D3 + Angular JS = Visual Awesomesauce

John Niedzwiecki

Lead UI Developer - ThreatTrack
@RHGeek on Twitter and GitHub

In addition to turning caffeine into code...
disney geek, runner, gamer, father of two

Live in a World of Data

Everything uses it.

Everyone wants to see it.

But people don’t want to see the numbers.

What do we do? We visualize.

What We'll Do

Look at D3

Look at AngularJS

Create a directive

Look at a component

What is D3?

Data-Driven Documents

“D3.js is a JavaScript library for manipulating documents based on data. D3 helps you bring data to life using HTML, SVG, and CSS. D3’s emphasis on web standards gives you the full capabilities of modern browsers without tying yourself to a proprietary framework, combining powerful visualization components and a data-driven approach to DOM manipulation.”

D3

Best (imho) SVG manipulation library

SVG, Canvas support

Centered on document manipulation driven by data

Created for visualizations, but capable of so much more

What is AngularJS?

“AngularJS is a structural framework for dynamic web apps. It lets you use HTML as your template language and lets you extend HTML's syntax to express your application's components clearly and succinctly. Angular's data binding and dependency injection eliminate much of the code you would otherwise have to write. And it all happens within the browser, making it an ideal partner with any server technology.”

AngularJS

Front-end web application framework

Opinionated framework

Plays nice with other JS libraries

Build reusable pieces through directives (1.x) and components (2.0)

AngularJS + D3

In Angular, we can get our data, bind it to our directive, get more data, set filters, etc

Then, we let D3 do it’s thing.

Angular provides the framing and the piping, D3 provides the power source.

Due to time constraints, concentrate on the D3 code, with an Angular 1.5.8 directive.

Angular

Angular 1.x

- Set it up with a directive or component (special kind of directive, 1.5)

Angular 2

- Set it up with a component

Style based off John Papa Style Guide

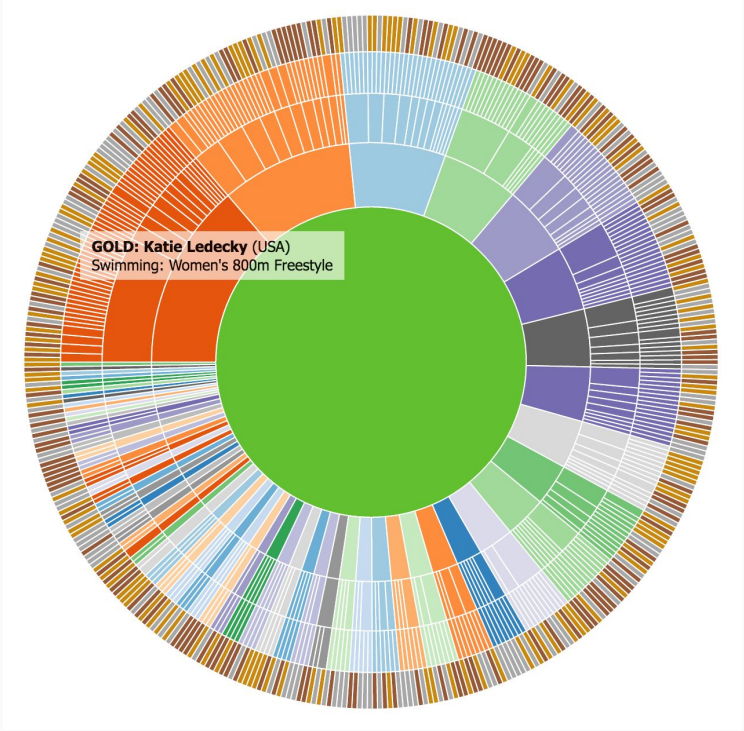
iife, controllerAs, structure...

https://github.com/johnpapa/angular-styleguide

Example

Graph the Olympic medals from Rio 2016

Use a sunburst (based off Mike Bostock's Sunburst)



Angular Directive

```
(function() {
  'use strict';
  angular
    .module('rhgeek.sunburst')
    .directive('sunburst', Sunburst);

  function Sunburst() {
    var directive = {
      restrict: 'E',
      scope: {},
      link: linkFunc,
      controller: SunburstController,
      controllerAs: 'vm',
      bindToController: {}
    };
    return directive;

})();
```

Angular Directive

```
function linkFunc(scope, ele, attr, vm) {

}

SunburstController.$inject = [];

function SunburstController() {
  var vm = this;

}
```

Angular Directive

Bind our list and a loading variable

```
bindToController: {
  loading : '=',
  medals  : '='
}
```

Angular Controller

Simple controller and service

Get medal list from json

Use timer to periodically get day worth of data then update medal list

Page that includes directive

Angular Controller

```
<sunburst medals="om.medals" loading="om.loading"></sunburst>
```

Angular Controller

```
(function() {
  'use strict';
  angular
    .module('rhgeek.olympic-medals')
    .controller('OlympicMedalsController', OlympicMedalsController);

    OlympicMedalsController.$inject = ['OlympicMedalsService', '$timeout'];
    function OlympicMedalsController(OlympicMedalsService, $timeout) {
      var vm = this;
      vm.loading = true;
      vm.medals = [];
      var day = 6;
      var dayTimer;
      getMedals();

    }
})();
```

Angular Controller

```
function getMedals() {
  vm.loading = true;
  return OlympicMedalsService.getDayOfMedals(day)
    .then(function(resp) {
      vm.medals = vm.medals.concat(resp.data);
      if(day<15) {
        dayTimer = $timeout(getMedals, 3000);
      }
      return vm.results;
    }, function(e) {
      console.log('e', e);
    })
    .finally(function() {
      ++day;
      vm.loading = false;
    });
}
```

Angular Directive

Watch loading to know when we have new data
My personal preference to watch instead of watching full list.

`dataToTree` transforms the chronological list to the tree structure required for the visualization.

```
scope.$watch('vm.loading', function() {
  if(!vm.loading) {
    medalTree = dataToTree(vm.medals);
    updateChart(medalTree);
  }
});
```

Angular Directive

Do the bulk of our work in the `linkFunc`
Start by sizing calculations.

```
var width = (attr['width'] ? attr['width'] : Math.floor(ele[0].getBoundingClientRect().width)),
height = (attr['height'] ? attr['height'] : Math.floor(ele[0].getBoundingClientRect().height)),
  padd = 10,
  areaHeight = height - padd * 2,
  areaWidth  = width - padd * 2,
  radius = Math.min(areaWidth, areaHeight) / 2.25;
```

D3 Sunburst

Set up the scales for mapping the data values

```
var xScale = d3.scale.linear()
    .range([0, 2 * Math.PI]);

var yScale = d3.scale.sqrt()
    .range([0, radius]);

var color = d3.scale.category20c();
```

D3 Sunburst

Create our svg element

```
var svg = d3.select(ele[0])
    .append('svg')
    .attr('width', areaWidth)
    .attr('height', areaHeight)
    .append('g')
    .attr('transform', 'translate(' + width / 2 + ',' + height / 2 + ') rotate(-90 0 0)');
```

D3 Sunburst

Partition layout configuration (variant of node-like tree)

```
var partition = d3.layout.partition()
    .value(function(d) {
      return d.size;
    });
```

D3 Sunburst

Arc generation function

```
var arc = d3.svg.arc()
    .startAngle(function(d) {
      return Math.max(0, Math.min(2 * Math.PI, xScale(d.x)));
    })
    .endAngle(function(d) {
      return Math.max(0, Math.min(2 * Math.PI, xScale(d.x + d.dx)));
    })
    .innerRadius(function(d) {
      return Math.max(0, yScale(d.y));
    })
    .outerRadius(function(d) {
      return Math.max(0, yScale(d.y + d.dy));
    });
```

D3 Sunburst

`updateChart` follows a general update pattern
https://bl.ocks.org/mbostock/3808234

```
/* JOIN new data with old elements */

/* ENTER new elements present in new data */

/* UPDATE old elements present in new data */

/* EXIT old elements not present in new data */
```

D3 Sunburst

```
/* JOIN new data with old elements */

var gs = svg.selectAll('g')
    .data(partition.nodes(root));
```

D3 Sunburst

```
/* ENTER new elements present in new data */

var g = gs.enter().append('g')
    .on('click', click)
    .on('mouseover', mouseoverArc)
    .on('mousemove', mousemoveArc)
    .on('mouseout', mouseoutArc);

var path = g.append('path');
```

D3 Sunburst

```
/* UPDATE old elements present in new data */
gs.select('path')
  .style('fill', function(d) {
    return color((d.co ? d.co : 'rio'));
  })
  .transition().duration(500)
  .attr('d', arc)
  .attr('class', function(d) {
    if(d.medal) {
      return 'medal-'+d.medal;
    } else if(!d.co) {
      return 'rio';
    } else {
      return '';
    }
  })
  .each( /*stash*/ );
```

D3 Sunburst

```
/* EXIT old elements not present in new data */

gs.exit()
  .transition()
  .duration(500)
  .style('fill-opacity', 0)
  .remove();
```

In Action

http://localhost:8000/app/index.html

Github: https://github.com/RHGeek/devfest-dc-visual-awesomesauce

D3 Sunburst

Click for zoom tweens the arcs with selected arc as new root

```
function click(d) {
  root = d;

  path.transition()
      .duration(750)
      .attrTween('d', arcTween(d));
  }
```

D3 Sunburst

Tween function to interpolate scale

```
function arcTween(d) {
  var xd = d3.interpolate(xScale.domain(), [d.x, d.x + d.dx]),
      yd = d3.interpolate(yScale.domain(), [d.y, 1]),
      yr = d3.interpolate(yScale.range(), [d.y ? 20 : 0, radius]);
  return function(d, i) {
    return i ? function(t) {
      return arc(d);
    } : function(t) {
      xScale.domain(xd(t));
      yScale.domain(yd(t)).range(yr(t));
      return arc(d);
    };
  };
}
```

D3 Sunburst

Need to handle resizing the svg element

```
Sunburst.$inject = ['$window'];

function Sunburst($window) {

  /* ... */

  angular.element($window).on('resize', resizeSunburst);

  scope.$on('$destroy', function () {
      angular.element($window).off('resize', resizeSunburst);
  });

}
```

D3 Sunburst

Get size and update some scales

```
function resizeSunburst() {
  width = (attr['width'] ? attr['width'] : Math.floor(ele[0].getBoundingClientRect().width));
  height = (attr['height'] ? attr['height'] : Math.floor(ele[0].getBoundingClientRect().height));
  padd   = 10;
  areaHeight = height - padd * 2;
  areaWidth  = width - padd * 2;
  radius = Math.min(areaWidth, areaHeight) / 2.25;
  yScale = d3.scale.sqrt()
      .range([0, radius]);
  d3.select('svg')
    .attr('width', areaWidth)
    .attr('height', areaHeight);
  svg.attr('transform', 'translate(' + width / 2 + ',' + height / 2 + ') rotate(-90 0 0)');
  updateChart(medalTree);
}
```

Angular 2

Angular 2 is written in TypeScript

TypeScript is superset of ES6
- provides typing
- transpiles down to ES5

AngularJS 2.0 is official
- a lot of scaffolding to set up
- still feels like it’s evolving

Angular 2

Create a component, then leverage the same D3 code

```
@Component({
  selector: 'rhgeek-sunburst',
  templateUrl: 'rhgeek-sunburst.component.html',
  styleUrls: ['rhgeek-sunburst.component.css']
})

export class RhgeekSunburst {
  ngOnInit() { //impl }
  ngOnChanges { //impl }}
}
```

Github: https://github.com/RHGeek/devfest-dc-visual-awesomesauce
(I promise I'll finish editing the README.md

Twitter: @rhgeek

LinkedIn: https://www.linkedin.com/in/johnniedzwiecki2

Email: john.niedzwiecki.ii@gmail.com

Blog: http://www.rungeekrundisney.com