerp() is your friend. Lerp stands for linear interpolation, look it up in the Processing library. Note that if you want to be fancy, there is research that show that "slow-in-slow-out" is better than lerp.
d) Other functions you MIGHT use, beginShape() and endShape(), curve(), curveVertex(), arc(), beizer(), beizerVertex() etc.

Additional requirements

You can at most get a grade in the B range if you finish all the basic requirements for this assignment. You need to implement features in additional requirements to get a better grade (in the A range).

In the basic requirements, your visualizations are bivariate. In this section, you will implement visualizations that consider additional data dimensions.

The three new visualizations that you implement are: ThemeRiver (which can be seen as the multivariate version of a line chart), the Nightingale's Rose (which can be seen as the multivariate version of a pie chart), and the Stacked Bar chart (which can be seen as the multivariate version of a bar chart). Your job will be to implement the animated transition between the bivariate and the multivariate visualizations.

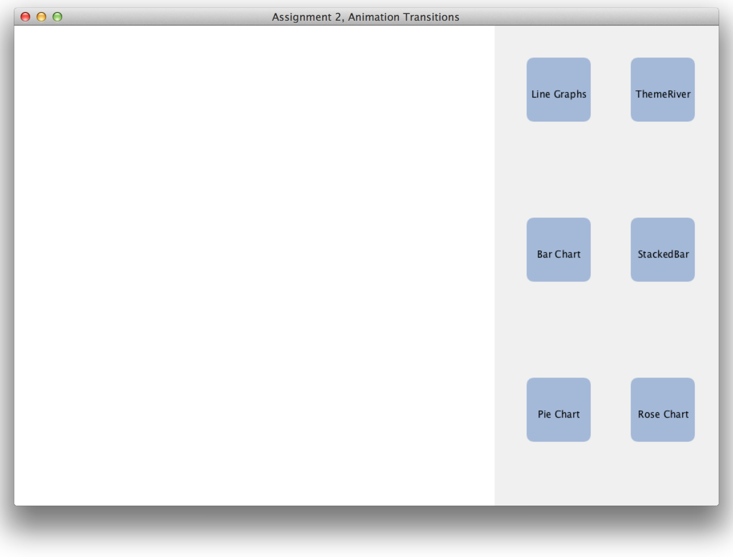
0. Dataset
You are going to use the same dataset. But you are considering all the columns now.

1. Visualizations

a) ThemeRiver was introduced in the paper "*ThemeRiver: Visualizing Thematic Changes in Large Document Collections*" by Susan Havre, Elizabeth Hetzler, Paul Whitney, and Lucy Nowell. For your implementation, you will be making use of splines. For your convenience, use the "curve" function from Processing (or the "Bezier" function, depending on your preference) to make a smooth curve.
b) Nightingale's Rose, or Rose chart (or sometimes referred to as the Coxcomb Diagram) is a multivariate version of a pie chart. Note that unlike the pie chart where every wedge has the same radius but different angles, the rose chart makes use of varying radii to depict difference in magnitude (instead of using angle).
c) Stacked Bar is the multivariate version of a bar chart. As the name suggests, it stacks bars representing different dimensions of data on top of each other.

2. **Animation Transitions**
   Recall the definition of “direct” animation transitions. You need to add animation transitions between each bivariate visualization with its multivariate counterpart (and back). **Note that you do NOT need to implement animated transitions between the different multivariate visualizations**!

3. **Interface and interaction**
   a) Your new interface will have 6 buttons (see image below).
   b) If you click any button, your program should be able to conduct a series of animation to transit to the responding visualization through the “direct” transitions you’ve implemented. For example, if your program is displaying Bar chart, then you click ThemeRiver button, your program should show animation transitions Bar -> Line, Line -> ThemeRiver.
   c) If you are currently displaying a multivariate visualization (e.g. ThemeRiver) and the user clicks on another multivariate visualization (e.g. Rose chart), you do not have to directly transition between these two. Instead, your transition path should be ThemeRiver -> Line -> Pie -> Rose



4. **Checklist of additional requirements**
   a) Implement the three new visualizations: ThemeRiver, Rose Chart, and Stacked Bar Chart.
   b) Implement these “direct” animation transitions,
      Line -> ThemeRiver, ThemeRiver -> Line
      Bar -> Stacked Bar, Stacked Bar -> Bar
      Pie -> Rose, Rose -> Pie
   c) Update your interface

d) Integrate all 6 visualizations and 12 animations transitions together. When you click a button, you should trigger a series of animations transitions to the target visualization.

What is NOT Required:

1. Still, you DON'T need to implement hovering
2. You are free to use colors, but that's not required.

Submission:

1. Submit your work using "provide" by **Mon October 6**, 11:59pm (midnight). The assignment is called "a2":

   ```
   provide comp150viz a2 file1 file2 file3  or
   provide comp150viz a2 * (to submit all files in a directory)
   ```

2. Name your "main" pde file "a2.pde"
3. As part of your submission, provide a file named "team.txt" where you:
   a. Give the name of all the team members
   b. Provide a 1-2 sentence description regarding what each member did in the project.
4. Note that each team only needs to make one submission. We will keep track of the individuals' grades based on the names provided in the team.txt file.
5. Extra credit will be given for creative and novel designs. If you plan on implementing cool features to your visualization, please document them in a separate file called "extracredit.txt"
6. For grading, your team will demo the project to the instructor(s). Tentative date for grading is October 10 (Friday). We will put up a signup sheet later.