goo.gl/DPJ3vG

Victory.js

ReactJS && D3

goo.gl/DPJ3vG

@colinmegill

formidable.com

goo.gl/DPJ3vG

Thanks for having me

@ReactiveConf

goo.gl/DPJ3vG

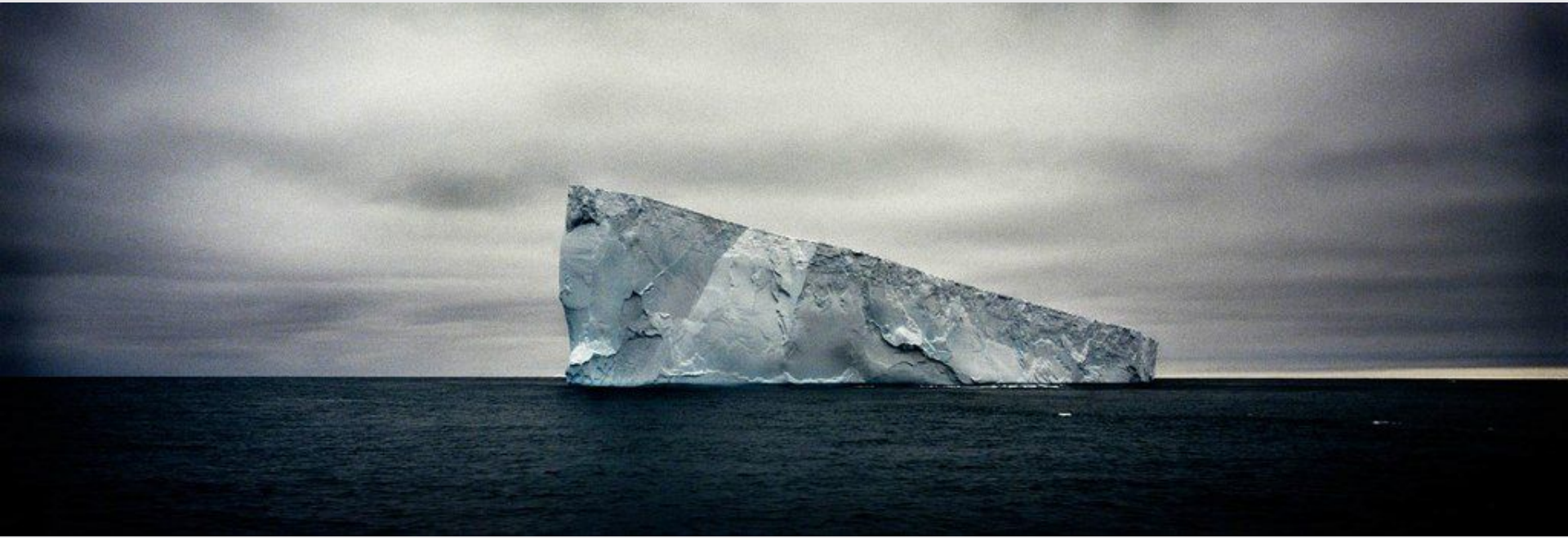
Dedicated to:

@mbostock @worrydream @shancarter

goo.gl/DPJ3vG

a quick ‘what’


Vladimir Todorov



goo.gl/DPJ3vG

A new foundation for interactive data visualization.

- ✓ React DOM model
- ✓ React data binding / lifecycle
- ✓ D3 layout math

goo.gl/DPJ3vG

how

goo.gl/DPJ3vG

Gone:	Replaced with:
`.selectAll()`	`react`
`.select()`	
`.data()`	
`.enter()`	
`.exit()`	
`.update()`	
`.transition()`	`victory-animation`
`.axis()`	`victory-axis`
`.brush()`	`victory-brush`

D3 pioneered interactive data visualization

- extensive api
- extensive ecosystem of gists to start from / learn from
- **DOM model inspired React authors**
- encouraged functional paradigm
- extensive geo functionality*
- well conceived and maintained by Bostock
- **brought academic layout algorithms onto client**

* victory does not yet

goo.gl/DPJ3vG

demos :D

(caveat installer :)

victory-scatter JavaScript ★ 3 ⑂ 0
D3 scatter plot component for React
Updated an hour ago

victory-line JavaScript ★ 4 ⑂ 0
victory line component
Updated an hour ago

victory-bar JavaScript ★ 1 ⑂ 0
Bar chart victory component
Updated an hour ago

victory-label JavaScript ★ 0 ⑂ 0
A label for use with other Victory components
Updated 2 hours ago

victory-util JavaScript ★ 0 ⑂ 0
Victory helpers and utility functions
Updated 2 hours ago

victory-axis JavaScript ★ 2 ⑂ 0
Axis component for victory
Updated 2 hours ago

victory-pie JavaScript ★ 12 ⑂ 2
D3 pie & donut chart component for React
Updated 2 hours ago

goo.gl/DPJ3vG

victory-pie demo

https://github.com/colinmegill/reactive2015

goo.gl/DPJ3vG

victory-pie npm

https://www.npmjs.com/package/victory-pie

goo.gl/DPJ3vG

victory-pie repo

https://github.com/FormidableLabs/victory-pie/

goo.gl/DPJ3vG

victory-pie src

https://github.com/FormidableLabs/victory-pie/blob/master/src/components/victory-pie.jsx

goo.gl/DPJ3vG

victory-pie docs

inventing on principle

media for thinking the unthinkable

@worrydream

https://github.com/FormidableLabs/victory-pie/tree/master/docs

goo.gl/DPJ3vG

victory-pie native

https://github.com/FormidableLabs/victory-pie/blob/feature-native/src/components/victory-pie.jsx

goo.gl/DPJ3vG

composition

https://github.com/colinmegill/reactive2015

goo.gl/DPJ3vG

chart

https://github.com/colinmegill/reactive2015

goo.gl/DPJ3vG

11 reasons why

goo.gl/DPJ3vG

0. Accessible to designers.

goo.gl/DPJ3vG

1. High level APIs !== sacrificing low level control

goo.gl/DPJ3vG

2. SVG as markup. Loops are explicit.

SVG
in functions
D3

```
var link = svg.selectAll(".link")
    .data(links)
  .enter().append("path")
    .attr("class", "link")
    .attr("d", diagonal);
var node = svg.selectAll(".node")
    .data(nodes)
  .enter().append("g")
    .attr("class", "node")
    .attr("transform", function(d) {
      return "rotate(" + (d.x - 90)
    })
node.append("circle")
    .attr("r", 4.5);
node.append("text")
    .attr("dy", ".31em")
    .attr("text-anchor", function(d)
      return d.x < 180 ? "start" : "
    })
    .attr("transform", function(d) {
      return d.x < 180 ? "translate(
    })
    .text(function(d) { return d.nam
```

SVG
as markup
Victory

```
1  <path
2    className={"link"}
3    d={diagonal} />
4
5  <g
6    className={node}
7    transform={
8      "rotate("+(d.x - 90)+")
9      translate("+ d.y + ")"
10     }>
11   <circle r={4.5}/>
12   <text
13     dy={".31em"}
14     textAnchor={
15       d.x < 180 ? "start" : "end"
16     }
17     transform={
18       return d.x < 180 ?
19       "translate(8)" :
20       "rotate(180)translate(-8)";
21     }> {d.name} </text>
22 </g>
```

Iteration and data binding

D3

```
var link = svg.selectAll(".link")
    .data(links)
  .enter().append("path")
    .attr("class", "link")
    .attr("d", diagonal);
var node = svg.selectAll(".node")
    .data(nodes)
  .enter().append("g")
    .attr("class", "node")
    .attr("transform", function(d) {
      return "rotate(" + (d.x - 90)
    })
node.append("circle")
    .attr("r", 4.5);
node.append("text")
    .attr("dy", ".31em")
    .attr("text-anchor", function(d)
      return d.x < 180 ? "start" : "
    })
    .attr("transform", function(d) {
      return d.x < 180 ? "translate(
    })
    .text(function(d) { return d.nam
```

Iteration
and data binding
Victory

```
1  let nodesSVG = arrayOfNodes.map((node) => {
2    return (
3      <g
4        className={"node"}
5        transform={
6          "rotate("+(node.x - 90)+")
7          translate("+ node.y + ")"
8          }>
9        <circle r={node.r}/>
10     </g>
11   )
12 })
```

goo.gl/DPJ3vG

3. composability

(just functions after all…)

goo.gl/DPJ3vG

4. radium :D

(styles become data) talk1 talk2

Styles as data

```
style={{
    element1: {
        margin: this.props.foo > 30 ? ‘5px’ : ‘10px’
    }
}}
```

goo.gl/DPJ3vG

5. state management

State

D3

(punt)

```
// Toggle children on click.
function click(d) {
  if (d.children) {
    d._children = d.children;
    d.children = null;
  } else {
    d.children = d._children;
    d._children = null;
  }
  update(d);
}
```

goo.gl/DPJ3vG

6. Repos

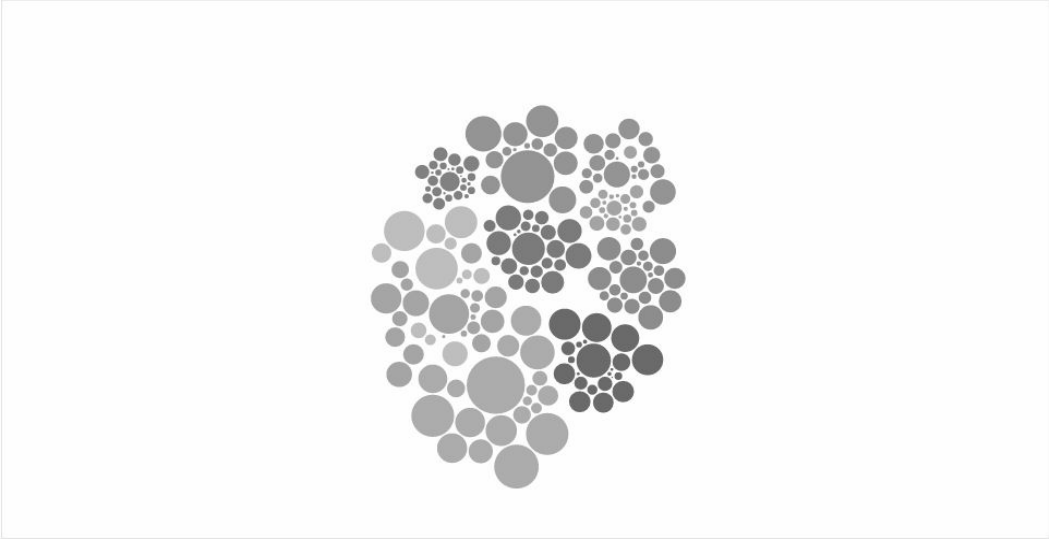
(issues, forks, PRs)

Le cargo cult

Gists encourage
copy paste coding

mbostock's block #1748247 February 5, 2012

Clustered Force Layout II



This clustered force layout is implemented using two custom forces. The first, `cluster`, pushes nodes towards the largest node of the same color. A second `collide` force prevents circles from overlapping by detecting collisions.

Open in a new window.

This example uses custom gravity applied only to the largest node of each color; compare to standard gravity.

index.html

```
<!DOCTYPE html>
<meta charset="utf-8">
<body>
<script src="https://cdnjs.cloudflare.com/ajax/libs/d3/3.5.5/d3.min.js"></script>
<script>
```

goo.gl/DPJ3vG

7. Build, infra && dist

goo.gl/DPJ3vG

- eslintrc
- Tests / Karma
- Travis
- React.PropTypes (function barrier validation)
- Babel
- Webpack
- hot reloading
- package.json
- README.md / great docs
- Sourcemaps
- minified dists
- git && Github: forking / issues / ZenHub

goo.gl/DPJ3vG

100's of component repos with infra **isn't DRY**

goo.gl/DPJ3vG

Introducing Builder

```
$ npm install builder
```

https://github.com/FormidableLabs/builder

find packages

sign up

victory-animation public

animation wrapper for victory components

8. npm

build passing

`victory-animation` is a React wrapper component that uses the D3 `interpolate` and `ease` libraries to provide transitions between prop sets.

##Examples

The most basic set up you can use will require supplying a `data` prop and rendering a functional child, as shown below:

```
<VictoryAnimation data={x: 500}>
  {(data) => {
    return <div style={{left: data.x}}/>
  }}
</VictoryAnimation>
```

The way `victory-animation` works is, when you supply the initial value for the data prop, the functional child gets called and your child/children are rendered with that data. Any subsequent data supplied via the data prop is interpolated against the original or current value, and the child is rerendered along a transition sequence until it reaches its final value, which is the prop that was supplied.

`npm i victory-animation`

boygirl published 5 days ago

0.0.10 is the latest of 10 releases

github.com/formidablelabs/victory-animation

MIT license

Collaborators

Stats

22 downloads in the last day

338 downloads in the last week

557 downloads in the last month

4 open issues on GitHub

goo.gl/DPJ3vG

9. Interactive docs with **ecology.js**

goo.gl/DPJ3vG

10. react native :)

conversion

goo.gl/DPJ3vG

In the works

- chord
- voronoi
- force
- tree
- treemap
-

goo.gl/DPJ3vG

The peak of inflated expectations was frontend before this.

We are not hiding the bad.

It’s just functions.

goo.gl/DPJ3vG

Formidable built this

Build with us

@colinmegill
formidable.com