

William Blommaert
Student Multimedia Technology
Karel de Grote Hogeschool Antwerpen

DATAVISUALISATIE MET D3.JS

WEB SPECIALISATIE 2014-2015

INHOUDSTAFEL

Inhoud

Introductie	1
Logboek	2
Werkwijze	2
Verloop	2
Slot	2
Onderzoek	3
Aanvang	3
Concurrenten	3
Syntax	4
Toepassing	5
Conclusie	6
Lessons learned	6
Gebruikte bronnen	7

Introductie

In het kader van het vak Web Specialisatie besloot ik te kiezen voor het onderwerp “Datavisualisatie”, specifiek gericht op D3.js (Data Driven Documents), aangezien dat een van de bekendere libraries is. Het doel was om dit gericht te onderzoeken en enige gebreken of nieuwigheden aan het licht te brengen. Doorheen de loop van de periode waarin dit onderzoek verliep is het vaak gebleken dat er altijd wel iets nieuws bij te leren valt en dat technologie niet stilstaat. Dingen waarvan je dacht dat ze reeds een gevestigde waarde waren kunnen alsnog verfijnd en verbeterd worden. Dit stemt dan ook overeen met een van de slagzinnen van onze opleiding, waar je altijd iets nieuws zal blijven leren doorheen je carrière.

Dit verslag legt uitgebreid(er) vast hoe mijn onderzoek naar deze library is verlopen en welke lessen ik er uit geleerd heb. Verder toon ik ook het verschil aan tussen vergelijkbare libraries en waarom men wel of niet zou moeten kiezen voor D3.js, met uiteraard de bijhorende verschillen.

Logboek

WERKWIJZE

Het logboek was een makkelijkere manier om de voortgang bij te houden. Het was een korte samenvatting en tijdsverloop dat aan de basis zal liggen van dit verslag. De werkwijze hiervan was dat we op onze persoonlijke GitHub-account een bestand zouden aanvullen, in de vorm van een logboek.

VERLOOP

Doorheen de periode van het onderzoek zijn er bronvermeldingen (ook te vinden aan het einde van dit document) toegevoegd aan dit logboek die verder geraadpleegd kunnen worden en enorm verrijkend waren op het vlak van informatie. Initieel was er een grote stroom aan informatie te verkrijgen die geleverd werd door de documentatiepagina's van de betreffende libraries. Verder was er ook enorm veel feedback te vinden in de communities, die zowel de pro's als de contra's vermelden. Op die manier was het makkelijker om objectief te oordelen over D3.js, wat zeer belangrijk was voor mijn onderzoek.

In het logboek gaf ik ook iedere keer duidelijk weer wat exact ik op dat moment heb onderzocht en welke libraries er dan bekeken werden. Verder hield ik ook een to do list bij die frequent werd geraadpleegd om bepaalde onderwerpen verder te bekijken en/of te onderzoeken.

Het belangrijkste was wellicht om dit logboek zo vaak mogelijk aan te passen, zodat ik maar ook anderen een duidelijk overzicht krijgen van waar ik

SLOT

Tot slot was dit logboek de basis voor het eindverslag. Alle informatie die ik zou kunnen aanwenden, alsook de stand van zaken was makkelijk terug te vinden in dit bestand.

Onderzoek

AANVANG

Aanvankelijk was het de bedoeling om meerdere libraries voor datavisualisatie te onderzoeken, maar door de vrij ruime beschikbaarheid en de grotere populariteit van D3.js was de keuze om daar in te specialiseren vrij snel gemaakt. Ik had er voordien nog niet met gewerkt, maar er wel al van gehoord en voorbeelden van gezien, echter nog geen praktische ervaring met deze toch wel vrij bekende library. Het sprak dan ook voor zich om me eerst wat te verdiepen in de manier van werken en welke libraries directe concurrenten waren voor D3.js.

CONCURRENTEN

D3 mag dan wel een van de grotere en bekendere libraries zijn, maar er zijn toch nog een groot aantal concurrenten beschikbaar die vergelijkbare resultaten opleveren. Het leuke daaraan is dat men zich niet moet beperken tot enkel D3.js, maar dat men vaak ook combinaties kan gebruiken van verschillende libraries. Fastcolabs.com en Sitepoint.com hadden beide een soort top 12 en top 5 gemaakt, waarin D3 iedere keer voorkwam, maar die ook een groot aantal andere libraries bevatten. Op datavisualization.ch kon ik dan weer terecht voor een groot overzicht.

De twee voornaamste namen die naar voor bleven komen tijdens het onderzoek waren Processing.js en Sigma.js. Vaak vonden er ook combinaties plaats die enorm mooie resultaten opleverden. Het leuke is dat de developers van beide libraries tot nu toe nog geen conflicten hebben opgelopen met D3.js. Het is dus perfect mogelijk om de krachtigste functies van een library te combineren met die van een andere.

Na wat verder uit te pluizen welke libraries er nog vrij bekend zijn kwam ik op de namen Raphaël en Protovis, waarvan de laatste uiteindelijk bleek dat het de voorloper van D3.js is. Uit verder onderzoek bleek ook dat Raphaël zich meer toespitst op het implementeren van vectoriële graphics in plaats van eigenlijke datavisualisatie. Dat wil echter niet zeggen dat het niet gebruikt kan worden voor het laatstgenoemde.

SYNTAX

Het belangrijkste bij het onder de knie krijgen en het begrijpen van een library is vaak toch wel de syntax. Logischerwijs begon ik dan ook de syntax van D3.js wat uit te diepen. Al snel bleek dat deze enorm lijkt op die van jQuery op het vlak van selectors en het chainen van statements. Zo hoeft je niet voor elk statement een aparte lijn code te schrijven maar kan je ze simpelweg achter elkaar plaatsen, net zoals dat het geval is bij jQuery. Blijf je liever toch aparte statements schrijven dan kan dit ook nog, D3 ondersteunt beide manieren en bekomt hoe dan ook hetzelfde resultaat.

Chainless:

```
var body = d3.select("body");  
  
var p = body.append("p");  
p.text("New paragraph!");
```

Chained:

```
d3.select("body")  
    .append("p")  
    .text("New paragraph!");
```

Een groot voordeel aan de vergelijkbare syntax is dat veel ontwikkelaars sneller hun weg zullen vinden in deze library en dat op het vlak van code alles overzichtelijk en leesbaar blijft. Door middel van de tutorial op Dashing D3.js krijgt men al snel een goed beeld van de kracht van D3. Persoonlijk gaat mijn voorkeur uit naar de chained methode, aangezien dit overzichtelijker is, al kan het natuurlijk in bepaalde gevallen nuttig zijn om met extra variabelen te werken.

TOEPASSING

De toepassing van D3 kan gaan van simpele charts, tot volledig geanimeerde overzichten die tot in de puntjes aanpasbaar zijn. Door het gebruik van SVG is het op de meest gangbare schermen altijd even duidelijk, wat bij vele eindgebruikers zowel een must als een vereiste is. Verder kan men door het gebruik van plugins de bruikbaarheid van de library nog verder uitbreiden. De hoeveelheid mogelijke toepassingen is dan ook eindeloos.

Dat de syntax makkelijk is zou voor velen kunnen betekenen dat men makkelijk en snel nieuwe charts, pies en dergelijke zou kunnen implementeren maar niks is minder waar. D3 is uitsluitend een middel om data in te laden in de browser, je zal hoe dan ook altijd zelf je charts of visualisaties in het algemeen moeten aanmaken door middel van de library.

Een nadeel zou dus kunnen zijn dat een onervaren ontwikkelaar in het begin moeite zal hebben om aan de slag te kunnen met de library, al is D3 indirect bedoeld voor de iets meer ervaren ontwikkelaar. Een kleine voorkennis en gewenning aan de syntax van een library zoals jQuery zijn zeker een pluspunt, en zelfs dan is het in het begin nog even zoeken hoe men net iets aantrekkelijk verkrijgt op het scherm.

Er zijn echter wel projecten die bouwen op de fundamentals van D3 en die trachten herbruikbaarheid te implementeren. Een voorname daarvan is NVD3, die het toelaat herbruikbare componenten en charts te implementeren zonder de kracht van D3 zelf teniet te doen. Het is een vrij jong project dat toch al zeer uitgebreid is en zeer aanpasbaar is.

Crossfilter is dan weer een javascript library die gebruik maakt van D3 om de visualisaties op te bouwen. Het is voornamelijk een library om grote datasets, sommige zelfs tot een miljoen of meer records, te kunnen verkennen en dit op een zeer performante manier. Qua aanpassing op vlak van D3 is er hier weinig te doen, maar het is wel sterk dat men op deze manier met de library aan de slag kan. Er zijn dus verschillende uitvalswegen mogelijk.

CONCLUSIE

De algemene conclusie is dat D3 een enorm krachtige tool is voor het weergeven van data. Men hoeft zich echter niet te beperken tot deze library, er zijn namelijk veel alternatieven ter beschikking die misschien makkelijker zijn in (aanvankelijk) gebruik en die niet veel verdere aanpassingen vereisen. De community rond datavisualisatie is de laatste jaren enorm gegroeid. Libraries zoals D3 hebben mee geboomd rond de SVG-groei en zullen zeker en vast nog meer interesse krijgen.

LESSONS LEARNED

- ❖ Beperk jezelf niet tot een enkele library, onderzoek de voor- en nadelen van concurrenten
- ❖ Keep it simple, de syntax maakt een groot verschil.
- ❖ Probeer hetzelfde resultaat te bekomen met verschillende libraries/API's

Gebruikte bronnen

- ❖ D3.JS - <http://d3js.org>
- ❖ Sigma.js - <http://sigmajs.org>
- ❖ Processing.js - <http://processingjs.org>
- ❖ Raphaël - [http://raphaeljs.com](http://dmitrybaranovskiy.github.io/raphael/)
- ❖ Protovis - <http://mbostock.github.io/protovis>
- ❖ Dashing D3.js - <https://www.dashingd3js.com/table-of-contents>
- ❖ Fastcolabs.com article - <http://www.fastcolabs.com/3029760/the-five-best-libraries-for-building-data-vizualizations>
- ❖ Sitepoint.com article - <http://www.sitepoint.com/twelve-javascript-libraries-data-visualization>
- ❖ DataVisualization.ch overview - <http://selection.datavisualization.ch>
- ❖ NVD3 - <http://nvd3.org>
- ❖ CrossFilter - <http://square.github.io/crossfilter>
- ❖ AlignedLeft.com - <http://alignedleft.com/tutorials/d3>