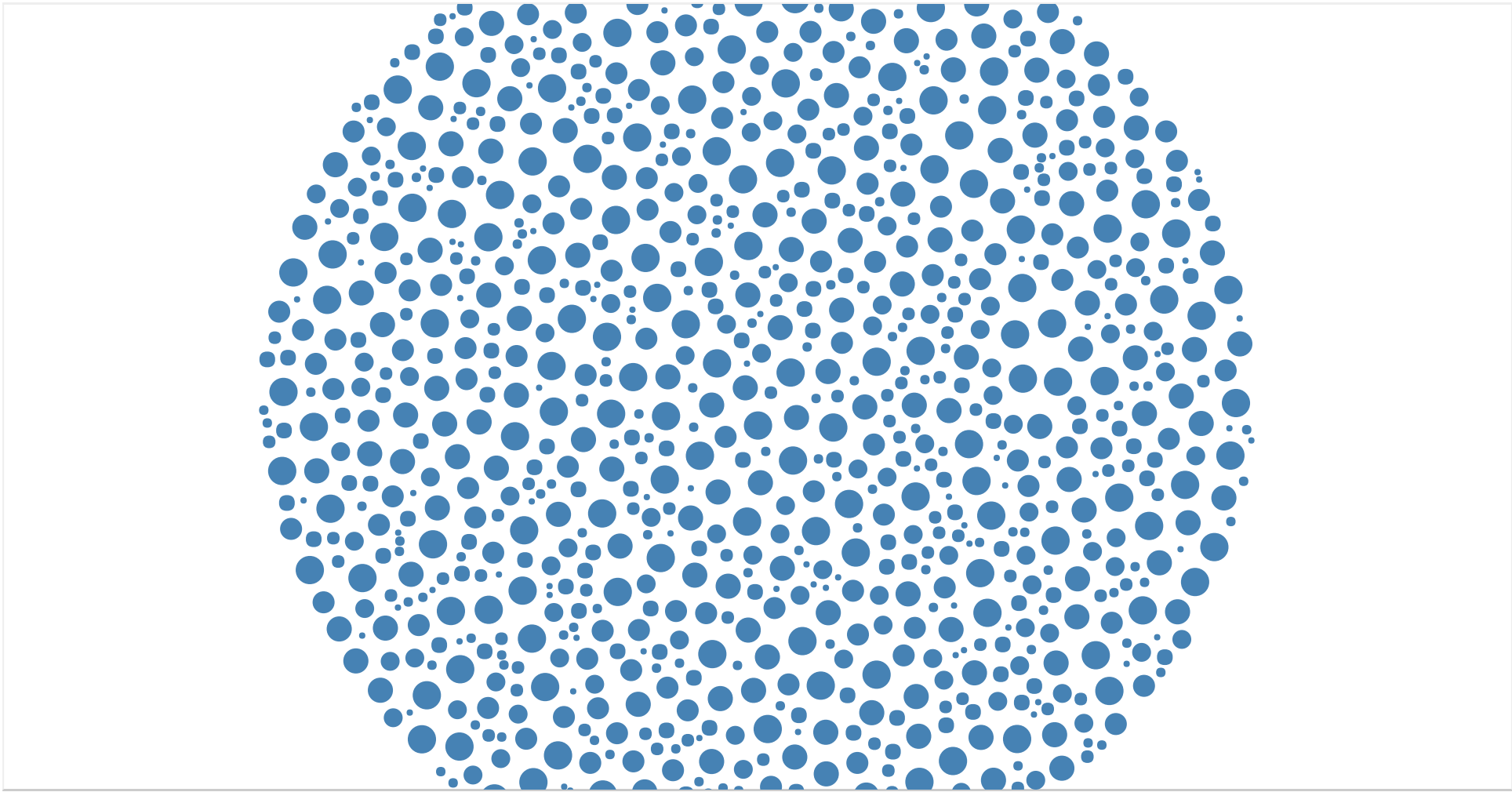
ReactJS + D3 Force



This is an example of using React.js to render a force directed graph. D3.js is used to calculate the position of each circle at each tick, and React is handling actually drawng the circles on the page. Compare performance to the Pure D3 Force example. React is actually significantly slower than just using d3 to manipulate the DOM directly.

Open

Also compare to Canvas + D3 Force.

index.html

```
<!DOCTYPE html>
<html>
<head>
  <script src="http://d3js.org/d3.v3.min.js" charset="utf-8"></script>
  <script src="//cdnjs.cloudflare.com/ajax/libs/react/0.11.0/react.min.js"></script>
</head>
<body>
<div id="container"></div>
<script>
var size = 1000;
var height = 500;
var width = 960;
var charge = -0.3;

var data = d3.range(size).map(function(){
  return {r: Math.floor(Math.random() * 8 + 2)};
});

var start = new Date();
var time = 0;
var ticks = 0;

var force = d3.layout.force()
  .size([width, height])
  .nodes(data)
  .charge(function(d){
    return d.r * d.r * charge;
  })
  .start();


var Chart = React.createClass({displayName: 'Chart',
  render: function() {
    return (
```

```
      React.DOM.svg({width: this.props.width, height: this.props.height}, this.props.children)
    );
  }
});

var DataSeries = React.createClass({displayName: 'DataSeries',
  getDefaultProps: function() {
    return {
      title: '',
      data: []
    }
  },

  render: function() {

    var circles = this.props.data.map(function(point, i) {
      return (
        React.DOM.circle({cx: point.x, cy: point.y, r: point.r, fill: "steelblue"})
      )
    });

    return (
      React.DOM.g(null, circles)
    );
  }
});

var BubbleChart = React.createClass({displayName: 'BubbleChart',
  render: function() {
    var data = this.props.data;
    return (
      Chart({width: this.props.width, height: this.props.height},
        DataSeries({data: data})
      )
    );
  }
});

force.on('tick', function(){
  var renderStart = new Date();
  React.renderComponent(
```

```
    BubbleChart({data: data, height: height, width: width}),
    document.getElementById('container')
  );
  time += (new Date() - renderStart);
  ticks++;
});

force.on('end', function(){
  var totalTime = new Date() - start;
  console.log('Total Time:', totalTime);
  console.log('Render Time:', time);
  console.log('Ticks:', ticks);
  console.log('Average Time:', totalTime / ticks);
});

</script>


</html>
```

LICENSE