Project Integration

CPSC 583 F2011

Tutorial #10 Dec 1, 2011

Jagoda Walny  jkwalny@ucalgary.ca

http://people.ucalgary.ca/~jkwalny
Some inspiration

Alice Thudt & Uta Hinrichs: The Bohemian Bookshelf
What can you do?

Really basic: Put them together in one screen
What can you do?

A little more: brushing, linking, mutual update
Discussion: Be more creative...

- Visualize your vis
  -> How do you place them?
  -> How much space do they take up?
  -> How much do you show in the preview?
Discussion: Be more creative...

- Visualize your vis
  -> How do you place them?
  -> How much space do they take up?
  -> How much do you show in the preview?

- Draw people in
  -> How do you make it intriguing?
  (See: Vis in public spaces lecture)
Discussion: Be more creative...

- Visualize your vis
  -> How do you place them?
  -> How much space do they take up?
  -> How much do you show in the preview?

- Draw people in
  -> How do you make it intriguing?
(See: Vis in public spaces lecture)

- The common thread
  -> Theme of this project is representing information about yourself
Discussion: Be more creative...

- Visualize your vis
  -> How do you place them?
  -> How much space do they take up?
  -> How much do you show in the preview?

- Draw people in
  -> How do you make it intriguing?
  (See: Vis in public spaces lecture)

- The common thread
  -> Theme of this project is representing information about yourself

- Comparison
  -> What is common about your data? Timeframes? Duration? Data type?
  -> You can fake it (e.g. compare one week in Nov to one week in Sept)
Brushing & Linking Exercise

adapted from CPSC 583 F2010 tutorial by Uta Hinrichs

**Brushing**
select a subset of the data to highlight or de-emphasize it

**Linking**
show connections between data in different visualizations
Example: House dataset

<table>
<thead>
<tr>
<th>factor</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>bungalow</td>
<td>2 storey</td>
<td>2 storey</td>
<td>bungalow</td>
<td>2 storey</td>
<td>condomin</td>
</tr>
<tr>
<td>price</td>
<td>490000</td>
<td>550000</td>
<td>520000</td>
<td>1490000</td>
<td>830000</td>
<td>450000</td>
</tr>
<tr>
<td>Squ footage</td>
<td>1400</td>
<td>1700</td>
<td>1200</td>
<td>2100</td>
<td>1900</td>
<td>950</td>
</tr>
<tr>
<td>Distance parks</td>
<td>10</td>
<td>1</td>
<td>1</td>
<td>5</td>
<td>10</td>
<td>5</td>
</tr>
<tr>
<td>Distance downtown</td>
<td>20</td>
<td>20</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>Distance LTR</td>
<td>5</td>
<td>10</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>Distance shops</td>
<td>5</td>
<td>10</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>G</th>
<th>H</th>
<th>I</th>
<th>J</th>
<th>K</th>
<th>L</th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>bungalow</td>
<td>2 storey</td>
<td>2 storey</td>
<td>bungalow</td>
<td>condomin</td>
</tr>
<tr>
<td>price</td>
<td>630000</td>
<td>780000</td>
<td>890000</td>
<td>490000</td>
<td>450000</td>
</tr>
<tr>
<td>Squ footage</td>
<td>990</td>
<td>1900</td>
<td>2500</td>
<td>1000</td>
<td>1100</td>
</tr>
<tr>
<td>Distance parks</td>
<td>2</td>
<td>5</td>
<td>1</td>
<td>1</td>
<td>10</td>
</tr>
<tr>
<td>Distance downtown</td>
<td>20</td>
<td>20</td>
<td>20</td>
<td>5</td>
<td>10</td>
</tr>
<tr>
<td>Distance LTR</td>
<td>20</td>
<td>20</td>
<td>20</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>Distance shops</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>5</td>
<td>1</td>
</tr>
</tbody>
</table>
Housing dataset: Linking & Brushing

vis 1 (left):
- price, sq footage, year built, house type
  - x-axis: year built
  - y-axis: price
  - size: square footage
  - color: type of house

vis 2 (right):
- distance vis
  - left to right:
    - park
downtown
    - LRT
    - shops
Housing dataset: Linking & Brushing

step 1: read in the data

step 2: create two “visualization windows”

step 3: draw vis1

step 4: draw vis2 (distances)

step 5: shapes interactively react to mouse hover
Step 1: Read in the data

ArrayList<Houses> houseList = new ArrayList<Houses>();

holds all houses + their attributes

copy and paste “readData” function into your main class

call “readData” from setup() to read in houses.csv

every house and attributes get stored in Houses object
Step 2: Create Two Vis Windows

create function that draws two simple squares
  • each square will define the space for one of the visualizations

make use of variables to define position & size of squares

EXAMPLE
int w1_xPos = 100;
int w1_yPos = 50;
int w2_xPos = 650;
int w2_yPos = 50;
int windowSize = 500;
Step 2: Create Two Vis Windows

create function that draws two simple squares
• each square will define the space for one of the visualizations

make use of variables to define position & size of squares

EXAMPLE
int w1_xPos = 100;
int w1_yPos = 50;
int w2_xPos = 650;
int w2_yPos = 50;
int windowSize = 500;
Step 2: Create Two Vis Windows

create function that draws two simple squares
  • each square will define the space for one of the visualizations

make use of variables to define position & size of squares

```cpp
int w1_xPos = 100;
int w1_yPos = 50;
int w2_xPos = 650;
int w2_yPos = 50;
int windowSize = 500;

void drawWindows() {
  rectMode(CORNER);
  stroke(0);
  strokeWeight(1);
  rect(w1_xPos, w1_yPos, windowSize, windowSize);
  rect(w2_xPos, w2_yPos, windowSize, windowSize);

  rectMode(CENTER);
}
```
Step 3: Create Vis 1

- x-axis: year built
- y-axis: price
- size: square footage
- color: type of house
- important: create visuals based on Shape class

In Houses class...
ArrayList<Shape> shapeList = new ArrayList<Shape>();

public void initializeHouseShape(int windowPosX, int windowPosY, int windowSize) {
    // create a Shape
    // define & define position, size, and color of this Shape (square) here
    // add Shape to shapeList
}
public void drawCostSizeYearVis() {
    // have Shape drawn here -> shape.drawFunction...
}
Mapping data to space

normalize data to fit into our vis window

define a maxValue
define a minValue

(value - minValue)/(maxValue - minValue) * windowSize

between 0 & 1
Mapping data to space

EXAMPLE: PRIZE & YEAR
float maxPrize = 1490000;
float minPrize = 450000;
float maxYear = 2007;
float minYear = 1911;

float xpos = windowPosX + ((year - minYear)/(maxYear - minYear) * windowSize);

float ypos = windowPosY + windowSize - ((price - minPrize)/(maxPrize - minPrice) * windowSize);
Hints for Vis 1

each Houses object should know about its Shapes

each Shape knows what Houses object it belongs to
-> constructor of Shape class has been changed accordingly

coloring
- each Houses object should have a fixed color (based on its type)
- each Houses object should have a current color that changes depending on mouse interaction
Step 4: Draw Vis 2

draw four vertical axis (one for each distance)
for each house mark the different distances as a circle
add each circle to the shapeList of each house
for each house connect the according circles through lines
color circles and lines according to the house type

Hints:
- distribute data evenly on axis
--->maxDistance, minDistance
Step 5: Interaction on Hover

go through all shapes and check if the mouse pointer “touches” them
- Shape.contains(x,y);

if mouse pointer touches:
- highlight the shape in a special color
- also highlight the according data in the other visualization
- draw strokes in highlighting color
- change thickness of strokes