Exploring Web Standards for Data Visualization

Nicolas Garcia Belmonte

@philogb
Nicolas Garcia Belmonte

@philogb
Why so many standards for Graphics?

- SVG
- WebGL
- HTML
- 2D Canvas
- CSS
- JavaScript
What is the right standard for my Visualization?

- SVG
- WebGL
- HTML
- 2D Canvas
- CSS
- JavaScript
Political Engagement Map

@BarackObama Tweets

@MittRomney Tweets

Barack Obama (@BarackObama)

President Obama: "Ohio, we do not need more tax cuts for folks who are already doing really well. We need tax cuts for working Americans."

1 Aug 12

Map of the United States with states highlighted in different shades of blue, indicating varying degrees of political engagement.
<table>
<thead>
<tr>
<th>Visual Component</th>
<th>Tweet Histogram</th>
<th>Choropleth Map</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><img src="image" alt="Tweet Histogram" /></td>
<td><img src="image" alt="Choropleth Map" /></td>
</tr>
<tr>
<td><strong># of Elements</strong></td>
<td>Small (~40)</td>
<td>Small (~50)</td>
</tr>
<tr>
<td><strong>Shape Complexity</strong></td>
<td>Simple: (Rectangle)</td>
<td>Complex: (Concave, Convex, Connected, Disconnected)</td>
</tr>
<tr>
<td><strong>Interactive</strong></td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>Standard Chosen</strong></td>
<td>HTML</td>
<td>SVG</td>
</tr>
</tbody>
</table>
HTML / SVG

Good for a small # of simple-to-complex shaped interactive elements
Mobility Flow in France

Per State and County Mobility Data for France
# Mobility Flow in France

Per State and County Mobility Data for France

<table>
<thead>
<tr>
<th>Visual Component</th>
<th>Choropleth Map</th>
</tr>
</thead>
<tbody>
<tr>
<td># of Elements</td>
<td>Medium/Big: ~40,000. US has only ~3,000.</td>
</tr>
<tr>
<td>Shape Complexity</td>
<td>Complex: (Concave, Convex, Connected, Disconnected)</td>
</tr>
<tr>
<td>Interactive</td>
<td>Yes</td>
</tr>
<tr>
<td>Standard Chosen</td>
<td>?</td>
</tr>
</tbody>
</table>
Mobility Flow in France
Take I
Use SVG to render the Map
Mobility Flow in France

Take 2

2D Canvas / CSS3
Mobility Flow in France
Take 2 - 2D Canvas / CSS3

- Use Layered Images to render the Map
- Canvas Color Picking for Interaction
- CSS Transitions / Transforms for Zooming / Panning
Mobility Flow in France
Canvas / CSS3
Map of France
Mobility Flow in France

Images to render the Map

outline  data  picking
Mobility Flow in France

Canvas Color Picking for fast Interaction

Each State and County is assigned a unique (r, g, b, a) tuple. We can encode up to $256^4 - 1$ data elements.
Canvas
An HTML Element

In which you can paste images

```javascript
var canvas = document.querySelector('#map'),
    ctx = canvas.getContext('2d'),
    img = new Image();

img.src = 'map.jpg';
img.onload = function() {
    ctx.drawImage(img, 0, 0);
};
```

And then retrieve it’s pixels

```javascript
var pixelArray = ctx.getImageData(0, 0, width, height);
```
2D Canvas Color Picking for fast Interaction

Offline: Encode index to county data array in colors

```javascript
counties.forEach(function(county, i) {
  var r = i % 256,
  g = ((i / 256) >>> 0) % 256,
  b = ((i / (256 * 256)) >>> 0) % 256;
  county.setAttribute('fill', 'rgb(' + r + ',' + g + ',' + b + ')');
});
```

Online: Decode RGB color to array index

```javascript
//decode index from image
function getCounty(canvas, counties, x, y) {
  var imageData = canvas.getImageData(),
      width = imageData.width,
      data = imageData.data,
      index = (x + y * width) * 4, //RGBA components
      r = data[index],
      g = data[index + 1],
      b = data[index + 2],
      i = r + (g + b * 256) * 256;
  return counties[i];
}
```
CSS3 for Zooming

CSS transition definition

```
.maps {
  transition: transform ease-out 500ms;
}
```

Set CSS transform via JavaScript

```
var style = map.style;
style.transform = 'translate(' + dx + 'px,' + dy + 'px) scale(' + s + ')';
```
Mobility Flow in France

CSS Transitions for Zooming

- Not good for synchronized / responsive animations
- GPU compositing messes up images when scaling
Almost had it...
Mobility Flow in France

WebGL

• Same image tile principle
• More control on animations
• More control on GPU management
How does WebGL work?  
... and why is it so fast?

JavaScript

WebGL JS API

GLSL API

Vertex Shader

GLSL API

Fragment Shader

Computer Screen
How does WebGL work?

The 3D scene

From Computer Desktop Encyclopedia
Reprinted with permission.
© 1998 Intergraph Computer Systems

viewing frustum
viewplane
viewpoint

image source: http://computer.yourdictionary.com/graphics
How does WebGL Scale?

Examples using PhiloGL
US Wind Patterns

Visualize the last 72 hour wind motion in the US.

Markers: disk circle line

This map tracks wind direction (lines), speed (radius) and temperature (hue) measurements from 1200 weather stations across the country. You can play an animation of the wind movements and hover the weather stations in the map to get more information. Use the slider at the bottom to focus on a particular hour.

Copyright © Nicolas Garcia Belmonte - Elevation map by UNISYS - Weather data by the National Weather Service.
Data Facts

- 1200 weather stations
- 72 hours of data
- 5 variables - latitude, longitude, speed & wind direction, temperature

= 460,000 items
I am one of the weather observers that works on Mount Washington, for the Mount Washington Observatory. I also happen to be a degreed meteorologist.

This is really neat...nice work! I can tell you for sure that the wind data you are getting for Mount Washington is completely realistic. During our winter months (October through May) we average a wind gust over hurricane force every other day, and a wind gust over 100 mph every 4th day. If you want to have a reference to compare your data to, check out the MOS data for KMWN. Because we supply our hourly observation data (on a contract basis) to the NWS, they run MOS data for the summit itself. The wind and temperature forecasts from the MOS data are typically very good, although this year they have been unusually bad, perhaps due to the unusual winter weather pattern, especially over the eastern US.

One more thing, with regards to the "Home of the World's Worst Weather" moniker. Although I will admit that I do have some degree of bias, I believe that moniker to hold a whole lot of truth. You will be hard pressed to find anywhere else in the world that sees such a regular combination of extremes like we do here: very high winds, freezing fog (rime ice), blowing snow, snow, sub-zero temperatures, visibility as low as 15 to 20 feet, etc. Making us especially unique, in addition to those weather variables, is that fact that we are continuously staffed and monitoring the conditions. Road to the top or not, Mount Washington is a very unique place in the entire world. If you lived here (week on, week off) like I have for the last 5 years, you would understand :-)

Thursday, February 28, 13
World temperature changes from 1880 to 2011

Visualize world temperature changes by clicking on the date ranges on the right. Scroll to zoom in/out and drag and drop to move the earth.
//Create application
PhiloGL('canvasId', {
    program: {
        from: 'uris',
        vs: 'shader.vs.glsl',
        fs: 'shader.fs.glsl'
    },
    camera: {
        position: {
            x: 0, y: 0, z: -50
        }
    },
    textures: {
        src: ['arroway.jpg', 'earth.jpg']
    },
    events: {
        onDragMove: function(e) {
            //do things...
        },
        onMouseWheel: function(e) {
            //do things...
        }
    },
    onError: function() {
        alert("There was an error creating the app.");
    },
    onLoad: function(app) {
        /* Do things here */
    }
});
When choosing a Standard for your Viz you could start by asking yourself about...

<table>
<thead>
<tr>
<th># of Elements</th>
<th>Small, Large</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shape Complexity</td>
<td>Simple, Complex</td>
</tr>
<tr>
<td>Interaction</td>
<td>Yes, No</td>
</tr>
<tr>
<td>Animation</td>
<td>Yes, No</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Desktop, Mobile, Browsers, etc.</td>
</tr>
<tr>
<td>Libraries</td>
<td>d3js, three.js, etc.</td>
</tr>
</tbody>
</table>
Thanks

@philogb

http://philogb.github.com/