d3.packSiblings() Tutorial

Michael Nguyen

CMPS 165

Setting the Circle's Styles

In your CSS file, add this code snippet to set your circle's stroke and stroke width. Adjust the width to your liking.

```
circle {
  stroke: #000;
  stroke-width: 1.5px;
}
```

Creating the SVG Element

Add the following code your HTML file.

```
<svg width="960" height="960"><g transform="translate(480,480)"></g></svg>
<script src="//d3js.org/d3.v4.0.0-alpha.35.min.js"></script>
```

Using d3.packSiblings()

This specific D3 Command creates a pack layout which is basically a collection of various-sized circles that represents data. In this example, packSiblings takes in an argument of a circle array. This example creates 2000 circles where each has a random radius size between 8 and 26, and is filtered when the x and y attributes are inclusively between -500 and 500.

```
var svg = d3.select("svg"), // Attaching our previously created SVG Element to the javascript
    width = +svg.attr("width"),
    height = +svg.attr("height"),
    size = Math.max(width, height);

var color = d3.scaleRainbow() // Adding colors to the circles
    .domain([0, 2 * Math.PI]);

var circles = d3.packSiblings(d3.range(2000)
    .map(d3.randomUniform(8, 26))
    .map(function(r) { return {r: r}; }))
  .filter(function(d) { return -500 < d.x && d.x < 500 && -500 < d.y && d.y < 500; });
```

Translating the Circles to the screen

This code snippet will add our pack layout to the SVG, visualizing the circles to the screen.

```
svg
  .select("g")
  .selectAll("circle")
  .data(circles)
  .enter().append("circle")
    .style("fill", function(d) { return color(d.angle = Math.atan2(d.y, d.x)); })
    .attr("cx", function(d) { return Math.cos(d.angle) * (size / Math.SQRT2 + 30); })
    .attr("cy", function(d) { return Math.sin(d.angle) * (size / Math.SQRT2 + 30); })
    .attr("r", function(d) { return d.r - 0.25; })
  .transition()
    .ease(d3.easeCubicOut)
    .delay(function(d) { return Math.sqrt(d.x * d.x + d.y * d.y) * 10; })
    .duration(1000)
    .attr("cx", function(d) { return d.x; })
    .attr("cy", function(d) { return d.y; });
```